ESP32-S3

Series SoC Errata Version 1.2

ESPRESSIF

Table of contents

1 Chip Revision Identification

Espressif is introducing a new **vM.X** numbering scheme to indicate chip revisions. This guide outlines the structure of this scheme and provides information on chip errata and additional identification methods.

1.1 Chip Revision Numbering Scheme

The new numbering scheme **vM.X** consists of the major and minor numbers described below.

M –Major number, indicating the major revision of the chip product. If this number changes, it means the software used for the previous version of the product is incompatible with the new product, and the software version shall be upgraded for the use of the new product.

X –Minor number, indicating the minor revision of the chip product. If this number changes, it means the software used for the previous version of the product is compatible with the new product, and there is no need to upgrade the software.

The **vM.X** scheme replaces previously used chip revision schemes, including ECOx numbers, Vxxx, and other formats if any.

1.2 Primary Identification Methods

eFuse Bits

The chip revision is encoded using two eFuse fields:

- EFUSE_RD_MAC_SPI_SYS_5_REG[25:23]
- EFUSE_RD_MAC_SPI_SYS_3_REG[20:18]

Table 1.1: Chip Revision Identification by eFuse Bits

	eFuse Bit	Chip Revision		
		v0.0	v0.1	v0.2
Major Number	EFUSE_RD_MAC_SPI_SYS_5_REG[25]	0	0	0
	EFUSE_RD_MAC_SPI_SYS_5_REG[24]	0	0	0
Minor Number	EFUSE_RD_MAC_SPI_SYS_5_REG[23]	0	0	0
	EFUSE_RD_MAC_SPI_SYS_3_REG[20]	0	0	0
	EFUSE_RD_MAC_SPI_SYS_3_REG[19]	0	0	1
	EFUSE_RD_MAC_SPI_SYS_3_REG[18]	0	1	0

Chip Marking

- **Espressif Tracking Information** line in chip marking

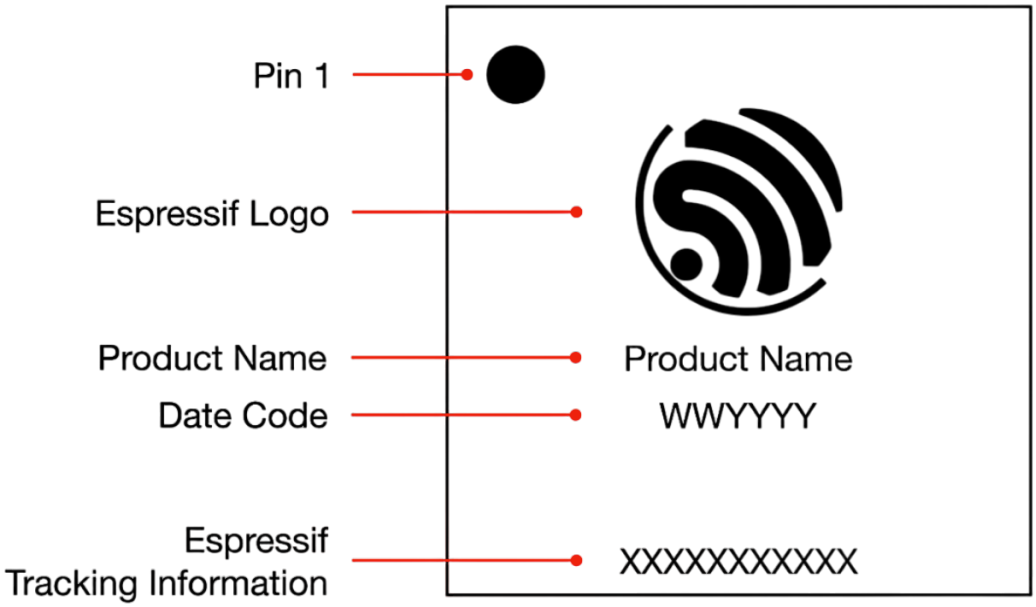

Figure 1.1: Chip Marking Diagram

Table 1.2: Chip Revision Identification by Chip Marking

Chip Revision	Espressif Tracking Information
v0.0	X A XXXXXX
v0.1	X B XXXXXX
v0.2	X C XXXXXX

