RENESAS TECHNICAL UPDATE

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan Renesas Electronics Corporation

Product Category	MPU & MCU	Document No.	TN-RX*-A115A/E Rev. 1.0			
Title	Corrections to Manual regarding the USB 2.0 F Module (USBa) in the RX630 Group	Information Category	Technical Notification			
		Lot No.				
Applicable Product	RX630 Group	Reference Document	RX630 Group User' Hardware Rev.1.60 (R01UH0040EJ016		ual:	

This document describes corrections to section 31, USB 2.0 Function Module (USBa) in RX630 Group User's Manual: Hardware.

Overall

The following function name is corrected:

Before correction SOF interpolation function

<u>After correction</u> SOF recovery function

•Page 963 of 1681

Note 1 is added to Table 31.1 as follows:

ltem	Specifications					
Features	USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated.					
	One port is provided.					
	 Self-power mode or bus-power mode can be selected. 					
	 Programmable intervals for isochronous and interrupt transfers 					
	 Full-speed transfer (12 Mbps) is supported. *1 					
	Control transfer stage control function					
	Device state control function					
	 Auto response function for SET_ADDRESS request 					
	SOF recovery function					



0 1 0: USB bus reset in progress or full-speed connection

•Page 967 of 1681

The description column for DVSTCTR0.RHST[2:0] bits in 31.2.3 is corrected as follows:

Before correction

Bit	Symbol	Bit Name	Description					
b2 to b0	RHST[2:0]	USB Bus Reset Status	b2 b1 b0					
			0 0 0: Communication speed not determined 1 0 0: USB bus reset in progress					
			0 1 0: Full-speed connection					
After corre	ection							
		Dit Nome	·					
Bit	Symbol	Bit Name	Description					
After corre Bit b2 to b0		Bit Name USB Bus Reset Status	·					

•Page 967 of 1681

The following note for the DVSTCTR0.WKUP bit in 31.2.3 is deleted: Note 1. Only 1 can be written.

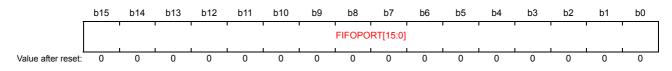


• Pages 968, 969 of 1681

The descriptions for registers CFIFO, D0FIFO, D1FIFO in 31.2.4 are corrected as follows:

Before correction

Address(es): USB0.CFIFO 000A 0014h, USB0.D0FIFO 000A 0018h, USB0.D1FIFO 000A 001Ch



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	FIFOPORT[15:0]	FIFO Port	The valid bits in a FIFO port register depend on the settings of the corresponding MBW and BIGEND bits as listed in Table 31.4 and Table 31.5	R/W

FIFOPORT[15:0] Bits (FIFO Port)

Accessing the FIFOPORT[15:0] bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY bit in each port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW and BIGEND bits of port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL), as listed in Table 31.4 and Table 31.5.

Table 31.4 Endian Operation in 16-Bit Access

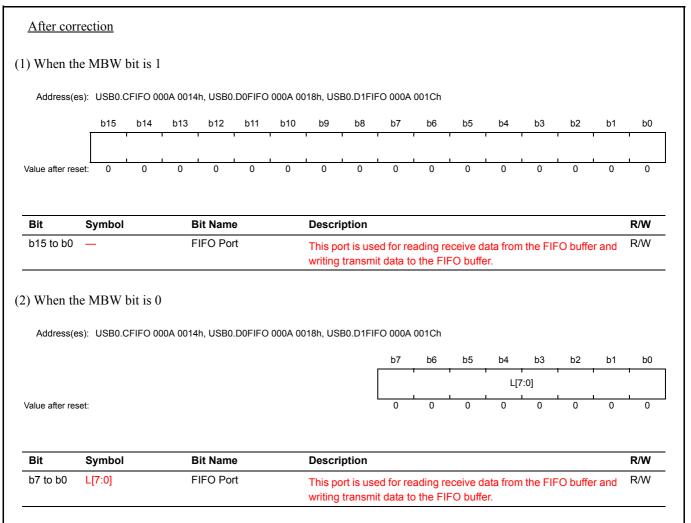
CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit			
D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0	
0	N + 1 data	N + 0 data	
1	N + 0 data	N + 1 data	

Table 31.5 Endian Operation in 8-Bit Access

CFIFOSEL.BIGEND Bit			
D0FIFOSEL.BIGEND Bit			
D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0	
0	Access prohibited *1	N + 0 data	
1	Access prohibited *1	N + 0 data	

Note 1. Accessing an access-prohibited area is not allowed.





