SAMA5D2 Series Silicon Errata and Data Sheet Clarification

SAMA5D2 Series

Product Page Links

SAMA5D2 Series

The SAMA5D2 series devices that you have received conform functionally to the current device data sheet (DS60001476), except for the anomalies described in this document.

The silicon issues discussed in the following pages are for silicon revisions with the Device and Revision IDs listed in the following tables. The silicon issues are summarized in the Silicon Issue Summary.

Note: This document summarizes all silicon errata issues from all revisions of silicon, both previous as well as current.

Data sheet clarifications and corrections (if applicable) are located in the Data Sheet Clarifications section, following the discussion of silicon issues.

The Device and Revision ID values for the various SAMA5D2 series silicon revisions are shown in the following tables.

Table 1. SAMA5D2x Silicon Device Identification

Part Number	Silicon Revision	Device Identification	
		CHIPID_CIDR[31:0]	CHIPID_EXID[31:0]
ATSAMA5D22A-CU	A	0x8A5C08C0	0x00000059
ATSAMA5D24A-CU	A	0x8A5C08C0	0x00000014
ATSAMA5D27A-CU	A	0x8A5C08C0	0x00000011
ATSAMA5D28A-CU	A	0x8A5C08C0	0x00000010
ATSAMA5D21B-CU	B	0x8A5C08C1	0x0000005A
ATSAMA5D22B-CN	B	0x8A5C08C1	0x00000069
ATSAMA5D22B-CU	B	0x8A5C08C1	0x00000059
ATSAMA5D23B-CN	B	0x8A5C08C1	0x00000068
ATSAMA5D23B-CU	B	0x8A5C08C1	0x00000058
ATSAMA5D24B-CU	B	0x8A5C08C1	0x00000014
ATSAMA5D26B-CN	B	0x8A5C08C1	0x00000022
ATSAMA5D26B-CU	B	0x8A5C08C1	0x00000012
ATSAMA5D27B-CN	B	0x8A5C08C1	0x00000021
ATSAMA5D27B-CU	B	0x8A5C08C1	0x00000011
ATSAMA5D28B-CN	B	0x8A5C08C1	0x00000020
ATSAMA5D28B-CU	B	0x8A5C08C1	0x00000010
ATSAMA5D21C-CU	C	0x8A5C08C2 or 0x8A5C08C4	0x0000005A
ATSAMA5D22C-CN	C	0x8A5C08C2 or 0x8A5C08C4	0x00000069
ATSAMA5D22C-CU	C	0x8A5C08C2 or 0x8A5C08C4	0x00000059
ATSAMA5D23C-CN	C	0x8A5C08C2 or 0x8A5C08C4	0x00000068
ATSAMA5D23C-CU	C	0x8A5C08C2 or 0x8A5C08C4	0x00000058
ATSAMA5D24C-CU	C	0x8A5C08C2 or 0x8A5C08C4	0x00000014

Table 1. SAMA5D2x Silicon Device Identification (continued)

Part Number	Silicon Revision	Device Identification	
		CHIPID_CIDR[31:0]	CHIPID_EXID[31:0]
ATSAMA5D26C-CN	C	0x8A5C08C2 or 0x8A5C08C4	0x00000022
ATSAMA5D26C-CU	C	0x8A5C08C2 or 0x8A5C08C4	0x00000012
ATSAMA5D27C-CN	C	0x8A5C08C2 or 0x8A5C08C4	0x00000021
ATSAMA5D27C-CU	C	0x8A5C08C2 or 0x8A5C08C4	0x00000011
ATSAMA5D28C-CN	C	0x8A5C08C2 or 0x8A5C08C4	0x00000020
ATSAMA5D28C-CU	C	0x8A5C08C2 or 0x8A5C08C4	0x00000010
SAMA5D27(T)-V/8GW	C	0x8A5C08C4	0x00000021
SAMA5D27(T)-V/8GW-SL3	C	0x8A5C08C4	0x00000003

Note: Refer to the “Chip Identifier (CHIPID)” and "Product Identification System" sections in the current device data sheets (DS60001476 and DS60001532) for detailed information on chip identification and version for your specific device.

1. Silicon Issue Summary

Table 1-1. Silicon Issue Summary

Module	Item/Feature	Summary	Affected Silicon Revisions		
			A	B	C
ACC	ACC output connection issue	The Analog Comparator (ACC) output is not connected to the PWM event line.	X		
ADC	SleepWalking	ADC SleepWalking is not functional.	X		
ADC	Last channel trigger	Last channel trigger limitation	X		
ADC	Trigger events	ADC trigger events RTCOUT0 and RTCOUT1 are not functional.	X		
CLASSD	Differential Output mode	Unexpected offset and noise level in Differential Output mode	X		
AUDIO PLL	Audio PLL output frequency range	Audio PLL output frequency range not compliant	X		
FLEXCOM	FLEXCOM SMBUS alert	FLEXCOM SMBUS alert signaling is not functional	X	X	X
GMAC	Timestamps and PTP packets	Bad association of timestamps and PTP packets	X	X	X
GMAC	Screening registers not working	Screening registers (GMAC_ST1RPQx and GMAC_ST2RPQx) not working	X	X	X
HSIC	HSIC startup	At HSIC start-up, the strobe default state is wrong	X		
I^{2}SC	I^{2}SC sent data	I^{2}SC first sent data corrupted	X	X	X
MCAN(1)	CRC	Flexible data rate feature does not support CRC	X	X	X
MCAN(1)	MCAN_IR.MRAF interrupt	Needless activation of interrupt MCAN_IR.MRAF	X	X	X
MCAN(2)	Bus Integration state	Return of receiver from Bus Integration state after Protocol Exception Event	X	X	X
MCAN(2)	Message RAM/RAM Arbiter	Message RAM/RAM Arbiter not responding in time	X	X	X
MCAN(2)	Frame receiving	Data loss (payload) in case storage of a received frame has not completed until end of EOF field is reached	X	X	X
MCAN(1)	Edge filtering	Edge filtering causes mis-synchronization when falling edge at Rx input pin coincides with end of integration phase	X	X	X
MCAN(2)	MCAN_NBTP.NTSEG2	Configuration of MCAN_NBTP.NTSEG2 = '0' not allowed	X	X	X
MCAN(2)	DAR mode	Retransmission in DAR mode due to lost arbitration at the first two identifier bits	X	X	X
MCAN(2)	Tx FIFO message	Tx FIFO message sequence inversion	X	X	X
MCAN(2)	HPM interrupt	Unexpected High Priority Message (HPM) interrupt	X	X	X
MCAN(2)	Transmitted message	Issue message transmitted with wrong arbitration and control fields	X	X	X
MCAN(2)	Debug message handling state machine not reset	Debug message handling state machine not reset to Idle state when CCCR.INIT is set	X	X	X
MCAN(1)	Frame transmitted despite cancellation	Frame transmitted despite confirmed transmit cancellation for CAN-FD messages with more than 8 data bytes	X	X	X
MPDDRC	t_{FAW} timing	t_{FAW} timing violation	X		
PMC	GCLK fields	GCLK fields are reprogrammed unexpectedly	X		
PMC	PMC SleepWalking	PMC SleepWalking is not functional	X		

