Technical Note

TN_159

FT90x Errata

Version 1.2

Issue Date: 15-12-2020

This errata technical note gives a detailed description of known functional or electrical issues with the FT90X series device.

The current revision of the FT90X series is **Revision B, released February 2015.**

Bridgetek Pte Ltd (BRTChip)
178 Paya Lebar Road, #07-03, Singapore 409030
Tel: +65 6547 4827 Fax: +65 6841 6071
Web Site: http://www.brtchip.com

Table of Contents

1 FT90x Revision

FT90x part numbers are listed in **Table 1.** The letter at the end of date code identifies the device revision, for example YYWW-B. See section **Error! Reference source not found.** for more details.

The current revision of the FT90x series is **revision B, released Feb 2015.** At the time of releasing this Technical Note there are known issues with this silicon revision.

Part Number	Package
FT900Q/FT901Q/FT902Q/FT903Q	100-pin QFN
FT900L/FT901L/FT902L/FT903L	100-pin LQFP
FT905Q/FT906Q/FT907Q/FT908Q	76-pin QFN
FT905L/FT906L/FT907L/FT908L	80-pin LQFP

Table 1 - FT90x Part Numbers

This errata technical note covers the revisions of FT90X listed in **Table 2**.

Revision	Notes
A	First device revision. Engineering Samples.
B	Second device revision. Launched Feb 2015
C	Third device revision. Launched 2019.

Table 2 - FT90x Series Revisions

2 Errata History Tables

2.1 Functional Errata

Functional Errata	Short description	Errata occurs in device revision
Timers	Software access of 16 bit timer count is not atomic	B
PWM	Changing PWM duty cycle in during operation may cause errors	B
ADC	FIFO Data Count is not accurate in some scenarios	B
DAC	FIFO Data Count must be interpreted along with DAC interrupt bit to detect FIFO full/empty	B
Interrupt Controller	Nested Interrupt Enable bit appears at different bit positions for read and write operations.	B
CAN	Transmit Error Count not increased	B
CAN	Wrong behaviour when an 'error passive' station transmits a PASSIVE ERROR FLAG	B
CAN	Wrong behaviour when sending an OVERLOAD FRAME	B
Ethernet	Limited Receive Buffer Size	B
Ethernet	100BASE-FX Support	B, C
USB Host	Overcurrent Protection	B
USB Slave	The USB Device Controller still updates the frame number after receiving a SOF package with PID/CRC5 error	B, C
USB Slave	Control transfer sends an unexpected NYET response in Status stage for High-speed mode	B, C
USB Slave	The USB Device Controller doesn't check the incoming packet in Setup stage	B, C
USB Slave	The USB Device Controller doesn't check for bus turn-around timeout between OUT/SETUP token and data phases	B, C
USB Slave	The USB Device Controller writes past the range of the data buffer when a babble error occurs	B, C
USB Slave	Error handling for the last data transaction differs from USB 2.0 Specification	B, C
UART	Back-to-back Reading of the Receive Buffer Register	B
I2S	TX FIFO Count Corruption during Underrun	B, C
I2S	Left/Right Samples are swapped after underrun	B, C
SPI Slave	TX FIFO Missing Data	B, C
SPI Slave	Data has to align to the RX FIFO size during reception	B, C

Table 3 - Functional Errata

2.2 Electrical and Timing Specification Deviations

Deviations	Short description	Errata occurs in device revision
FSOURCE pin issue	Doesn't meet the design specification.	B

Table 4 - Electrical and Timing Errata

3 Electrical Specification Deviation Errata of FT90x

3.1 FSOURCE Pin issue

Introduction:

The FT90x series Revision B has dedicated VPP and FSOURCE input pins to receive power for writing to the 64-bit EFUSE. During EFUSE writing, the typical voltages required by VPP and FSOURCE are +1.85V and +3.70V respectively. For EFUSE reading or in standby mode, keep the pins floating or tie them to ground.

Issue:

The FSOURCE pin has a potential ESD issue and is not bonded out to the package in the FT90x series Revision B.