FIFO Port Bits

Accessing the FIFO port bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY bit in each FIFO port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW bit of the FIFO port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL).

When the MBW bit is 1 (16-bit width), the data arrangement may differ from the data arrangement on the RAM depending on the value of the MDEB.MDE[2:0] bits or the MDES.MDE[2:0] bits and the setting of the BIGEND bit (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, or D1FIFOSEL.BIGEND). Table 31.4 lists the endian operation in 16-bit access. Note that if the total number of transmit data bytes is odd, access the L[7:0] bits in bytes when writing the last data.

When the MBW bit is 0 (8-bit width), access the L[7:0] bits in bytes.

Table 31.4 Endian Operation in 16-Bit Access

MDEB.MDE[2:0] bits MDES.MDE[2:0] bits	CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0	Remarks
000h (big opdian)	0 (little endian)	Data in address N + 1	Data in address N	Bytes reversed
000b (big endian)	1 (big endian)	Data in address N	Data in address N + 1	
111b (little endian)	0 (little endian)	Data in address N + 1	Data in address N	
	1 (big endian)	Data in address N	Data in address N + 1	Bytes reversed



•Page 974 of 1681

The note for the BCLR bit in registers CFIFOCTR, D0FIFOCTR, and D1FIFOCTR in 31.2.6 is corrected as follows:

Before correction

Note 1. Only 0 can be read and 1 can be written.

After correction

Note 1. This bit is read as 0.

•Page 974 of 1681

The following note for the BVAL bit in registers CFIFOCTR, D0FIFOCTR, and D1FIFOCTR in 31.2.6 is deleted: Note 2. Only 1 can be written.

•Page 981 of 1681

The notes for the value after reset of the INTSTS0 register in 31.2.12 are corrected as follows:

Before correction

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	[DVSQ[2:0]	VALID	(CTSQ[2:0]
Value after reset:	0	0	0	0/1 *1	0	0	0	0	0 *3	0	0	0/1 *2	0	0	0	0

Note 1. This bit is initialized to 0b by a power-on reset and 1b by a USB bus reset. Note 2. These bits are initialized to 000b by a power-on reset and 001b by a USB bus reset. Note 3. This bit is 1 when the USB0 VBUS pin input is high and 0 when the input is low.

After correction

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	[DVSQ[2:0)]	VALID	(CTSQ[2:0]	1
Value after reset:	0	0	0	0/1 *1	0	0	0	0	0 *2	0 *3	0 *3	0/1 *3	0	0	0	0

Note 1. The value is 0 after the MCU is reset, and the value is 1 after a USB bus reset.

Note 2. The value is 1 when the USB0_VBUS pin is high, and the value is 0 when the USB0_VBUS pin is low. Note 3. The value is 000b after the MCU is reset, and the value is 001b after a USB bus reset.

•Page 986 of 1681

The note for bits CRCE and OVRN of the FRMNUM register in 31.2.16 is corrected as follows:

Before correction

Note 1. Only 0 can be written.

After correction

Note 1: When setting each status to 0, write 0 to the bit that is cleared, and write 1 to the other bit.



•Page 986 of 1681

The descriptions for the CRCE bit in 31.2.16 is corrected as follows:

Before correction

On detecting a CRC error, the USB module does not generate the internal NRDY interrupt request.

After correction

The USB module generates an internal NRDY interrupt request when a CRC error is detected.

•Page 991 of 1681

The note for bits SQSET and SQCLR of the DCPCTR register in 31.2.24 is corrected as follows:

Before correction

Note 1. This bit is read as 0. Only 1 can be written.

After correction

Note 1. This bit is read as 0.

•Page 996 of 1681

Descriptions for the PIPEMAXP.MXPS[8:0] bits in 31.2.27 are corrected as follows:

Before correction

Specifies the maximum data payload (maximum packet size) for the selected pipe. These bits should be set to the appropriate value for each transfer type based on USB Specifications. While MXPS[8:0] = 0, do not write to the FIFO buffer or set PID to BUF.

After correction

Specifies the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on USB Specifications 2.0. Note that the maximum value for PIPE1 and PIPE2 is 256. While the MXPS[8:0] bits are 000h, do not write to the FIFO buffer or do not set the PID[1:0] bits to 01b (BUF).

•Pages 998, 1002 of 1681

The note for bits SQSET and SQCLR of the PIPEnCTR register in 31.2.29 is corrected as follows:

Before correction

Note 1. Only 0 can be read and 1 can be written.

