BRT_TN_006

BT88x Errata Technical Note

Version 1.0

Issue Date: 10-03-2024

The intention of this errata technical note is to document a detailed description of known functional or electrical issues with the BRTChip's BT88x series devices, including BT880, BT881, BT882, BT883.

Bridgetek Pte Ltd (BRTChip)
1 Tai Seng Avenue, Tower A, #03-05, Singapore 536464
Tel: +65 6547 4827
Web Site: http://www.brtchip.com

Table of Contents

1 BT88x Revision – Part Numbers

BT88x part numbers are listed in Table 1. The suffix letter following the date code on the package marking identifies the device revision.

Part Number and Revision	Description
BT880Q RevA	EVE with resistive touch, 18bit RGB interface
BT881Q RevA	EVE with capacitive touch, 18bit RGB interface
BT882Q RevA	EVE with resistive touch, 24bit RGB interface
BT883Q RevA	EVE with capacitive touch, 24bit RGB interface

Table 1 - BT88x Part Numbers

2 Errata History Table – Functional Issues

Functional Errata	Short Description	Errata occurs in device revision
1	Display list instruction over fetch	**Rev A**

Table 2 – Functional Issues

2.1 Errata History Table – Electrical and Timing Specification Deviations

Deviations	Short Description	Errata occurs in device revision
-	No known issues	**-**

Table 3 – Electrical and Timing Specification Deviations

3 Functional Issues of BT88x

3.1 Display list instruction overfetch

Introduction:

In the graphics engine, display list fetch is double-operation: the hardware fetches and decodes two 32-bit instructions per clock. When the **DISPLAY** instruction, which marks the end of display list, is placed in an even slot, the following instruction in the odd slot is erroneously fetched and can be partially executed by some bitmap pipeline stages. The table below shows the layout of **RAM_DL** content when the issue happens:

Instruction 1	Instruction 0
Instruction 3	Instruction 2
...	
Instruction 2N+1	Instruction 2N (**DISPLAY**)

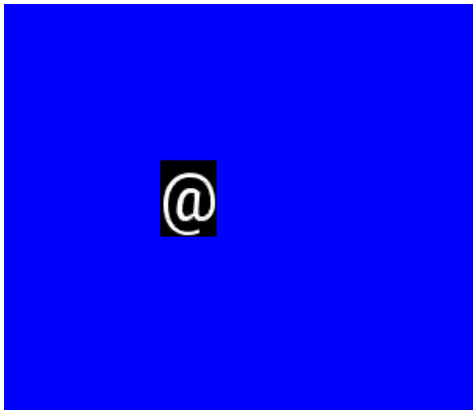
The specific problematic instruction is **BITMAP_SIZE**. If it occurs after the **DISPLAY** instruction, it is executed by the graphics engine, and erroneously sets the height parameter of current bitmap handle. Consequently, the rendered image associated with the impacted bitmap handle could be enlarged or shortened vertically.

Note that the problematic extra instruction BITMAP_SIZE could be either left over by an older display list using the same RAM_DL, or by chance the initial content of the RAM upon chip power on if that RAM_DL location is not used in any older display list.

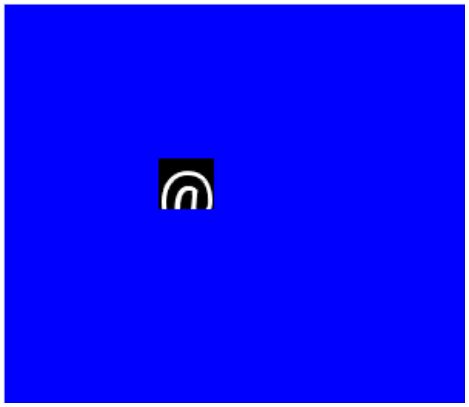
For example, given the following display list:

```
CLEAR_COLOR_RGB(0, 0, 254)                    // Instruction 0
CLEAR(1, 1, 1)                                // Instruction 1
BITMAP_HANDLE(31)                             // Instruction 2
BEGIN(BITMAPS)                                // Instruction 3
BLEND_FUNC(SRC_ALPHA, ZERO)                   // Instruction 4
VERTEX2II(100, 100, 31, '@')                  // Instruction 5
DISPLAY()                                     // Instruction 6
BITMAP_SIZE(NEAREST, BORDER, BORDER, 36, 33) // Instruction 7
```

The expected image is:



Since instruction 7 is partially executed, the height parameter of **BITMAP_SIZE** is set to 33:



Exception:

When the erroneously fetched instruction is not **BITMAP_SIZE**, there's no impact to the display content. When the **DISPLAY** instruction is placed in the odd slot, there's no impact to the display content.