Table 1-1. Silicon Issue Summary (continued)

Module	Item/Feature	Summary	Affected Silicon Revisions		
			A	B	C
PMC	PMC_MCKR.PRES field	Change of the field PMC_MCKR.PRES is not allowed if Master/Processor Clock Prescaler frequency is too high	X	X	X
PTC	Wrong pull-up value on PD[18:3] during reset	Incorrect pull-up value	X	X	X
PWM	Fault Protection to Hi-Z for PWMx output	Fault Protection to Hi-Z for PWMx output is not functional	X	X	X
QSPI	DLYCS delay	QSPI hangs with long DLYCS	X	X	X
RTC	RTC_SR.TDERR flag	RTC_SR.TDERR flag is stuck at 0	X	X	X
RTC	Truncated read access to RTC_TIMALR (UTC_MODE)	Read access truncated to the first 24 bits for register RTC_TIMALR (UTC_MODE)	X	X	X
ROM Code	Frequency support for SAM-BA Monitor	Main external clock frequency support for SAM-BA® Monitor limitation	X		
ROM Code	Watchdog	Watchdog reset occurs when reenabling the watchdog	X		
ROM Code	SPI frequency	SPI frequency at bootup is not 11 MHz	X		
ROM Code	JTAG_TCK	JTAG_TCK on IOSET 4 pin has a wrong configuration after boot	X	X	X
ROM Code	SDMMC0 and SDMMC1 boot	The card detect pin is not correctly sampled in the ROM code		X	
ROM Code	UART connection to SAM-BA Monitor	UART blocks USB connection to SAM-BA Monitor	X	X	X
ROM Code	Secure Boot Mode: AES-RSA X.509 Certificate Serial Number Length Limit	The length of serial numbers is limited to 16 bytes by the ROM code.	X	X	X
SDMMC	Software 'Reset For all' command	Software 'Reset For all' command is not guaranteed	X	X	X
SDMMC	Status flag INTCLKS	Status flag INTCLKS may not work correctly	X		
SDMMC	Sampling clock tuning procedure	Sampling clock tuning procedure may freeze	X	X	X
SDMMC	SDMMC I/O calibration does not work	The impedance calibration mechanism for the SDMMC I/Os does not work	X	X	X
SFC	Partial Fuse Masking	The Partial Fuse Masking function does not work	X		
SFC	Fuse matrix first bits	The first two bits of each 32-bit block of the fuse matrix cannot be written	X		
SFC	Fuse matrix programming	Fuse matrix programming requires a main clock (MAINCK) frequency between 10 and 15 MHz	X	X	X
SFC	Fuse matrix read	Fuse matrix read requires a main clock (MAINCK) frequency below 28 MHz	X	X	X
SFR	Serial number	The serial number stored in the SFR registers (SFR_SN0 and SFR_SN1) is not correct	X		
SSC	TD output	Unexpected delay on TD output	X	X	X
TWIHS	Clear command	The TWI/TWIHS Clear command does not work	X	X	X
WDT	Restart command	Restart command of WDT may reset the DDR controller	X	X	X

Notes:

1. This erratum is not relevant for CAN 2.0.
2. This erratum is applicable for CAN 2.0.

2. Analog Comparator Controller (ACC)

2.1. ACC output connection issue

The Analog Comparator (ACC) output is not connected to the PWM event line.

Work around

None

Affected Silicon Revisions

A	B	C					
X							

3. Analog-to-Digital Converter (ADC)

3.1. ADC SleepWalking is not functional

Work around

None

Affected Silicon Revisions

A	B	C					
X							

3.2. Last channel trigger limitation

The last channel can be triggered at low speed but cannot be programmed by the OUT1 field of the RTC. Only the 1-Hz sampling period is available.

Work around

None

Affected Silicon Revisions

A	B	C					
X							

3.3. ADC trigger events RTCOUT0 and RTCOUT1 are not functional

RTCOUT0 issue leads to ADC Sleepwalking not functional.

RTCOUT1 issue makes the last channel specific measurement trigger work at 1 Hz only.

Work around

None

Affected Silicon Revisions

A	B	C					
X							

4. Audio Class D Amplifier (CLASSD)

4.1. Unexpected offset and noise level in Differential Output mode

When the CLASSD peripheral is set to Differential Output mode (PWMTYP = 1), a significant output offset and an increased level of noise are observed on the audio outputs. The offset is systematic and is equal to 1/16 of the digital full scale.

Work around

To avoid the offset, add the opposite offset on the input signal of the CLASSD peripheral.

Affected Silicon Revisions

A	B	C					
X							

5. Audio PLL

5.1. Audio PLL output frequency range

The frequency range of the AUDIOCORECLK signal (AUDIOPLL output) provided in table “Audio PLL Characteristics” (f_{CORE} parameter), in section Electrical Characteristics of the device data sheet (DS60001476), does not comply with the applicable specification.

Work around

The AUDIOCORECLK signal can be operated from 720 MHz to 790 MHz if the following restricted operating conditions are met:

- Junction temperature (T_j) range: 0°C to +40°C
- VDDCORE/VDDPLL supply range: 1.20V to 1.32V
- Bits <29:28> in register PMC_AUDIO_PLL0 are set to $(01)_2$

Affected Silicon Revisions

A	B	C					
X							

6. Controller Area Network (MCAN)

6.1. Flexible data rate feature does not support the ISO 16845-1:2016 CRC

Attention: This erratum is not relevant for CAN 2.0.