Workaround:

There is no workaround for this revision. In FT90x series Revision C, the EFUSE has been replaced by an OTP flash; hence the pins are used to receive power for the internal PLL instead.

Hardware reference design:

1) For PCBs that are designed for Revision B, no change is needed.
2) For PCBs that are designed for Revision C, FSOURCE should be connected to the 1.2V regulator output of the chip and VPP should be connected to the 3.3V power.

Package Specific:

The affected packages are listed in **Table 5**. All are Revision B.

Package	Applicable (Yes/No)
FT900Q/FT901Q/FT902Q/FT903Q	Yes
FT900L/FT901L/FT902L/FT903L	Yes
FT905Q/FT906Q/FT907Q/FT908Q	Yes
FT905L/FT906L/FT907L/FT908L	Yes

Table 5 - Affected Packages

4 Functional Errata

4.1 Timers – Reading Timer value

Introduction:

The FT90X series has 16-bit timers that are accessed via two 8-bit registers. Since they are not latched, the user must be careful to handle overflows while determining the final 16 bit value.

Issue:

If either of the 8-bit registers overflows during the read, the resulting concatenated 16-bit value will be incorrect as illustrated in **Table 6**.

Timer Registers TMR_READ_MS : TMR_READ_LS	**Timer read sequence**	**Stored Timer Value** TimeH : TimeL
00FFh	Read TMR_READ_LS	xxxxh
0100h	Store TMR_READ_LS into TimeL	xxFFh
0101h	Read TMR_READ_MS	xxFFh
0102h	Store TMR_READ_MS into TimeH	01FFh

Table 6 - Timer low byte overflows during read

Workaround:

The workaround for this is to read the high byte twice to check if there was an overflow while the low byte was being read as shown in Error! Reference source not found..