Module Marking

- **Specification Identifier** line in module marking

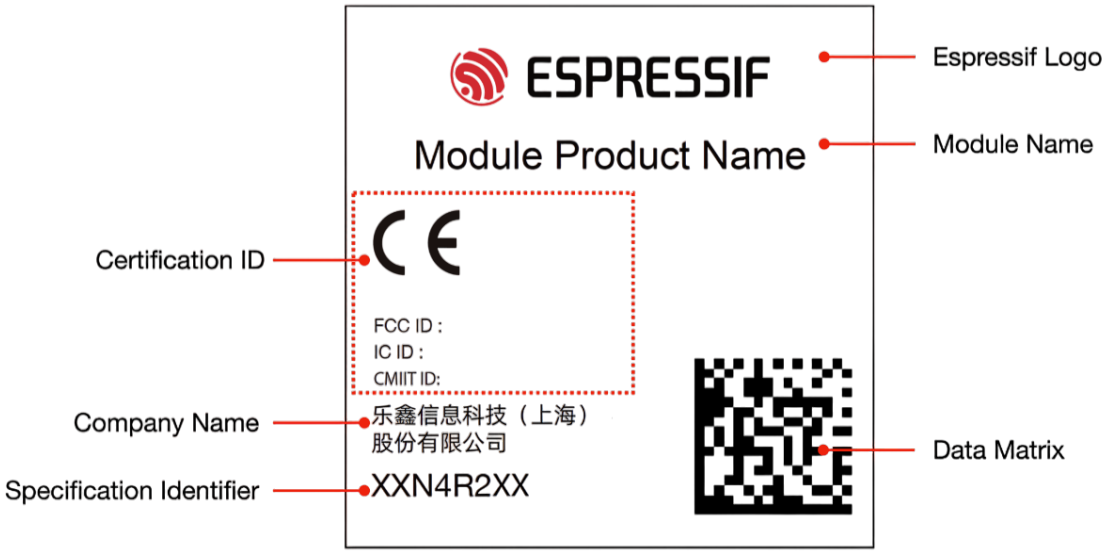

Figure 1.2: Module Marking Diagram

Table 1.3: Chip Revision Identification by Module Marking

Chip Revision	Specification Identifier
v0.0	—[1]
v0.1	M0 XXXX
v0.2	MC XXXX

[1] Missing specification identifier "—" means modules with this chip revision are not mass produced.

1.3 Additional Identification Methods

Date Code

Some errors in the chip product don't need to be fixed at the silicon level, or in other words in a new chip revision.

In this case, the chip may be identified by **Date Code** in chip marking (see *Chip Marking Diagram*). For more information, please refer to Espressif Chip Packaging Information.

PW Number

Modules built around the chip may be identified by **PW Number** in product label (see *Module Product Label*). For more information, please refer to Espressif Module Packaging Information.

Figure 1.3: Module Product Label

Note: Please note that **PW Number** is only provided for reels packaged in aluminum moisture barrier bags (MBB).

1.4 ESP-IDF Release Compatibility

Information about ESP-IDF release that supports a specific chip revision is provided in Compatibility Between ESP-IDF Releases and Revisions of Espressif SoCs.

1.5 Related Documents

- For more information about the chip revision upgrade and their identification of series products, please refer to ESP32-S3 Product/Process Change Notifications (PCN).
- For more information about the chip revision numbering scheme, see Compatibility Advisory for Chip Revision Numbering Scheme.

2 Errata Summary

Table 2.1: Errata summary

Category	Descriptions	Affected Revisions [1]		
		v0.0	v0.1	v0.2
RTC	*[RTC] RTC Register Read Error After Wake-up from Light-sleep Mode*	Y	Y	Y
Analog Power	*[Analog Power] Chip Will Be Damaged When BIAS_SLEEP = 0 and PD_CUR = 1*	Y	Y	Y
LCD	*[LCD] The LCD Module Exhibits Unreliable Behavior When Certain Clock Dividers Are Used*	Y	Y	Y
USB-OTG	*[USB-OTG] The USB-OTG Download Function Is Unavailable*	Y	Y	Y*
RMT	*[RMT] The Idle State Signal Level Might Run into Error in RMT Continuous TX Mode*	Y	Y	Y
Touch Sensor	*[Touch Sensor] The TOUCH_SCAN_DONE_INT Interrupt Raw Data Value Is Undefined*	Y	Y	Y
SAR ADC	*[SAR ADC] The Digital Controller (DMA) of SAR ADC2 Cannot Work*	Y	Y	Y

[1] Y* means some batches of a revision are affected.

3 All Errata Descriptions

3.1 [RTC] RTC Register Read Error After Wake-up from Light-sleep Mode

Affected revisions: v0.0 v0.1 v0.2

Description

If an RTC peripheral is turned off in Light-sleep mode, there is a certain probability that after waking up from Light-sleep, the CPU of ESP32-S3 will read the registers in the RTC power domain incorrectly.

Workarounds

Users are suggested not to power down RTC peripherals in Light-sleep mode. There will be no impact on power consumption.

This issue has been bypassed in ESP-IDF v4.4 and above.

Solution

No fix scheduled.