CAN-FD peripheral does not support the ISO 16845-1:2016 CRC scheme which includes the stuff bit count introduced by the ISO standardization committee.

CAN 2.0 operation is not impacted.

Work around

None

Affected Silicon Revisions

A	B	C					
X	X	X					

6.2. Needless activation of interrupt MCAN_IR.MRAF

Attention: This erratum is applicable for CAN 2.0.

During frame reception while the MCAN is in Error Passive state and the Receive Error Counter has the value MCAN_ECR.REC = 127, it may happen that MCAN_IR.MRAF is set although there was no Message RAM access failure. If MCAN_IR.MRAF is enabled, an interrupt to the Host CPU is generated.

Work around

The Message RAM Access Failure interrupt routine needs to check whether MCAN_ECR.RP = ’1’ and MCAN_ECR.REC = 127. In this case, reset MCAN_IR.MRAF. No further action is required.

Affected Silicon Revisions

A	B	C					
X	X	X					

6.3. Return of receiver from Bus Integration state after Protocol Exception Event

Attention: This erratum is not relevant for CAN 2.0.

In case a started transmission is aborted shortly before the transmission of the FDF bit, a receiver will detect a recessive FDF bit followed by a recessive res bit. In this case receiving MCANs with Protocol Exception Event Handling enabled will detect a protocol exception event and will enter Bus Integration state. These receivers are expected to leave Bus Integration state after 11 consecutive recessive bits.

Instead of starting to count 11 recessive bits directly after entering Bus Integration state, the MCAN needs to see at least one dominant bit.

Work around

Disable Protocol Exception Event Handling (MCAN_CCCR.PXHD = ’1’).

Affected Silicon Revisions

A	B	C					
X	X	X					

6.4. Message RAM/RAM Arbiter not responding in time

Attention: This erratum is applicable for CAN 2.0.

When the MCAN wants to store a received frame, and the Message RAM/RAM Arbiter does not respond in time, this message cannot be stored completely and it is discarded with the reception of the next message. Interrupt flag MCAN_IR.MRAF is set. It may happen that the next received message is stored incomplete. In this case, the respective Rx Buffer or Rx FIFO element holds inconsistent data.

Work around

Configure the RAM Watchdog to the maximum expected Message RAM access delay. In case the Message RAM / RAM Arbiter does not respond within this time, the Watchdog Interrupt MCAN_IR.WDI is set. In this case discard the frame received after MCAN_IR.MRAF has been activated.

Affected Silicon Revisions

A	B	C					
X	X	X					

6.5. Data loss (payload) in case storage of a received frame has not completed until end of EOF field is reached

Attention: This erratum is applicable for CAN 2.0.

This erratum is applicable only if the MCAN peripheral clock frequency is below 77 MHz.

During frame reception, the Rx Handler needs access to the Message RAM for acceptance filtering (read access) and storage of accepted messages (write access).

The time needed for acceptance filtering and storage of a received message depends on the MCAN peripheral clock frequency, the number of MCANs connected to a single Message RAM, the Message RAM arbitration scheme, and the number of configured filter elements.

In case storage of a received message has not completed until the end of the received frame is reached, the following faulty behavior can be observed:

- The last write to the Message RAM to complete storage of the received message is omitted, this data is lost. Applies for data frames with DLC > 0, worst case is DLC = 1.
- Rx FIFO: FIFO put index MCAN_RXFnS.FnPI is updated although the last FIFO element holds corrupted data.
- Rx Buffer: New Data flag MCAN_NDATn.NDxx is set although the Rx Buffer holds corrupted data.
- Interrupt flag MCAN_IR.MRAF is not set.

Work around

Reduce the maximum number of configured filter elements for the MCANs attached to the Message RAM until the calculated clock frequency is below the MCAN peripheral clock frequency used with the device.

Affected Silicon Revisions

A	B	C					
X	X	X					

6.6. Edge filtering causes mis-synchronization when falling edge at Rx input pin coincides with end of integration phase

Attention: This erratum is not relevant for CAN 2.0.

When edge filtering is enabled (MCAN_CCCR.EFBI = ’1’) and when the end of the integration phase coincides with a falling edge at the Rx input pin, it may happen that the MCAN synchronizes itself wrongly and does not correctly receive the first bit of the frame. In this case the CRC will detect that the first bit was received incorrectly; it will rate the received FD frame as faulty and an error frame will be sent.

The issue only occurs when there is a falling edge at the Rx input pin (CANRX) within the last time quantum (tq) before the end of the integration phase. The last time quantum of the integration phase is at the sample point of the 11th recessive bit of the integration phase. When the edge filtering is enabled, the bit timing logic of the MCAN sees the Rx input signal delayed by the edge filtering. When the integration phase ends, the edge filtering is automatically disabled. This affects the reset of the FD CRC registers at the beginning of the frame. The Classical CRC registers are not affected, so this issue does not affect the reception of Classical frames.

In CAN communication, the MCAN may enter integrating state (either by resetting MCAN_CCCR.INIT or by protocol exception event) while a frame is active on the bus. In this case the 11 recessive bits are counted between the Acknowledge bit and the following start of frame. All nodes have synchronized at the beginning of the dominant Acknowledge bit. This means that the edge of the following Start-of-Frame bit cannot fall on the sample point, so the issue does not occur. The issue occurs only when the MCAN is, by local errors, mis-synchronized with regard to the other nodes, or not synchronized at all.

Glitch filtering as specified in ISO 11898-1:2015 is fully functional.

Edge filtering was introduced for applications where the data bit time is at least two tq (of the nominal bit time) long. In that case, edge filtering requires at least two consecutive dominant time quanta before the counter counting the 11 recessive bits for idle detection is restarted. This means edge filtering covers the theoretical case of occasional 1-tq-long dominant spikes on the CAN bus that would delay idle detection. Repeated dominant spikes on the CAN bus would disturb all CAN communication, so the filtering to speed up idle detection would not help network performance.