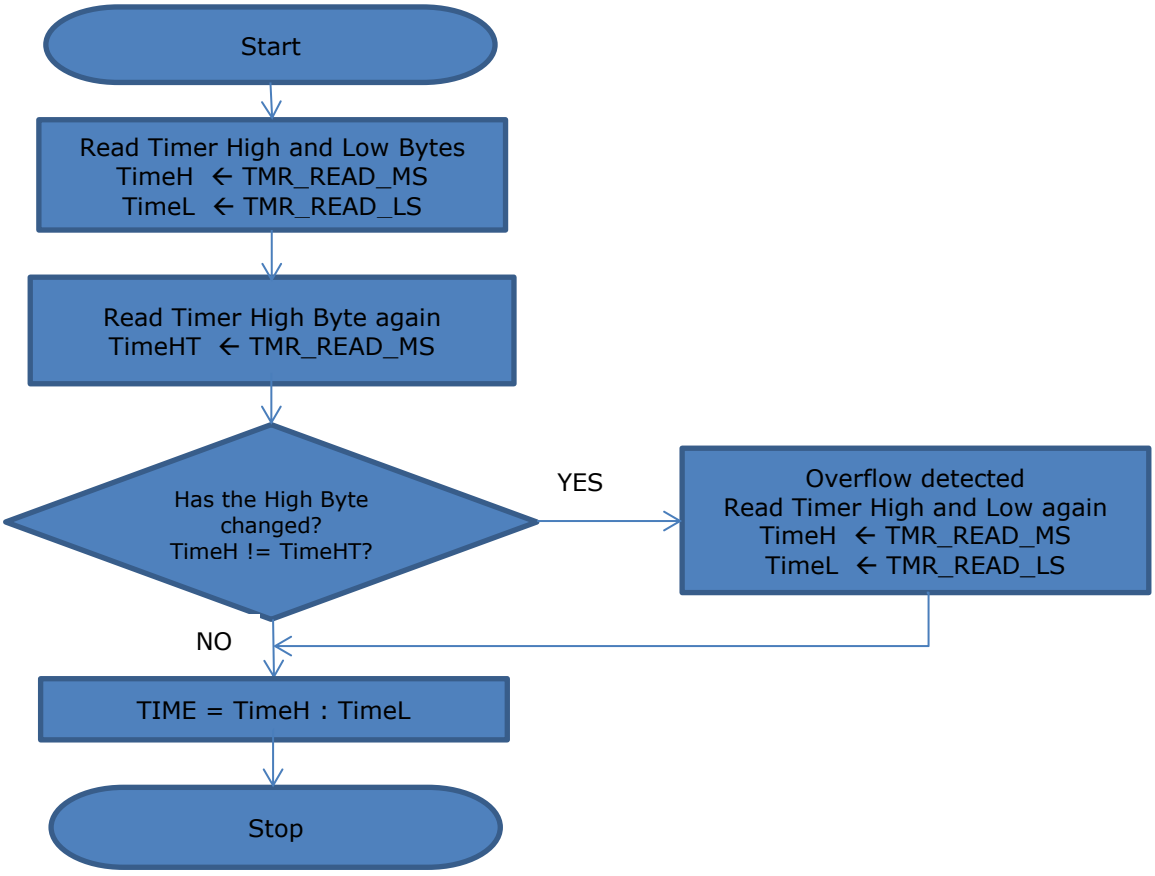

Figure 4.1 - Timer Read taking overflows into account

Note: Users must also take care that the above read sequence is not interrupted, as that could potentially lead to inconsistent values.

4.2 PWM – Changing duty cycle in runtime

Introduction:

The FT90X series PWM modules have one internal global 16 bit counter and a per channel 16 bit comparator. The value of the comparator register controls the duty cycle of the generated PWM wave. Since the comparator register is not buffered, the user must be careful while writing it at runtime to avoid glitches in the PWM output. Care must also be taken while writing the 16 bit registers as it can only be accessed in 8 bits each.

Issue:

1. If the comparator register is written while the PWM module is in operation the transition to the new duty cycle may have a glitch and could also invert the signal, depending on the value of the internal count register when the comparator is updated. This occurs because the counter used by the PWM module is a free running counter which is not synchronized with the PWM comparator register. An example case is shown in Table 7.

PWM Internal Counter	PWM Comparator value	PWM Pin State	Comments
0000h	3FFFh	1	A 25% duty cycle wave.
0001h	3FFFh	1	
…	…		
3FFFh	3FFFh	0	
4000h	3FFFh	0	
…	…		
FFFFh	3FFFh	1	
…	…	…	
BFFEh	BFFFh	0	Software changes comparator to BFFFh to generate a 75% duty cycle wave while counter was some xxxxh (where 3FFFh < xxxxh < BFFFh). Signal toggles again when counter reaches BFFFh. Therefore the output signal toggles twice in the same count cycle, generating a glitch and reversing the polarity of the output signal.
BFFFh	BFFFh	1	
C000h	BFFFh	1	
…	…		
FFFFh	BFFFh	0	
0000h	BFFFh	0	

Table 7 - Changing duty cycle of PWM at run-time may cause glitches

2. Since the Comparator register can only be accessed as 8 bit, changing the count by writing the low byte first could also cause false matches and unexpected glitches depending on the state of the internal counter.

Workaround:

1. For PWM operating in continuous mode, the PWM module must be stopped before the duty cycle is updated.
2. For PWM operating in one-shot mode (up to 255 shots), the new comparator value can be loaded in the PWM Interrupt.

4.3 ADC – FIFO Data Count

Introduction:

The FT90X series ADC module has 128 sample FIFO. The ADC_DATA_COUNT bits (DAC_ADC_CNT[23:16]) indicate the amount of samples available for reading the FIFO at the most recent interrupt. The user is recommended to not use this counter as it is incorrect in some scenarios.

Issue:

The ADC_DATA_COUNT bits indicate the number of ADC samples available to read in the FIFO. Note that the actual number of samples available to read is one more than the ADC_DATA_COUNT value. Counts of 7Eh (126) and 7Fh (127) appear as FEh and FFh. In case of FIFO overflow, the counter will also overflow and will not reflect the current data availability. It is therefore unable to unambiguously indicate FIFO overflow.