3.2 [Analog Power] Chip Will Be Damaged When BIAS_SLEEP = 0 and PD_CUR = 1

Affected revisions: v0.0 v0.1 v0.2

Description

If the analog power is configured as BIAS_SLEEP = 0 and PD_CUR = 1, the chip will be permanently damaged. This issue might be triggered when ULP and/or touch sensor is used during Light-sleep or Deep-sleep.

Workarounds

Users are suggested to disable such analog power configuration in sleep mode through software.

This issue has been bypassed by disabling the above configuration in ESP-IDF v4.4.2+, v5.0 and above.

Solution

No fix scheduled.

3.3 [LCD] The LCD Module Exhibits Unreliable Behavior When Certain Clock Dividers Are Used

Affected revisions: v0.0 v0.1 v0.2

Description

1. When the RGB format is used, if the clock divider is set to 1, i.e., LCD_CAM_LCD_CLK_EQU_SYSCLK = 1:

- The pixel clock output (LCD_PCLK) will not be able to be set to falling edge trigger.
- When frames are continuously sent in this mode (i.e., LCD_CAM_LCD_NEXT_FRAME_EN = 1), it might occur that the second frame inserts the last data of the previous frame in the first frame.

2. When the I8080 format is used, if the clock cycle of the LCD core clock (LCD_CLK) before data transmission is less than or equal to 2, it can result in incorrect value of the first data and the subsequent data quantity.

   Note: Please refer to the following steps to obtain the clock cycle before data transmission with the I8080 format.

   The clock cycle before data transmission depends on the following factors:

   - VFK cycle length (unit: LCD_PCLK): The clock cycle length during the VFK phase
   - CMD cycle length (unit: LCD_PCLK): The clock cycle length during the CMD phase
   - DUMMY cycle length (unit: LCD_PCLK): The clock cycle length during the DUMMY phase
   - LCD_CAM_LCD_CLK_EQU_SYSCLK: Decides if LCD_PCLK equals LCD_CLK
   - LCD_CAM_LCD_CLKCNT_N: Decides the division relationship between LCD_PCLK and LCD_CLK

   Based on the information above, three variables are defined below:

   - **total_pixels** = VFK cycle length + CMD cycle length + DUMMY cycle length
   - **cycle_unit** =
     - 1, if LCD_CAM_LCD_CLK_EQU_SYSCLK = 1
     - LCD_CAM_LCD_CLKCNT_N + 1, if LCD_CAM_LCD_CLK_EQU_SYSCLK = 0
   - **ahead_cycle** = **total_pixels** * **cycle_unit**

   ahead_cycle indicates the clock cycle before data transmission, which, if less than or equal to 2, will cause an error.

Workarounds

Users are suggested to do the followings:

- When using the RGB format, avoid configuring LCD_CAM_LCD_CLK_EQU_SYSCLK as 1.
- When using the I8080 format:
  - try to avoid configuring LCD_CAM_LCD_CLK_EQU_SYSCLK as 1.
  - ensure that **ahead_cycle** is larger than 2 if LCD_CAM_LCD_CLK_EQU_SYSCLK has to be set as 1.

This issue has been bypassed through the methods described above in ESP-IDF v4.4.5+, v5.0.3+, v5.1 and above.

Solution

No fix scheduled.

3.4 [USB-OTG] The USB-OTG Download Function Is Unavailable

Affected revisions: v0.0 v0.1 v0.2

Description

For ESP32-S3 series chips manufactured before the Date Code 2219 and series of modules and development boards with the PW Number before PW-2022-06-XXXX, the EFUSE_DIS_USB_OTG_DOWNLOAD_MODE (BLK0 B19[7]) bit of eFuse is set by default and cannot be modified. Therefore, the USB-OTG Download function is unavailable for these products.

Note: For detailed information about the Date Code and the PW Number, please refer to *Chip Revision Identification*.

Workarounds

ESP32-S3 also supports downloading firmware through USB-Serial-JTAG. Please refer to USB Serial/JTAG Controller Console.

Solution

This issue has been fixed in some batches of chip revision v0.2.

For ESP32-S3 series chips manufactured on and after the Date Code 2219 and ESP32-S3 series modules and development boards with the PW Number of and after PW-2022-06-XXXX, the bit (BLK0 B19[7]) will not be programmed by default and thus is open for users to program. This will enable the USB-OTG Download function.

For more details and recommendations for users, please refer to Security Advisory for USB_OTG & USB_Serial_JTAG Download Functions of ESP32-S3 Series Products.

3.5 [RMT] The Idle State Signal Level Might Run into Error in RMT Continuous TX Mode

Affected revisions: v0.0 v0.1 v0.2

Description

In ESP32-S3’s RMT module, if the continuous TX mode is enabled, it is expected that the data transmission stops after the data is sent for RMT_TX_LOOP_NUM_CHn rounds, and after that, the signal level in idle state should be controlled by the “level” field of the end-marker.