<u>After correction</u> Note 1. This bit is read as 0.



fore correction				
(1) Example of zero-	length packet reception or	data packet reception whe		node)
JSB bus —	Token Packet	Data Packet	ACK Handshake	
- IFO buffer status ——				
	Ready for reception			Ready for read access
BRDY interrupt change in correspondin bit in PIPEBRDY)	ng			
			A BRDY buffer be	「 interrupt is generated because the comes ready for read access. ²
(2) Example of data	packet reception when BF	RE = 1 (single-buffer mode)		1
JSB bus —	Token Packet	<last> Data Packet</last>	ACK Handshake	
FIFO buffer status	Ready for reception	I		Ready for read access
BRDY interrupt change in correspondin it in PIPEBRDY)	ng			
			The buffer becomes ready for read acces	A BRDY interrupt is generated s. ² because the transfer has ende
(3) Example of pack	et transmission (single-bu	ffer mode)		1
JSB bus —	Token Packet	Data Packet	ACK Handshake	
FIFO buffer status	Ready for transmi	ssion		Ready for write access
BRDY interrupt change in correspondin bit in PIPEBRDY)	ng			
			A BRDY interrup because the buff ready for write a	fer becomes
Packet transmitted	by host device Pac	ket transmitted by peripheral de	,	
Note 2. The FIFO bu When a pack Note 3. A transfer en (1) When a s	ket is received while no data re ids under either of the following hort packet including a zero-le	ccess under the following condi mains unread in the buffer in th g conditions:	e CPU.	



4

USB bus -	Token Packet	Data Packet	ACK Handshake	_*'
FIFO buffer status	Ready for receptior			Ready for read access
-				
BRDY interrupt (BRDYSTS.PIPEnBRDY - bit)				4
(2) Example of data pack	et reception when BFRE = 1	(single buffer mode)	A BRDY interrupt of buffer becomes read	Accurs because the FIFO ady for read access.*2
USB bus -	Token Packet	- <last> Data Packet</last>	ACK Handshake	* ¹
FIFO buffer status	Boody for recention			Ready for read access
-	Ready for receptior	1		Ready for read access
BRDY interrupt (BRDYSTS.PIPEnBRDY - bit)				
				Å Å
			The FIFO buff ready for read	access.*2
(3) Example of packet tra	nsmission (single-buffer m	ode)		A BRDY interrupt occurs becaute the transfer has ended.*3
USB bus -	Token Packet	Data Packet	ACK Handshake	<u>}</u>
FIFO buffer status	Ready for transmis	sion		Ready for write access
BRDY interrupt				
(BRDYSTS.PIPEnBRDY - bit)				
			A BRDY interrupt occ	T curs because the FIFO
Packet transmitted by he		nsmitted by function device	buffer becomes ready	for write access.
ote 2. The FIFO buffer becor When a packet is rece	s not used in isochronous trans nes ready for read access unde ived while no data remains unre either of the following condition:	er the following condition: ead in the FIFO buffer on th	ne CPU <mark>side</mark> .	
(1) When a short pack	et including a zero-length packe of packets specified in the trans	et is received	1	
(1) When a short pack (2) When the number of	et including a zero-length packe	et is received action counter are received	1	
(1) When a short pack (2) When the number of	et including a zero-length packe of packets specified in the trans	et is received action counter are received	1	
(1) When a short pack (2) When the number of	et including a zero-length packe of packets specified in the trans	et is received action counter are received	1	
(1) When a short pack (2) When the number of	et including a zero-length packe of packets specified in the trans	et is received action counter are received	1	

fore correction			
1) Example of data transm	nission (single-buffer mode)		
USB bus	IN Token Packet	NAK Handshake	*1
Buffer status	Ready for write access (th	nere is no data to be transmitte	ed)
NRDY interrupt (change in corresponding bit in PIPENRDY)* ²			
2) Example of data recepti	ion: OUT token reception (singl	e-buffer mode)	*1
USB bus	OUT Token Packet	Data Packet	NAK Handshake
Buffer status	Ready for read access (th	ere is no space to receive dat	a)
NRDY interrupt (change in corresponding bit in PIPENRDY)* ²	9		1
(CRC bit, etc.)* ³			1
3) Example of data recepti	ion: PING token reception (sing	le-buffer mode)	_
USB bus	PING Packet	NAK Handshake	
Buffer status	Ready for read access (th	ere is no space to receive dat	a)
NRDY interrupt (change in corresponding bit in PIPENRDY)* ²	9	1	
		lat transmitted by parishard	device
	ted by host device Pac	ket transmitted by peripheral	device