When this rare event occurs, the MCAN sends an error frame and the sender of the affected frame retransmits the frame. When the retransmitted frame is received, the MCAN has left the integration phase and the frame will be received correctly. Edge filtering is only applied during integration phase; it is never used during normal operation. As the integration phase is very short with respect to “active communication time”, the impact on total error frame rate is negligible. The issue has no impact on data integrity.

The MCAN enters integration phase under the following conditions:

- when MCAN_CCCR.INIT is set to ’0’ after start-up

- after a protocol exception event (only when MCAN_CCCR.PXHD = ’0’)

Work Around

Disable edge filtering or wait on retransmission in case this rare event happens.

Affected Device Revisions

A	B	C					
X	X	X					

6.7. Configuration of MCAN_NBTP.NTSEG2 = ’0’ not allowed

Attention: This erratum is applicable for CAN 2.0.

When MCAN_NBTP.NTSEG2 is configured to zero (Phase_Seg2(N) = 1), and when there is a pending transmission request, a dominant third bit of Intermission may cause the MCAN to wrongly transmit the first identifier bit dominant instead of recessive, even if this bit was configured as ’1’ in the MCAN’s Tx Buffer Element.

A phase buffer segment 2 of length ’1’ (Phase_Seg2(N) = 1) is not sufficient to switch to the first identifier bit after the sample point in Intermission where the dominant bit was detected.

The CAN protocol according to ISO 11898-1 defines that a dominant third bit of Intermission causes a pending transmission to be started immediately. The received dominant bit is handled as if the MCAN has transmitted a Start-of-Frame (SoF) bit.

The ISO 11898-1 specifies the minimum configuration range for Phase_Seg2(N) to be 2..8 tq. Therefore excluding a Phase_Seg2(N) of ’1’ will not affect MCAN conformance.

Work Around

Use the range 1..127 for MCAN_NBTP.NTSEG2 instead of 0..127.

Affected Device Revisions

A	B	C					
X	X	X					

6.8. Retransmission in DAR mode due to lost arbitration at the first two identifier bits

Attention: This erratum is applicable for CAN 2.0.

When the MCAN is configured in DAR mode (MCAN_CCCR.DAR = ’1’) the Automatic Retransmission for transmitted messages that have been disturbed by an error or have lost arbitration is disabled. When the transmission attempt is not successful, the Tx Buffer’s transmission request bit (MCAN_TXBRP.TRPxx) shall be cleared and its Cancellation Finished bit (MCAN_TXBCF.CFxx) shall be set.

When the transmitted message loses arbitration at one of the first two identifier bits, it may happen that instead of the bits of the actually transmitted Tx Buffer, the MCAN_TXBRP.TRPxx and MCAN_TXBCF.CFxx bits of the previously started Tx Buffer (or Tx Buffer 0 if there is no previous transmission attempt) are written (MCAN_TXBRP.TRPxx = ’0’, MCAN_TXBCF.CFxx = ’1’).

If in this case the MCAN_TXBRP.TRPxx bit of the Tx Buffer that lost arbitration at the first two identifier bits has not been cleared, retransmission is attempted.

When the MCAN loses arbitration again at the immediately following retransmission, then actually and previously transmitted Tx Buffers are the same and this Tx Buffer's MCAN_TXBRP.TRPxx bit is cleared and its MCAN_TXBCF.CFxx bit is set.

Work Around

None

Affected Device Revisions

A	B	C					
X	X	X					

6.9. Tx FIFO message sequence inversion

Attention: This erratum is applicable for CAN 2.0.

Assume the case that there are two Tx FIFO messages in the output pipeline of the Tx Message Handler. Transmission of Tx FIFO message 1 is started:

- Position 1: Tx FIFO message 1 (transmission ongoing)
- Position 2: Tx FIFO message 2
- Position 3: --

Now a non-Tx FIFO message with a higher CAN priority is requested. Due to its priority it will be inserted into the output pipeline. The TxMH performs so called "message scans" to keep the output pipeline up to date with the highest priority messages from the Message RAM. After the following two message scans, the output pipeline has the following content:

- Position 1: Tx FIFO message 1 (transmission ongoing)
- Position 2: non-Tx FIFO message with higher CAN priority
- Position 3: Tx FIFO message 2

If the transmission of Tx FIFO message 1 is not successful (lost arbitration or CAN bus error) it is pushed from the output pipeline by the non-Tx FIFO message with higher CAN priority. The following scan re-inserts Tx FIFO message 1 into the output pipeline at position 3:

- Position 1: non-Tx FIFO message with higher CAN priority (transmission ongoing)
- Position 2: Tx FIFO message 2
- Position 3: Tx FIFO message 1

Now Tx FIFO message 2 is in the output pipeline in front of Tx FIFO message 1 and they are transmitted in that order, resulting in a message sequence inversion.

Work Around

1. First Work Around

Use two dedicated Tx Buffers, e.g. use Tx Buffers 4 and 5 instead of the Tx FIFO. The pseudo-code below replaces the function that fills the Tx FIFO.

Write message to Tx Buffer 4.

Transmit loop:

- Request Tx Buffer 4 - write MCAN_TXBAR.A4
- Write message to Tx Buffer 5

- Wait until transmission of Tx Buffer 4 completed - MCAN_IR.TC, read MCAN_TXBTO.TO4
- Request Tx Buffer 5 - write MCAN_TXBAR.A5
- Write message to Tx Buffer 4
- Wait until transmission of Tx Buffer 5 is completed - MCAN_IR.TC, read MCAN_TXBTO.TO5

2. Second Work Around

Make sure that only one Tx FIFO element is pending for transmission at any time. The Tx FIFO elements may be filled at any time with messages to be transmitted, but their transmission requests are handled separately. Each time a Tx FIFO transmission has completed and the Tx FIFO gets empty (MCAN_IR.TFE = ’1’), the next Tx FIFO element is requested.

3. Third Work Around

Use only a Tx FIFO. Send the message with the higher priority also from Tx FIFO.

One drawback is that the higher priority message has to wait until the preceding messages in the Tx FIFO have been sent.

Affected Device Revisions

A	B	C					
X	X	X					

6.10. Unexpected High Priority Message (HPM) interrupt

Attention: This erratum is applicable for CAN 2.0.

This issue occurs in two configurations:

Configuration A:

- At least one Standard Message ID Filter Element is configured with Priority flag set (S0.SFEC = "100"/"101"/"110").
- No Extended Message ID Filter Element is configured.
- Non-matching extended frames are accepted (MCAN_GFC.ANFE = "00"/"01").