Workaround:

The system should be designed to not use the ADC_DATA_COUNT register. The ADC Module generates an interrupt after every conversion of 64 samples. This interrupt service routine could be written to read all converted 64 samples immediately. If required, software can also increment a counter in the ADC interrupt service routine to track how many samples are available to read at multiples of 64.

4.4 DAC – FIFO Data Count

Introduction:

The FT90X series DAC modules each have a 128 sample FIFO. The DAC_DATA_COUNTx bits (DAC_ADC_COUNT[0:7] and DAC_ADC_COUNT[8:15]) indicate the number of samples in the FIFO that are yet to be converted. The user must be careful while using this counter as it does not a count of 126/127 correctly. And to distinguish between a buffer full and empty condition, the user must check the counter along with the interrupt flag.

Issue:

The DAC_DATA_COUNTx bits indicate the number of DAC samples that are yet to be converted. Note that the number of samples in the FIFO is one more than the DAC_DATA_COUNTx value. Counts of 7Eh (126) and 7Fh (127) appear as FEh and FFh. A DAC_DATA_COUNTx value of 0xFFh could either indicate a buffer full or empty condition, as shown in **Table 8**.

Samples still present in the FIFO	DAC_DATA_COUNTx value	DAC_IRQ_PENDx bit [Set on FIFO empty and on 64 sample conversion]
00h (empty)	FFh	1
01h	00h	
02h	01h	
…	…	
7Eh	7Dh	
7Fh	FEh	
80h (full)	FFh	0

Table 8 - Behaviour of DAC data count

Workaround:

1. DAC_DATA_COUNTx values of FEh and FFh should be read as 7Eh and 7Fh.
2. To distinguish between FIFO full and empty, check the DAC_IRQ_PENDx bit (DAC_ADC_CONF[25] for DAC0 and DAC_ADC_CONF[26] for DAC1) in addition to the DAC_DATA_COUNTx count.
    - If DAC_IRQ_PENDx is asserted while DAC_DATA_COUNTx is FFh, it means the FIFO is empty
    - If DAC_IRQ_PENDx is deasserted while DAC_DATA_COUNTx is FFh, it means the FIFO is full

Note: The DAC interrupt is generated whenever 64 samples have been converted AND when the DAC FIFO is empty.

4.5 Interrupt Controller -Read of Nested Interrupt Enable bit

Introduction:

The FT90X series supports nested interrupts via a bit configured in the IRQ Control Register. The bit position is at 7th bit for writes and 6th bit for reads.

Issue:

The Nested Interrupt Enable bit in the IRQ Control register appears at different positions depending on whether the register is accessed in Read or Write mode. Therefore the user must take care that writing is done on the 7th bit and read operations such as bit test are on the 6th bit.

Workaround:

User should follow the example in to set and test the Nested Interrupt Enable bit. Care must be taken when using test-and-set code to modify this bit.

SET the Nested Interrupt Enable bit CLEAR the Nested Interrupt Enable bit	INTERRUPT->IRQ_CTRL |= (0x01 << 7); INTERRUPT->IRQ_CTRL &= ~(0x01 << 7);

TEST the Nested Interrupt Enable bit	if((INTERRUPT->IRQ_CTRL & (0x01 << 6)) == (0x01 << 6))
	{
	  // Bit is set
	}
	else
	{
	  // Bit is cleared
	}

Table 9 - Pseudo code to test and set the Nested Interrupt Enable bit

4.6 CAN

4.6.1 Transmit Error Count

Introduction:

According to the CAN Specification Version 2.0, when a TRANSMITTER sends an ERROR FLAG, the TRANSMIT ERROR COUNT is increased by 8. However, it is not increased under two exceptions. Only Exception 1 is related to the issue.