Figure 31.9 Timing of NRDY Interrupt Generation



USB bus —	IN Token Packet	NAK Handshake		
FIFO buffer status —			200 N	
NRDY interrupt (NRDYSTS.PIPEnNRDY	Ready for write access (tr	nere is no data to be transm	litted)	
bit) *2				
	An NRDY ii	nterrupt occurs		
) Example of data reception	: OUT token reception (single	buffer mode)		
USB bus —	OUT Token Packet	Data Packet	– NAK Handshake	* ¹
FIFO buffer status —		· · · · · · ·	<u> </u>	
NRDY interrupt (NRDYSTS.PIPEnNRDY bit) * ²	Ready for read access (there	e is no space to receive data	a)	
(CRCE bit, etc.) * ³				
) Example of data reception	: PING token reception (single		↑ DY interrupt occurs	
USB bus —	PING Packet	NAK Handshake]	
FIFO buffer status —	Ready for read access	(there is no space to receive	e data)	
NRDY interrupt (NRDYSTS.PIPEnNRDY bit) * ²				
,	T An NRDY	interrupt occurs		
	y the host device Pack	et transmitted by the function	n device	

Figure 31.9 Timing of NRDY Interrupt Generation



(1) Example of data to	transmissi	on					
USB bus	—	IN Token Packet	-	Data Packet	Н	ACK Handshake	*1
Buffer status		Ready for transr	mission				Ready for write access (there is no data to be
BEMP interrupt							transmitted)
(change in correspond bit in PIPEBEMP)	nding						
(2) Example of data r	reception						
USB bus	-	OUT Token Packet	t –	Data Packet (Maximun Packet size over)		STALL Handshake]
BEMP interrupt (change in correspond bit in PIPEBEMP)	nding						
		by host device		t transmitted by peripher	<mark>al</mark> devic	e	
Note 1. The har	indshake is	not used in isochror	nous trans	sters.			
			-pr	neration			
(1) Example of data t	transmissi	on					*1
(1) Example of data to	-	on IN Token Packet		Data Packet		ACK Handshake	*1
(1) Example of data to USB bus FIFO buffer status	-	on				ACK Handshake	*1 Ready for write access (there is no data to be transmitted)
1) Example of data to		on IN Token Packet]-[ACK Handshake	(there is no data to be
(1) Example of data to USB bus FIFO buffer status BEMP interrupt (BEMPSTS.PIPEnB bit)	Re BEMP	on IN Token Packet					(there is no data to be
(1) Example of data to USB bus FIFO buffer status BEMP interrupt (BEMPSTS.PIPEnB bit)	Re BEMP	on IN Token Packet					(there is no data to be transmitted)
 (1) Example of data to USB bus FIFO buffer status BEMP interrupt (BEMPSTS.PIPEnB bit) (2) Example of data ro USB bus BEMP interrupt (BEMPSTS.PIPEnB 	Re BEMP	on IN Token Packet ady for transmission		Data Packet		A BE	(there is no data to be transmitted)
 (1) Example of data to USB bus FIFO buffer status BEMP interrupt (BEMPSTS.PIPEnB bit) (2) Example of data ro USB bus BEMP interrupt 	Re BEMP	on IN Token Packet ady for transmission		Data Packet Data Packet (Maximun packet size over)		A BE STALL Handshake	(there is no data to be transmitted)
 (1) Example of data to USB bus FIFO buffer status BEMP interrupt (BEMPSTS.PIPEnB bit) (2) Example of data ro USB bus BEMP interrupt (BEMPSTS.PIPEnB 	Re BEMP reception BEMP	on IN Token Packet ady for transmission OUT Token Packet		Data Packet Data Packet (Maximun packet size over)	MP inte	A BE STALL Handshake	(there is no data to be transmitted)
FIFO buffer status BEMP interrupt (BEMPSTS.PIPEnB bit) (2) Example of data re USB bus BEMP interrupt (BEMPSTS.PIPEnB bit)	Re BEMP reception BEMP	on IN Token Packet ady for transmission OUT Token Packet		Data Packet Data Packet (Maximum packet size over)	MP inte	A BE STALL Handshake	
 (1) Example of data to USB bus FIFO buffer status BEMP interrupt (BEMPSTS.PIPEnB bit) (2) Example of data rouse bus BEMP interrupt (BEMPSTS.PIPEnB bit) Packet transmit 	Re BEMP reception BEMP ditted by the ake is not u	on IN Token Packet ady for transmission OUT Token Packet host device	t - (Data Packet Data Packet (Maximun packet size over) A BE et transmitted by the fun	MP inte	A BE STALL Handshake	(there is no data to be transmitted)
 (1) Example of data to USB bus FIFO buffer status BEMP interrupt (BEMPSTS.PIPEnB bit) (2) Example of data rous BEMP interrupt (BEMPSTS.PIPEnB bit) Packet transmit Note 1. The handshall 	Re BEMP reception BEMP ditted by the ake is not u	on IN Token Packet ady for transmission OUT Token Packet host device	t - (Data Packet Data Packet (Maximun packet size over) A BE et transmitted by the fun	MP inte	A BE STALL Handshake	(there is no data to be transmitted)