The HPM Interrupt flag MCAN_IR.HPM is set erroneously on reception of a non-high-priority extended message under the following conditions:

1. A standard HPM frame is received, and accepted by a filter with Priority flag set. Then, Interrupt flag MCAN_IR.HPM is set as expected.
2. Next, an extended frame is received and accepted due to the MCAN_GFC.ANFE configuration. Then, Interrupt flag MCAN_IR.HPM is set erroneously.

Configuration B:

- At least one Extended Message ID Filter Element is configured with Priority flag set (F0.EFEC = "100"/"101"/"110").
- No Standard Message ID Filter Element is configured.
- Non-matching standard frames are accepted (MCAN_GFC.ANFS = "00"/"01").

The HPM Interrupt flag MCAN_IR.HPM is set erroneously on reception of a non-high-priority standard message under the following conditions:

1. An extended HPM frame is received, and accepted by a filter with Priority flag set. Then, Interrupt flag MCAN_IR.HPM is set as expected.

2. Next, a standard frame is received and accepted due to the MCAN_GFC.ANFS configuration. Then, Interrupt flag MCAN_IR.HPM is set erroneously.

Work Around

Configuration A:

Set up an Extended Message ID Filter Element with the following configuration:

- F0.EFEC = "001"/"010" - select Rx FIFO for storage of extended frames
- F0.EFID1 = any value - value not relevant as all ID bits are masked out by F1.EFID2
- F1.EFT = "10" - classic filter, F0.EFID1 = filter, F1.EFID2 = mask
- F1.EFID2 = zero - all bits of the received extended ID are masked out

Now, all extended frames are stored in Rx FIFO 0 respectively Rx FIFO 1 depending on the configuration of F0.EFEC.

Configuration B:

Set up a Standard Message ID Filter Element with the following configuration:

- S0.SFEC = "001"/"010" - select Rx FIFO for storage of standard frames
- S0.SFID1 = any value - value not relevant as all ID bits are masked out by S0.SFID2
- S0.SFT = "10" - classic filter, S0.SFID1 = filter, S0.SFID2 = mask
- S0.SFID2 = zero - all bits of the received standard ID are masked out

Now, all standard frames are stored in Rx FIFO 0 respectively Rx FIFO 1 depending on the configuration of S0.SFEC.

Affected Device Revisions

A	B	C					
X	X	X					

6.11. Issue message transmitted with wrong arbitration and control fields

Attention: This erratum is applicable for CAN 2.0.

When the following conditions are met, a message with wrong ID, format, and DLC is transmitted:

- M_CAN is in state "Receiver" (PSR.ACT = "10") and there is no pending transmission.
- A new transmission is requested before the third Intermission bit is reached.
- The CAN bus is sampled dominant at the third Intermission bit which is treated as SoF (see ISO11898-1:2015 Section 10.4.2.2).

Then, it can happen that:

- The Shift register is not loaded with the ID, format and DLC of the requested message.
- The MCAN starts arbitration with wrong ID, format, and DLC on the next bit.
- If the ID wins arbitration, a CAN message with valid CRC is transmitted.
- If this message is acknowledged, the ID stored in the Tx Event FIFO is the ID of the requested Tx message, and not the ID of the message transmitted on the CAN bus, and no error is detected by the transmitting MCAN.

Work Around

Request a new transmission only if another transmission is already pending or when the MCAN is not in "Receiver" state (when PSR.ACT ≠ "10").

To avoid activating the transmission request in the critical time window between the sample points of the second and third Intermission bits, the application software can evaluate the Rx Interrupt flags IR.DRX, IR.RF0N and IR.RF1N, which are set at the last EoF bit when a received and accepted message becomes valid.

The last EoF bit is followed by three Intermission bits. Therefore, the critical time window has safely terminated three bit times after the Rx interrupt. Now a transmission can be requested by writing to TXBAR.

After the interrupt, the application has to take care that the transmission request for the CAN Protocol Controller is activated before the critical window of the following reception is reached.

A checksum covering the arbitration and control fields can be added to the data field of the message to be transmitted, to detect frames transmitted with wrong arbitration and control fields.

Affected Device Revisions

A	B	C					
X	X	X					

6.12. Debug message handling state machine not reset to Idle state when CCCR.INIT is set

Attention: This erratum is applicable for CAN 2.0.

In case MCAN_CCCR.INIT is set by the Host by writing to register MCAN_CCCR or when the CAN enters BusOff state, the debug message handling state machine stays in its current state instead of being reset to Idle state. Setting MCAN_CCCR.CCE does not change MCAN_RXF1S.DMS.

Work Around

In case the debug message handling state machine has stopped while MCAN_RXF1S.DMS="01" or MCAN_RXF1S.DMS="10", it can be reset to Idle state by a hardware reset or by reception of debug messages after MCAN_CCCR.INIT is reset to zero.

Affected Device Revisions

A	B	C					
X	X	X					

6.13. Frame transmitted despite confirmed transmit cancellation for CAN-FD messages with more than 8 data bytes

Attention: This erratum is not relevant for CAN 2.0.

In case the transmission of Tx Buffer nn was not successful and is restarted immediately afterwards by automatic retransmission, and the software requests a Tx cancellation for this Tx Buffer by setting the cancellation request bit MCAN_TXBCR.CRnn during transmission of the first 4 identifier bits, a successful cancellation is incorrectly signalled by setting MCAN_TXBCF.CFnn = ’1’ and by clearing MCAN_TXBRP.TRPnn. In addition, the respective Transmission Occurred bit remains zero (MCAN_TXBTO.TOnn = ’0’), incorrectly indicating that the frame was not transmitted on the bus.

Other than signalled by MCAN_TXBCF.CFnn and MCAN_TXBTO.TOnn, the transmission continues until the complete frame has been sent on the CAN bus. If the transmission is successful, MCAN_TXBTO.TOnn will be set.

If in this case new data is written to Tx Buffer nn while the transmission is still ongoing, a frame with inconsistent data may appear on the bus.

This problem is limited to the case of transmit cancellation of CAN-FD messages with more than 8 data bytes while automatic retransmission is enabled (MCAN_CCCR.DAR = ’0’).