Exception 1: If the TRANSMITTER is 'error passive' and detects and ACKNOWLEDGEMENT ERROR because of not detecting a 'dominant' ACK and does not detect a 'dominant' bit while sending its PASSIVE ERROR FLAG.

Issue:

When exception 1 does not occur, i.e.:

- The TRANSMITTER is 'error passive' and,
- The TRANSMITTER detects and ACKNOWLEDEMENT ERROR and,
- The TRASMITTER ***does*** detect a 'dominant' bit while sending its PASSIVE ERROR FLAG.

The TRANSMIT ERROR COUNT should only be increased (by 8) once. Currently, it is increased six times after each sample point. As a result, the count will reach 256 very quickly and the device enters BUS OFF state.

Workaround:

Currently, there is no workaround for this issue. The user has to reset the CAN controller to refresh the error counter if 'bus off' condition occurs.

4.6.2 Wrong behaviour when an 'error passive' station transmits a PASSIVE ERROR FLAG

Introduction:

According to the CAN Specification Version 2.0, an 'error passive' station detecting an error condition tries to signal this by transmission of a PASSIVE ERROR FLAG. *The 'error passive' station waits for six consecutive bits of equal polarity, beginning at the start of the PASSIVE EERROR FLAG*. The PASSIVE ERROR FLAG is complete when these 6 equal bits have been detected.

Issue:

The CAN controller violates the behavior described in the specification. Instead of checking the bus at the start of the PASSIVE ERROR FLAG, it checks the bus at the end of the PASSIVE ERROR FLAG.

Workaround:

Currently, there is no workaround for this issue.

4.6.3 Wrong behaviour when sending an OVERLOAD FRAME

Introduction:

According to the CAN Specification Version 2.0, there are three kinds of OVERLOAD conditions. Only condition 1 is related to this issue.

Condition 1: An OVERLOAD FRAME is sent due to the internal conditions of a receiver, which requires a delay of the next DATA FRAME or REMOTE FRAME.

The start of an OVERLOAD FRAME due to condition 1 is only allowed to be started at the first bit time of an expected INTERMISSION.

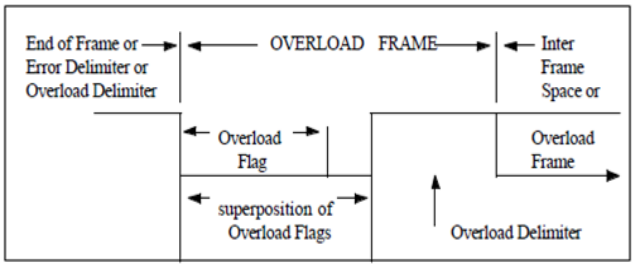

Figure 4.2 - CAN OVERLOAD Frame

Issue:

The CAN controller violates the behavior described in the specification. Instead of sending the OVERLOAD FRAME immediately after the End of Frame, the CAN controller adds one extra 'recessive' bit (1'b1) between the End of Frame and the OVERLOAD FRAME.

Workaround:

Currently, there is no workaround for this issue.

4.7 Ethernet

4.7.1 Limited Receive Buffer Size

Introduction:

The receive buffer is used to receive incoming Ethernet data for the software to process.

Issue:

The receive buffer size is currently limited to 2KB. This may cause data loss if multiple big Ethernet packets arrive too quickly and the software cannot clear the buffer fast enough. For example, if the user pings an FT90x from a PC with a data size of 3KB, the data will be broken into 2 packets, each with a size of about 1.5KB (the maximum Ethernet packet size). With a receive buffer size of 2KB, the FT90x can only receive one packet at a time. If the FT90x software doesn't finish processing the first packet before the second one arrives, it will be dropped, which results in a packet loss.

Workaround:

Currently, there is no workaround for this issue.

4.7.2 100BASE-FX Support

Introduction:

100BASE-FX is a version of Fast Ethernet over optical fiber.

Issue:

The Ethernet module in the FT90x MCUs does not support 100BASE-FX. Page 70 of "AN_324 FT900 User Manual" mistakenly states that 100BASE-FX is supported.