Workaround:

The software workarounds are as follows:

Identify the end of display list and append it with an *additional* **DISPLAY** instruction, guaranteeing that the erroneous fetch (if any) does nothing. For example:

```
CLEAR_COLOR_RGB(0, 0, 254)                  // Instruction 0
CLEAR(1, 1, 1)                              // Instruction 1
BITMAP_HANDLE(31)                           // Instruction 2
BEGIN(BITMAPS)                              // Instruction 3
BLEND_FUNC(SRC_ALPHA, ZERO)                 // Instruction 4
VERTEX2II(100, 100, 31, '@')                // Instruction 5
DISPLAY()                                   // Instruction 6
DISPLAY()                                   // Instruction 7
```

At instruction 7, an additional **DISPLAY** instruction is added, which prevents the unexpected impact of erroneously fetched instruction.

However, when the following coprocessor commands are used, the workaround above is not applicable because these commands generate the DISPLAY instruction and swap the display list immediately:

- *CMD_CALIBRATE*
- *CMD_SPINNER*
- *CMD_LOGO*

In this scenario, the workaround is to reset **RAM_DL** to zero before initiating the display list, effectively populating **RAM_DL** with **DISPLAY** instructions. For example:

```
CMD_DLSTART()
CMD_MEMZERO(RAM_DL, 8192)
CLEAR(1,1,1)
CMD_TEXT(80, 30, 27, OPT_CENTER, "PLEASE TAP ON THE DOT")
CMD_CALIBRATE()
```

Coprocessor command **CMD_MEMZERO** is added to reset the **RAM_DL** to zero, which guarantees the display list is ended with at least two **DISPLAY** instructions.

4 Electrical and Timing Specification deviations of BT88x

There are no known electrical or timing problems with any revision of silicon.

5 Contact Information

Refer to https://brtchip.com/contact-us/ for contact information.

Distributor and Sales Representatives
Please visit the Sales Network page of the Bridgetek Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A – References

Document References

NA

Acronyms and Abbreviations

Terms	Description
EVE	Embedded Video Engine

Appendix B – List of Tables & Figures

List of Tables

List of Figures

NA

Appendix C – Revision History

Document Title: BRT_TN_006 BT88x Errata
Document Reference No.: BRT_000433
Clearance No.: BRT#209
Product Page: https://brtchip.com/product-category/products/
Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	10-03-2024

Internal Revision History

Internal Revision history (internal use only, please clearly state all changes here before saving the file)

Revision	Date DD-MM-YYYY	Changes	Editor
Draft 0.1	26-02-2024	Initial Draft	David Wang
Draft 0.2	27-02-2024	Reviewed, commented, edited: 1. Update the headquarter address 2. Reformat the code style to align with programming guide 3. Rename the opcode to instruction to align with programming guide 4. Replace the firmware with coprocessor commands to align with programming guide 5.Advise chapter 5 is not required	Paul Jiao
Draft 0.3	01-03-2024	Reviewed and updated with respect to formatting, Grammer	L Subramanian
Draft 0.3	01-03-2024	Reviewed, no comments	A Jones
Draft	04-03-2024	Reviewed, 1 comment added	G Brown
Draft	05-03-2024	Updated to the review comment	David Wang
Draft	05-03-2024	Update looks good, no further comments	G Brown
Draft	05-03-2024	Reviewed added comments	Gavin Moore
Draft	06-03-2024	Updated to review comments	David Wang
Draft	06-03-2024	Reviewed – Recommend for approval	Gavin Moore
1.0	10-03-2024	Approved LCE	L Subramanian