Transmit cancellation of Classical CAN messages and CAN-FD messages with up to 8 data bytes is not affected.

CAN 2.0 operation is not impacted.

Work Around

Do not use transmit cancellation for CAN-FD messages with more than 8 data bytes.

Alternatively, wait for the duration of the expected transmission time of the cancelled Tx Buffer before writing new data to that Tx Buffer.

Affected Device Revisions

A	B	C					
X	X	X					

7. Ethernet MAC (GMAC)

7.1. Bad association of timestamps and PTP packets

An issue in the association mechanism between event registers and queued PTP packets may lead to timestamps incorrectly associated with these packets.

Even if it is highly unlikely to queue consecutive packets of the same type, there is no way to know which frame the content of the PTP event registers refers to.

Work around

None

Affected Silicon Revisions

A	B	C					
X	X	X					

7.2. Screening registers not working

GMAC Screening registers Type 1 and Type 2 (GMAC_ST1RPQx and GMAC_ST2RPQx) are not working.

Work around

None

Affected Silicon Revisions

A	B	C					
X	X	X					

8. Flexible Serial Communication Controller (FLEXCOM)

8.1. FLEXCOM SMBUS alert signalling is not functional

The TWI function embedded in the FLEXCOM does not support SMBUS alert signal management.

Work around

If this signal is mandatory in the application, the user can use one of the standalone TWIs (TWIHS0, TWIHS1) supporting the SMBUS alert signaling.

Affected Silicon Revisions

A	B	C					
X	X	X					

9. Inter-IC Sound Controller (I^{2}SC)

9.1. I^{2}SC first sent data corrupted

Right after I^{2}SC reset, the first data sent by I^{2}SC controller on the I2SDO line is corrupted. The following data are not affected.

Work around

None

Affected Silicon Revisions

A	B	C					
X	X	X					

10. Multiport DDR-SDRAM Controller (MPDDRC)

10.1. t_{FAW} timing violation

DDR2/LPDDR2 memory devices with 8 banks have an additional requirement for tFAW: no more than four Activate commands must be issued in any given t_{FAW} period.

Work around

Increase the value of t_{RRD} to 3 to avoid the issue.

Affected Silicon Revisions

A	B	C					
X							

11. Peripheral Touch Controller (PTC)

11.1. Wrong pull-up value on PD[18:3] during reset

The PTC ADC includes pull-up resistors (10 kΩ±30%) connected on PD[18:3] which are normally disabled at reset.

Because of an incorrect control of the internal pull-up disable signal, these pull-up resistors are connected temporarily to the pads at reset.

The 10 kΩ pullups are disconnected when the reset phase is completed and the internal resets have been released. The pull-up value is then ~380 kΩ.

Work around

None

Affected Silicon Revisions

A	B	C					
X	X	X					

12. Power Management Controller (PMC)

12.1. GCLK fields are reprogrammed unexpectedly

When configuring a peripheral featuring no GCLK, the GCLK fields (GCKEN, GCKCSS, GCKDIV) of FLEXCOM0 are reconfigured. No other parameter is modified.

Work around

When accessing a peripheral featuring no GCLK, fill GCLK fields with FLEXCOM0 GCLK configuration.

Affected Silicon Revisions

A	B	C					
X							

12.2. PMC SleepWalking is not functional

In Ultra-Low Power mode (ULP1) using simultaneously partial wakeup (SleepWalking) and full wakeup (PIOBU used as wakeup pins or internal events RTC, etc.) may not resume from ULP1.

Work around

None

Affected Silicon Revisions

A	B	C					
X							

12.3. Change of the field PMC_MCKR.PRES is not allowed if Master/Processor Clock Prescaler frequency is too high

PMC_MCKR.PRES cannot be changed if the clock applied to the Master/Processor Clock Prescaler (see “Master Clock Controller” in section “Power Management Controller (PMC)” of the SAMA5D2 Series data sheet) is greater than 312 MHz (VDDCORE[1.1, 1.32]) and 394 MHz (VDDCORE[1.2, 1.32]).

Work around

1. Set PMC_MCKR.CSS to MAIN_CLK.
2. Set PMC_MCKR.PRES to the required value.
3. Change PMC_MCKR.CSS to the new clock source (PLLA_CLK, UPLLCK).

Affected Silicon Revisions

A	B	C					
X	X	X					

13. Pulse Width Modulation Controller (PWM)

13.1. Fault Protection to Hi-Z for PWMx output not functional

While it is possible to force the output of PWMH and PWML to 0 or 1, the feature to set these outputs to Hi-Z by setting the corresponding field in PWM_FPV2 is not functional.

The protection values for PWML and PWMH are by default set to ‘0’.

Work around

None

Affected Silicon Revisions

A	B	C					
X	X	X					

14. Quad Serial Peripheral Interface (QSPI)

14.1. QSPI hangs with long DLYCS

QSPI hangs if a command is written to any QSPI register during the DLYCS delay. There is no status bit to flag the end of the delay.

Work around

The field DLYCS defines a minimum period for which Chip Select is de-asserted, required by some memories. This delay is generally < 60 ns and comprises internal execution time, arbitration and latencies. Thus, DLYCS must be configured to be slightly higher than the value specified for the slave device. The software must wait for this same period of time plus an additional delay before a command can be written to the QSPI.

Affected Silicon Revisions

A	B	C					
X	X	X					

15. Real-Time Clock (RTC)

15.1. RTC_SR.TDERR flag is stuck at 0

The TDERR flag reporting internal free counters errors is stuck at 0. The non-BCD or invalid date/ time values are not reported in the RTC Status register (RTC_SR).

Work around

None. A software procedure to check the validity of the RTC time and date values may be implemented.

Affected Silicon Revisions

A	B	C					
X	X	X					

15.2. Read access truncated to the first 24 bits for register RTC_TIMALR (UTC_MODE)

The register RTC_TIMALR (UTC_MODE) is a 32-bit read/write register but the bits 24:31 are write only.

RTC_TIMALR (UTC_MODE) is functioning properly but any read after write of this register will show the value 0 for the upper byte.

Work around

None.

Affected Silicon Revisions

A	B	C					
X	X	X					

16. ROM Code

16.1. Main external clock frequency support for SAM-BA Monitor limitation

ROM code v1.1 supports ONLY a 12 and 16 MHz external clock frequency to allow USB connection to be used for SAM-BA Monitor.