Workaround:

Currently, there is no workaround for this issue.

4.8 USB Host - Overcurrent Protection

Introduction:

The FT90x comes with overcurrent protection for USB Host. It can automatically cut off VBUS if an overcurrent is detected.

Issue:

After VBUS is cut off, overcurrent no longer occurs. The internal circuit will automatically turn on VBUS (and causes overcurrent again). This results in an infinite loop of cutting off/turning on VBUS when overcurrent is detected and can potentially damage the FT90x. There is no way to control VBUS from software.

Workaround:

Currently, there is no software workaround for this issue. However, the user may choose to use a switch, e.g. TPS2042B, to control the VBUS supply to the system. Any GPIO pin (other than GPIO 1, which is used of overcurrent detection) can be used to control the switch as follows:

- Set GPIO 1 to OCN function.
- Set another GPIO to GPIO function to control the switch.

- Enable overcurrent detection and interrupt by setting the OC_DETECT_EN and OC_DETECT bits in the PMCFG register.
- When USB host is used normally, control the switch to enable VBUS.
- When overcurrent occurs and triggers an interrupt, control the switch to disable VBUS until the problem is solved or the CPU is restarted.

4.9 USB Device

4.9.1 The USB Device Controller still updates the frame number after receiving a SOF package with PID/CRC5 error

Introduction:

When the USB Device Controller receives a SOF package with PID/CRC5 error, it should discard the packet and should not update the frame number in the DC_FRAME_NUMBER_LSB register (offset 0x14).

Issue:

After receiving the error packet, the USB Device Controller still update the frame number in the DC_FRAME_NUMBER_LSB register.

Workaround:

Currently, there is no workaround for this issue.

4.9.2 Control transfer sends an unexpected NYET response in Status stage for High-speed mode

Introduction:

According to the USB 2.0 Specification, section 8.5.3.1, there are 3 types of response packet for a Control transfer, namely ACK, STALL and NAK.

Issue:

The USB Device Controller sends a NYET response to the Host under the following scenario:

- Host sends an IN request to Device
- In Status stage, Host retries an OUT request (assume that Device responded with a corrupted ACK)
- Device sends a NYET packet to host (it should re-sends an ACK packet instead)

Host	Device
SETUP	
DATA0	
	ACK
IN	
	DATA1
ACK	
OUT	
DATA1	
	ACK
OUT	

Figure 4.3 - NYET packet response from the USB Device Controller

Workaround:

Currently, there is no workaround for this issue.

4.9.3 The USB Device Controller doesn't check the incoming packet in Setup stage

Introduction:

In Setup stage, if the USB Device Controller receives an invalid DATA0 packet, it should discard the data and should not respond with any packet. The Host would timeout and resend the SETUP packet.

Issue:

After receiving the invalid DATA0 packet, the USB Device Controller still sends and ACK response to the Host.

Workaround:

Currently, there is no workaround for this issue.

4.9.4 The USB Device Controller doesn't check for bus turn-around timeout between OUT/SETUP token and data phases

Introduction:

According to the USB 2.0 Specification, section 8.7.2, the device uses its bus turn-around timer between token and data or data and handshake phases. The host uses its timer between data and handshake or token and data phases.