However, in real situation, after the data transmission stops, the channel’s idle state signal level is not controlled by the “level” field of the end-marker, but by the level in the data wrapped back, which is indeterminate.

Workarounds

Users are suggested to set RMT_IDLE_OUT_EN_CHn to 1 to only use registers to control the idle level.

This issue has been bypassed since the first ESP-IDF version that supports continuous TX mode (v5.0). In these versions of ESP-IDF, it is configured that the idle level can only be controlled by registers.

Solution

No fix scheduled.

3.6 [Touch Sensor] The TOUCH_SCAN_DONE_INT Interrupt Raw Data Value Is Undefined

Affected revisions: v0.0 v0.1 v0.2

Description

For ESP32-S3 ' s touch sensor, the raw data value is undefined for the first two TOUCH_SCAN_DONE_INT interrupts.

Workarounds

Users are suggested to skip the first two TOUCH_SCAN_DONE_INT interrupts, then turn them off and stop using them.

Solution

No fix scheduled.

3.7 [SAR ADC] The Digital Controller (DMA) of SAR ADC2 Cannot Work

Affected revisions: v0.0 v0.1 v0.2

Description

The Digital Controller of SAR ADC2, i.e., DIG ADC2 controller, may receive a false sampling enable signal. In such a case, the controller will enter an inoperative state.

Workarounds

It is suggested to use RTC controller to control SAR ADC2.

Solution

No fix scheduled.

4 Revision History

Table 4.1: Revision History

Date	Version	Release Notes
2023-11-15	v1.2	•***Chip Revision Identification*** – Added information about how to identify chip revisions in modules – Added Section *Additional Identification Methods* •***All Errata Descriptions*** – Adjusted the section order – Added Section *[RTC] RTC Register Read Error After Wake-up from Light-sleep Mode* – Added Section *[LCD] The LCD Module Exhibits Unreliable Behavior When Certain Clock Dividers Are Used* – Added Section *[RMT] The Idle State Signal Level Might Run into Error in RMT Continuous TX Mode* – Added Section *[Touch Sensor] The TOUCH_SCAN_DONE_INT Interrupt Raw Data Value Is Undefined* • Other minor updates
2023-01-20	v1.1	Added Section *[USB-OTG] The USB-OTG Download Function Is Unavailable*
2022-10-14	v1.0	First release

5 Related Documentation and Resources

5.1 Related Documentation

- ESP32-S3 Datasheet –Specifications of the ESP32-S3 hardware.
- ESP32-S3 Technical Reference Manual –Detailed information on how to use the ESP32-S3 memory and peripherals.
- ESP32-S3 Hardware Design Guidelines –Guidelines on how to integrate the ESP32-S3 into your hardware product.
- Certificates
  https://espressif.com/en/support/documents/certificates
- ESP32-S3 Product/Process Change Notifications (PCN)
  https://espressif.com/en/support/documents/pcns?keys=ESP32-S3

- ESP32-S3 Advisories –Information on security, bugs, compatibility, component reliability.
  https://espressif.com/en/support/documents/advisories?keys=ESP32-S3
- Documentation Updates and Update Notification Subscription
  https://espressif.com/en/support/download/documents

5.2 Developer Zone

- ESP-IDF Programming Guide for ESP32-S3 –Extensive documentation for the ESP-IDF development framework.
- ESP-IDF and other development frameworks on GitHub.
  https://github.com/espressif
- ESP32 BBS Forum –Engineer-to-Engineer (E2E) Community for Espressif products where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
  https://esp32.com/
- The ESP Journal –Best Practices, Articles, and Notes from Espressif folks.
  https://blog.espressif.com/
- See the tabs SDKs and Demos, Apps, Tools, AT Firmware.
  https://espressif.com/en/support/download/sdks-demos

5.3 Products

- ESP32-S3 Series SoCs –Browse through all ESP32-S3 SoCs.
  https://espressif.com/en/products/socs?id=ESP32-S3
- ESP32-S3 Series Modules –Browse through all ESP32-S3-based modules.
  https://espressif.com/en/products/modules?id=ESP32-S3
- ESP32-S3 Series DevKits –Browse through all ESP32-S3-based devkits.
  https://espressif.com/en/products/devkits?id=ESP32-S3
- ESP Product Selector –Find an Espressif hardware product suitable for your needs by comparing or applying filters.
  https://products.espressif.com/#/product-selector

5.4 Contact Us

- See the tabs Sales Questions, Technical Enquiries, Circuit Schematic & PCB Design Review, Get Samples (Online stores), Become Our Supplier, Comments & Suggestions.
  https://espressif.com/en/contact-us/sales-questions