Work around

None

Affected Silicon Revisions

A	B	C					
X							

16.2. Watchdog reset occurs when reenabling the watchdog

When no bootable program is found in an external memory, the watchdog is disabled just before the ROM Code runs SAM-BA Monitor. The ROM code sets the watchdog Timer Mode register (WDT_MR) to the value 0x00008000 and then clears the counter value. If a program loaded and executed using the SAM-BA Monitor Go command reenables the watchdog, a watchdog reset is immediately executed whatever the value of the watchdog counter.

Work around

To avoid any unexpected watchdog reset when reenabling the watchdog, the following sequence has to be performed:

1. Write 0x00000000 in the WDT_MR.
2. Wait for three slow clock cycles.
3. Write the final value in the WDT_MR.

Affected Silicon Revisions

A	B	C					
X							

16.3. SPI frequency at bootup is not 11 MHz

The SPI frequency at the boot of the device is set to 16 MHz instead of 11 MHz. A margin was applied to the SPI timings and make them compliant with a 16 MHz clock. The SPI boot remains functional.

Work around

None

Affected Silicon Revisions

A	B	C					
X							

16.4. JTAG_TCK on IOSET 4 pin has a wrong configuration after boot

The JTAG_TCK signal on IOSET 4 shares its pin (PA22) with the clock signal of the following boot memory interfaces: SDMMC1, SPI1 IOSET 2, QSPI 0 IOSET 3.

If JTAG IOSET 4 is selected by the user as JTAG debug port in the Boot Configuration Word, and if the ROM Code boots, or tries to boot, on any of the external memory interfaces stated above, the JTAG clock pin (TCK) is reset at its default mode (PIO) at the end of the ROM Code execution.

This occurs as soon as EXT_MEM_BOOT_ENABLE is set.

Work around

Do not select or disable external memory boot interface SDMMC1, SPI1 IOSET 2 or QSPI0 IOSET 3. However, if using one of these boot interfaces is required, reconfigure the PA22 pin in JTAG TCK IOSET 4 mode in the bootstrap or application.

Affected Silicon Revisions

A	B	C					
X	X	X					

16.5. SDMMC0 and SDMMC1 boot issue

The card detect pin is not correctly sampled in the ROM code, which leads to a nondeterministic boot ability on the SDMMC0/SDMMC1 interfaces (SDCard or eMMC).

Work around

Use another boot media (e.g., serial flash) for the first level boot, and either deactivate the boot on SDMMC0/1 in the Boot Configuration Word in the Fuse area or remove any bootable program stored in the eMMC or SDCard connected to the chip at startup.

Affected Silicon Revisions

A	B	C					
	X						

16.6. UART blocks USB connection to SAM-BA Monitor

When a UART is used as the ROM Code console interface in the Boot Configuration Word, the USB Device connection may not be properly enabled, and thus the SAM-BA Monitor does not run.

Work around

Pull up the RX line of the UART.

Affected Silicon Revisions

A	B	C					
X	X	X					

16.7. Secure Boot Mode: AES-RSA X.509 Certificate Serial Number Length Limit

According to the standard RFC 5280 "Internet X.509 Public Key Infrastructure Certificate" section 4.1.2.2, the maximum length for serial numbers in X.509 certificates is 20 bytes.

When parsing the certificate chain in AES-RSA Secure Boot mode, the maximum serial number length allowed by the ROM code is 16 bytes.

Work Around

To use AES-RSA Secure Boot mode, do not use X.509 certificates with a serial number length higher than 16 bytes.

Affected Device Revisions

A	B	C					
X	X	X					

17. Secure Digital MultiMedia Card Controller (SDMMC)

17.1. Software 'Reset For all' command may not execute properly

The software 'Reset For All' command may not execute properly, and, as a result, some registers of the host controller may not reset properly. The setting of the different registers must be checked before reinitializing the SD card.

Work around

None

Affected Silicon Revisions

A	B	C					
X	X	X					

17.2. Status flag INTCLKS may not work correctly

When the SDMMC internal clock is disabled (SDMMC_CCR. INTCLKEN = 0) and reenabled after a few cycles (SDMMC_CCR. INTCLKEN = 1), the status flag INTCLKS may get stuck at 0.

Work around

A delay loop of 6 cycles minimum of the slowest clock (HCLOCK or BASECLK) must be inserted between SDMMC_CCR. INTCLKEN = 0 and SDMMC_CCR. INTCLKEN = 1.

Affected Silicon Revisions

A	B	C					
X							

17.3. Sampling clock tuning procedure

The sampling clock tuning procedure described in the “SD Host Controller Simplified Specification V3.00” may freeze in the latest verification of the “Wait until Buffer Read Ready” condition.

Work around

The condition “Check Execute Tuning = 0“ can be *OR’ed* to “Wait until Buffer Read Ready” condition in the loop issuing the *SEND_TUNING_BLOCK command (CMD19)*.

Affected Silicon Revisions

A	B	C					
X	X	X					

17.4. SDMMC I/O calibration does not work

The impedance calibration mechanism for the SDMMC I/Os does not work. A fixed calibration code is internally hard-wired and, as a result, the calibration procedure described in the device data sheet, section “ I/O Calibration”, has no effect.

The I/O impedance values are given in the section "Electrical Characteristics" of the data sheet.

With these values and careful PCB design (impedance controlled and matched traces), the SDMMC can be operated in HS200 and in SDR104 modes up to 120 MHz .

Work around

None

Affected Silicon Revisions

A	B	C					
X	X	X					

18. Secure Fuse Controller (SFC)

18.1. The Partial Fuse Masking function does not work

The fuse masking function described in section “Secure Fuse Controller (SFC)” does not work. If the ROM code is used in Secure mode, the overall fuses are masked by the ROM code even if some of them are not used.

Work around

None

Affected Silicon Revisions

A	B	C					
X							

18.2. The first two bits of each 32-bit block of the fuse matrix cannot be written

The first two bits of each 32-bit block of the fuse matrix cannot be written, so that any word (32 bits) written needs to set to 0 the first two bits of each word (32 bits) of the fuse matrix.