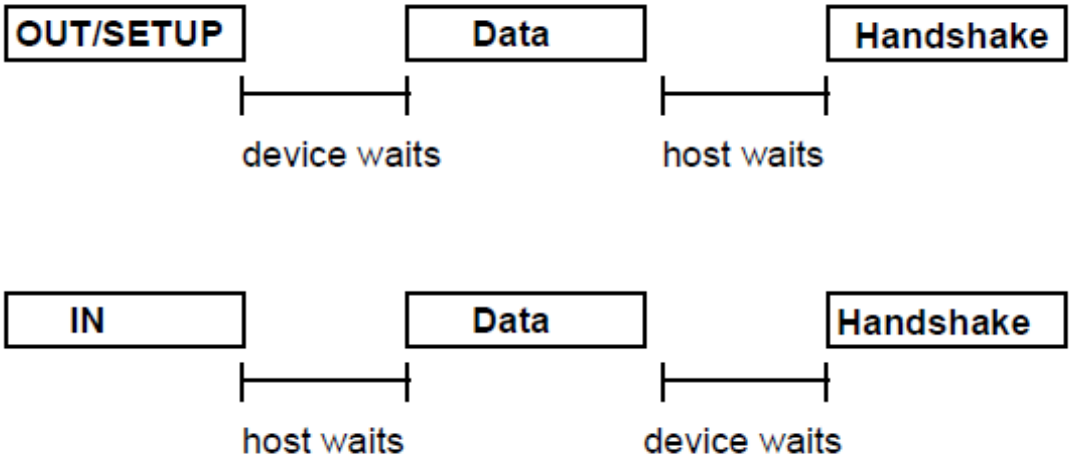

Figure 4.4 - Bus Turn-around Timer Usage (copied from USB 2.0 Specification)

Issue:

The USB Device Controller doesn't check for bus turn-around timeout between OUT/SETUP token and data phases. It will accept the data and respond with ACK.

Workaround:

Currently, there is no workaround for this issue.

4.9.5 The USB Device Controller writes past the range of the data buffer when a babble error occurs

Introduction:

A babble error occurs when USB device receives more data than the maximum packet size.

Issue:

If the data packet comes with the correct CRC16, the USB Device Controller accepts it and responds with ACK. It then writes the data past the address boundary of the data buffer of the endpoint.

If the data packet comes with the incorrect CRC16, the USB Device Controller discards it and times out. However, it then still writes the data past the address boundary of the endpoint.

Workaround:

Currently, there is no workaround for this issue.

4.9.6 Error handling for the last data transaction differs from USB 2.0 Specification

Introduction:

According to the USB 2.0 Specification, section 8.5.5.3, if the ACK handshake is corrupted on the last IN transaction of a Data stage, the function can interpret the start of the Status stage as verification that the host successfully received the data and respond with ACK.

Issue:

The USB Device Controller responds with STALL (instead of ACK) in the Status stage if the ACK on the last IN transaction of the Data stage is corrupted.

Workaround:

Currently, there is no workaround for this issue.

4.10 UART - Back-to-back Reading of the Receive Buffer Register

Introduction:

The FT90x UART provides a FIFO for receiving data of up to 128 bytes at a time. The FIFO is accessed via the Receive Buffer Register.

Issue:

The FIFO pointer needs 4 CPU clock cycles to be properly updated when the first byte in the FIFO is accessed. As a result, reading the FIFO with the "streamout.x" instruction (an FT90x assembly instruction) will result in the first byte in the FIFO being returned three times for the first three reads.

Workaround:

The software can choose to add some delay after reading the first byte to make sure the FIFO pointer is properly updated.

4.11 I²S

4.11.1 TX FIFO Counter may be corrupted when the FIFO is underrun in 20/24/32-bit mode

Introduction:

The FT90x I2S provides a counter for the TX FIFO in the form of register I2STXCOUNT. The register value is the number of 16-bit (data) words in the FIFO.

Issue:

The I2S FIFO only works with 16-bit data is 16-bits wide. In 20/24/32-bit sample mode, each sample takes two 16-bit writes to be completely written to the FIFO.

During normal operation, when the FIFO reaches empty, the I2S engine repeatedly outputs the last sample until new samples are loaded into the FIFO. In case of, 20/24/32-bit sample mode, 2 reads are required to load the last sample for transmission.

When the second read access coincides with the first 16-bit write of a new sample, the TX FIFO count does not change. The second 16-bit write of the new sample therefore causes the counter to increase by 1 and becomes an odd number. The next time the FIFO goes empty again, the TX FIFO counter will flip from 1 to 0xFFFF.

Workaround:

The software should try to prevent FIFO underrun by maintaining a half-empty threshold point. I.e. whenever the FIFO is half-empty, software shall fill the FIFO with fresh samples.