•Page 1032 of 1681

The descriptions for (4) Control Transfer Auto Response Function in 31.3.6.1 are corrected as follows:

Before correction

(4) Control Transfer Auto Response Function

The USB module automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- Any transfer other than a control read transfer: $bmRequestType \neq 00h$
- Request error : wIndex \neq 00h
- Any transfer other than a no-data control transfer: wLength $\neq 00h$
- Request error: wValue > 7Fh
- Control transfer of a device state error: INTSTS0.DVSQ[2:0] = 011b (Configured)

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

After correction

(4) Control Transfer Auto Response Function

The USB module automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wLength is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- The INTSTS0.DVSQ[2:0] bits are 011b (configured state): Control transfer of a device state error

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

•Page 1035 of 1681

The descriptions for (1) Counter Initialization in 31.3.9.3 are corrected as follows:

Before correction

(1) Counter Initialization

The USB module initializes the interval counter under the following conditions.

- Power-on Reset The PIPEPERI.IITV[2:0] bits are initialized.
- Buffer memory initialization using the ACLRM bit The PIPEPERI.IITV[2:0] bits are not initialized but the count value is initialized. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in the PIPEPERI.IITV[2:0] bits.

After correction

(1) Counter Initialization

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. The PIPEPERI.IITV[2:0] bits are not initialized when the interval counter is initialized by using the ACLRM bit.



•Page 1036 of 1681

The descriptions for "When IITV = 0" in 31.3.9.3 (2) and Figure 31.13 are corrected as follows:

Before correction

• When IITV = 0: The interval counting starts at the frame following the frame in which software has set the PID[1:0] bits for the selected pipe to BUF.

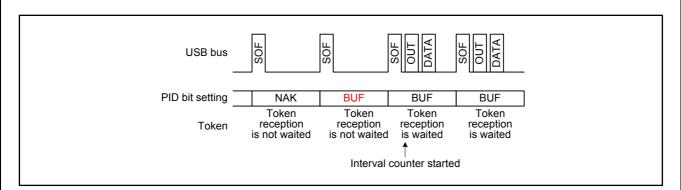


Figure 31.13 Relationship between Frames and Expected Token Reception when IITV = 0

After correction

• When IITV = 0 The interval counter starts when software has set the PID[1:0] bits for the selected pipe to BUF.

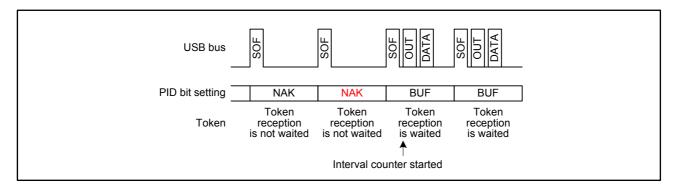


Figure 31.13 Relationship between Frames and Expected Token Reception when IITV = 0



•Page 1040 of 1681

The descriptions for SOF Interpolation Function in 31.3.10 are corrected as follows:

Before correction

31.3.10 SOF Interpolation Function

If data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB module interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in SYSCFG have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB module supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing when full-speed operation is being used, the FRMNUM.FRNM[10:0] bits are not updated.

After correction

31.3.10 SOF Recovery Function

If data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB module recovers the SOF. The SOF recovery operation begins when the USBE and SCKE bits in SYSCFG have been set to 1 and an SOF packet is received. The recovery function is initialized under the following conditions.

- MCU reset
- USB bus reset
- Suspended state detected

The SOF recovery operates as follows.

- The recovery function is not activated until an SOF packet is received.
- After the first SOF packet is received, recovery is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, recovery is carried out at the previous reception interval.
- Recovery is not carried out in the suspended state or while a USB bus reset is being received.

The USB module supports the following functions based on the SOF packet reception. These functions also operate normally with SOF recovery, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing when full-speed operation is being used, the FRMNUM.FRNM[10:0] bits are not updated.