Work around

None

Affected Silicon Revisions

A	B	C					
X							

18.3. Fuse matrix programming requires a main clock (MAINCK) frequency between 10 and 15 MHz

If the main clock is not within the range of 10 to 15 MHz while programming the fuse matrix, the correct fuse programming cannot be ensured.

Work around

- If the main clock is out of the 10 to 15 MHz range, then before programming the fuse matrix switches the main clock to the internal 12 MHz RC oscillator.
- To program the fuses during ROM Code execution, SAM-BA/Secure and SAM-BA version 3.2.2 or higher must be used.

Affected Silicon Revisions

A	B	C					
X	X	X					

18.4. Fuse matrix read requires a main clock (MAINCK) frequency below 28 MHz

If the main clock is higher than 28 MHz, fuse matrix content read cannot be ensured.

Work around

Do not use the main oscillator in Bypass mode with a frequency higher than 28 MHz.

Affected Silicon Revisions

A	B	C					
X	X	X					

19. Special Function Registers (SFR)

19.1. The serial number stored in the SFR registers (SFR_SN0 and SFR_SN1) is not correct

The serial number (SFR_SN0, SFR_SN1) has only 16 bits set. This serial number cannot be used as a 64-bit unique ID.

Work around

None

Affected Silicon Revisions

A	B	C					
X							

20. Synchronous Serial Controller (SSC)

20.1. Unexpected delay on TD output

When SSC is configured with the following conditions:

- RCMR.START = Start on falling edge/Start on Rising edge/Start on any edge,
- RFMR.FSOS = None (input),
- TCMR.START = Receive Start,

an unexpected delay of 2 or 3 system clock cycles is added to the TD output.

Work around

None

Affected Silicon Revisions

A	B	C					
X	X	X					

21. Two-wire Interface (TWIHS)

21.1. The TWI/TWIHS Clear command does not work

Bus reset using the “CLEAR” bit of the TWI/TWIHS control register does not work correctly during a bus busy state.

Work around

When the TWI master detects the SDA line stuck in low state the procedure to recover is:

1. Reconfigure the SDA/SCL lines as PIO.
2. Try to assert a Logic 1 on the SDA line (PIO output = 1).
3. Read the SDA line state. If the PIO state is a Logic 0, then generate a clock pulse on SCL (1-0-1 transition).
4. Read the SDA line state. If the SDA line = 0, go to Step 3; if SDA = 1, go to Step 5.
5. Generate a STOP condition.
6. Reconfigure SDA/SCL PIOs as peripheral.

Affected Silicon Revisions

A	B	C					
X	X	X					

22. USB High-Speed Inter-Chip Port (HSIC)

22.1. At HSIC startup, the strobe default state is wrong

The strobe line should be at logic state 0 when HSIC is powered ON, and disabled. Currently, powering up the product sets the strobe line at logic state 1 before the HSIC is enabled. In this case, a connected device tries to connect before the HSIC is enabled.

Work around

Connect the device after the SAMA5D2 has been started.

Affected Silicon Revisions

A	B	C					
X							

23. Watchdog Timer (WDT)

23.1. Restart command of WDT may reset the DDR controller

When using the WDT window with WDD and WDV field of the WDT:

- if 0<WDD<WDV
- and the WDT is restarted in the permitted window 0<wdt counter<WDD

then the WDT is restarted but a reset signal is sent to the fuse controller and the DDR controller, leading to DDR memory access and/or fuse access issues.

Work around

None. Do not use the window mode of the WDT.

Affected Silicon Revisions

A	B	C					
X	X	X					

24. Data Sheet Clarifications

The following typographic corrections and clarifications are to be noted for the latest version of the device data sheet available on www.microchip.com:

Note: Corrections in tables, registers, and text are shown in **bold**. Where possible, the original bold text formatting has been removed for clarity.

No clarifications to report at this time.

25. Revision History

25.1. Rev. J - 08/2025

Note: The Revision DS80000803I of this data sheet was not issued.

Added content related to the SAMA5D27(T)-V/8GW(-SL3) device

25.2. Rev. H - 08/2022

Updated Silicon Issue Summary.

Added in Controller Area Network (MCAN):

- Frame transmitted despite confirmed transmit cancellation for CAN-FD messages with more than 8 data bytes

Added in Secure Digital MultiMedia Card Controller (SDMMC):

- SDMMC I/O calibration does not work

25.3. Rev. G - 03/2022

Updated Silicon Issue Summary.

Deleted from Controller Area Network (MCAN):

- Message order inversion when transmitting from dedicated Tx Buffers configured with same message ID.

25.4. Rev. F - 09/2021

Updated Silicon Issue Summary.

Added in Controller Area Network (MCAN):

- Debug message handling state machine not reset to Idle state when CCCR.INIT is set
- Message order inversion when transmitting from dedicated Tx Buffers configured with same message ID.

Added in Watchdog Timer (WDT):

- Restart command of WDT may reset the DDR controller

25.5. Rev. E - 03/2021

Updated Table 1. SAMA5D2x Silicon Device Identification.

Updated Silicon Issue Summary.

Added in Ethernet MAC (GMAC):

- Screening registers not working

25.6. Rev. D - 09/2020

Added in Peripheral Touch Controller (PTC):

- Wrong pull-up value on PD[18:3] during reset

Added in ROM Code:

- Secure Boot Mode: AES-RSA X.509 Certificate Serial Number Length Limit

25.7. Rev. C - 02/2020

Updated all errata in Controller Area Network (MCAN).

Added in Real-Time Clock (RTC):

- RTC_SR.TDERR flag is stuck at 0
- Read access truncated to the first 24 bits for register RTC_TIMALR (UTC_MODE)

Added in ROM Code:

- UART blocks USB connection to SAM-BA Monitor

Updated Data Sheet Clarifications.

25.8. Rev. B - 02/2019

Added - Fault Protection to Hi-Z for PWMx output not functional - Sampling clock tuning procedure
Updated Data Sheet Clarifications.

25.9. Rev. A - 10/2018

This is the initial released version of this document.

Microchip Information

Trademarks

ISBN: 979-8-3371-1690-7

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Product Page Links

ATSAMA5D21, ATSAMA5D22, ATSAMA5D23, ATSAMA5D24, ATSAMA5D26, ATSAMA5D27, ATSAMA5D28