4.11.2 Left and right samples may be swapped when the TX FIFO is underrun

Introduction:

The FT90x I2S engine sends out data according to the left/right clock signal (LRCLK) according to the I2S specification.

Issue:

The FT90x I2S engine sends out data beginning with the left sample when it starts. When the TX FIFO is underrun, the I2S engine repeatedly outputs the last sample on both left and right channels. When new sample data is available, the I2S engine loads the fresh sample and transmits on the next channel without regard to left or right, instead of waiting for the left channel to start.

Workaround:

The software should try to prevent FIFO underrun by maintaining a half-empty threshold point. I.e. whenever the FIFO is half-empty, software shall fill the FIFO with fresh samples.

4.12 SPI Slave

4.12.1 Missing data in FIFO mode during transmission

Introduction:

The FT90x SPI Slave contains a TX FIFO so that the CPU can fill the data once and perform other tasks while the SPI engine handles the data transfer.

Issue:

If an external SPI Master reads the data from the FT90x SPI Slave FIFO multiple times and enables/disables the CS pin each time, the SPI Slave FIFO will start sending out data from the second available byte for each read, instead of the first byte.

Workaround:

The software should make sure to enable CS until all available data in the FIFO have been read. A protocol may also be agreed upon to make sure the SPI Slave fills in just enough data for the SPI Master to read in one attempt.

4.12.2 Data has to align to the RX FIFO size during reception

Introduction:

The FT90x SPI Slave contains a RX FIFO so that the CPU perform other tasks while the SPI engine handles the data reception.

Issue:

The SPI Slave RX FIFO cannot receive data that is not aligned to the FIFO size. For example:

- If the RX FIFO size is 16 bytes and the SPI Master sends less than 16 bytes, the data will be lost.
- If the RX FIFO size is 16 bytes and the SPI Master sends more than 16 bytes, only 16 bytes will be received.
- The same behaviour applies when the RX FIFO size is 64 bytes of 256 bytes.

Workaround:

The SPI Master should take care to send data in blocks whose size is aligned to the SPI Slave RX FIFO size. If the amount of data is not enough, some data padding needs to be done.

5 Contact Information

Head Quarters – Singapore

Bridgetek Pte Ltd
178 Paya Lebar Road, #07-03
Singapore 409030
Tel: +65 6547 4827
Fax: +65 6841 6071

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Branch Office – Taipei, Taiwan

Bridgetek Pte Ltd, Taiwan Branch
2 Floor, No. 516, Sec. 1, Nei Hu Road, Nei Hu District
Taipei 114
Taiwan, R.O.C.
Tel: +886 (2) 8797 5691
Fax: +886 (2) 8751 9737

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Branch Office - Glasgow, United Kingdom

Bridgetek Pte. Ltd.
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales.emea@brtichip.com
E-mail (Support) support.emea@brtchip.com

Branch Office – Vietnam

Bridgetek VietNam Company Limited
Lutaco Tower Building, 5th Floor, 173A Nguyen Van Troi,
Ward 11, Phu Nhuan District,
Ho Chi Minh City, Vietnam
Tel : 08 38453222
Fax : 08 38455222

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Web Site

http://brtchip.com/

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

http://brtchip.com/ft900/

Acronyms and Abbreviations

Terms	Description
ADC	Analog to Digital Converter
CAN	Controller Area Network
DAC	Digital to Analog Converter
FIFO	First In First Out
IRQ	Interrupt Request
PWM	Pulse Width Modulation
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver/Transmitter

Appendix B – List of Tables & Figures

List of Figures

List of Tables

Appendix C – Revision History

Document Title: TN_159 FT90x Errata

Document Reference No.: BRT_000165

Clearance No.: BRT#101

Product Page: http://brtchip.com/product

Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	13-10-2015
1.1	Document migrated from FTDI to BRT (Updated company logo; copyright info; contact information; hyperlinks)	01-11-2017
1.2	Updated Rev C related changes	15-12-2020