Toshiba Semiconductor and Storage - TMPM330FYFG(B) Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	40MHz
Connectivity	I²C, SIO, UART/USART
Peripherals	POR, WDT
Number of I/O	78
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/toshiba-semiconductor-and-storage/tmpm330fyfg-b

10.11.3 Receive Operation....244
10.11.3.1 Receive Buffer
10.11.3.2 Receive FIFO Operation
10.11.3.3 I/O interface mode with SCLK output
10.11.3.4 Read Received Data
10.11.3.5 Wake-up Function
10.11.3.6 Overrun Error
10.12 Transmission....248
10.12.1 Transmission Counter....248
10.12.2 Transmission Control....248
10.12.2.1 I/O Interface Mode
10.12.2.2 UART Mode
10.12.3 Transmit Operation....248
10.12.3.1 Operation of Transmission Buffer
10.12.3.2 Transmit FIFO Operation
10.12.3.3 I/O interface Mode/Transmission by SCLK Output
10.12.3.4 Under-run error
10.13 Handshake function....252
10.14 Interrupt/Error Generation Timing....253
10.14.1 RX Interrupts....253
10.14.1.1 Single Buffer / Double Buffer
10.14.1.2 FIFO
10.14.2 TX interrupts....254
10.14.2.1 Single Buffer / Double Buffer
10.14.2.2 FIFO
10.14.3 Error Generation....255
10.14.3.1 UART Mode
10.14.3.2 IO Interface Mode
10.15 Software Reset....255
10.16 Operation in Each Mode....256
10.16.1 Mode 0 (I/O interface mode)....256
10.16.1.1 Transmitting Data
10.16.1.2 Receive
10.16.1.3 Transmit and Receive (Full-duplex)
10.16.2 Mode 1 (7-bit UART mode)....267
10.16.3 Mode 2 (8-bit UART mode)....267
10.16.4 Mode 3 (9-bit UART mode)....268
10.16.4.1 Wakeup function
10.16.4.2 Protocol

11. Serial Bus Interface (I2C/SIO)

11.1 Configuration....272
11.2 Register....273
11.2.1 Registers for each channel....273
11.3 I2C Bus Mode Data Format....274
11.4 Control Registers in the I2C Bus Mode....275
11.4.1 SBIxCR0(Control register 0)....275
11.4.2 SBIxCR1(Control register 1)....276
11.4.3 SBIxCR2(Control register 2)....278
11.4.4 SBIxSR (Status Register)....279
11.4.5 SBIxBR0(Serial bus interface baud rate register 0)....280
11.4.6 SBIxDBR (Serial bus interface data buffer register)....280
11.4.7 SBIxI2CAR (I2Cbus address register)....281
11.5 Control in the I2C Bus Mode....282
11.5.1 Serial Clock....282
11.5.1.1 Clock source
11.5.1.2 Clock Synchronization
11.5.2 Setting the Acknowledgement Mode....283
11.5.3 Setting the Number of Bits per Transfer....283
11.5.4 Slave Addressing and Address Recognition Mode....283
11.5.5 Operating mode....283
11.5.6 Configuring the SBI as a Transmitter or a Receiver....284
11.5.7 Configuring the SBI as a Master or a Slave....284
11.5.8 Generating Start and Stop Conditions....284

6.3.3 Clock system Diagram

Figure 6-1 shows the clock system diagram.

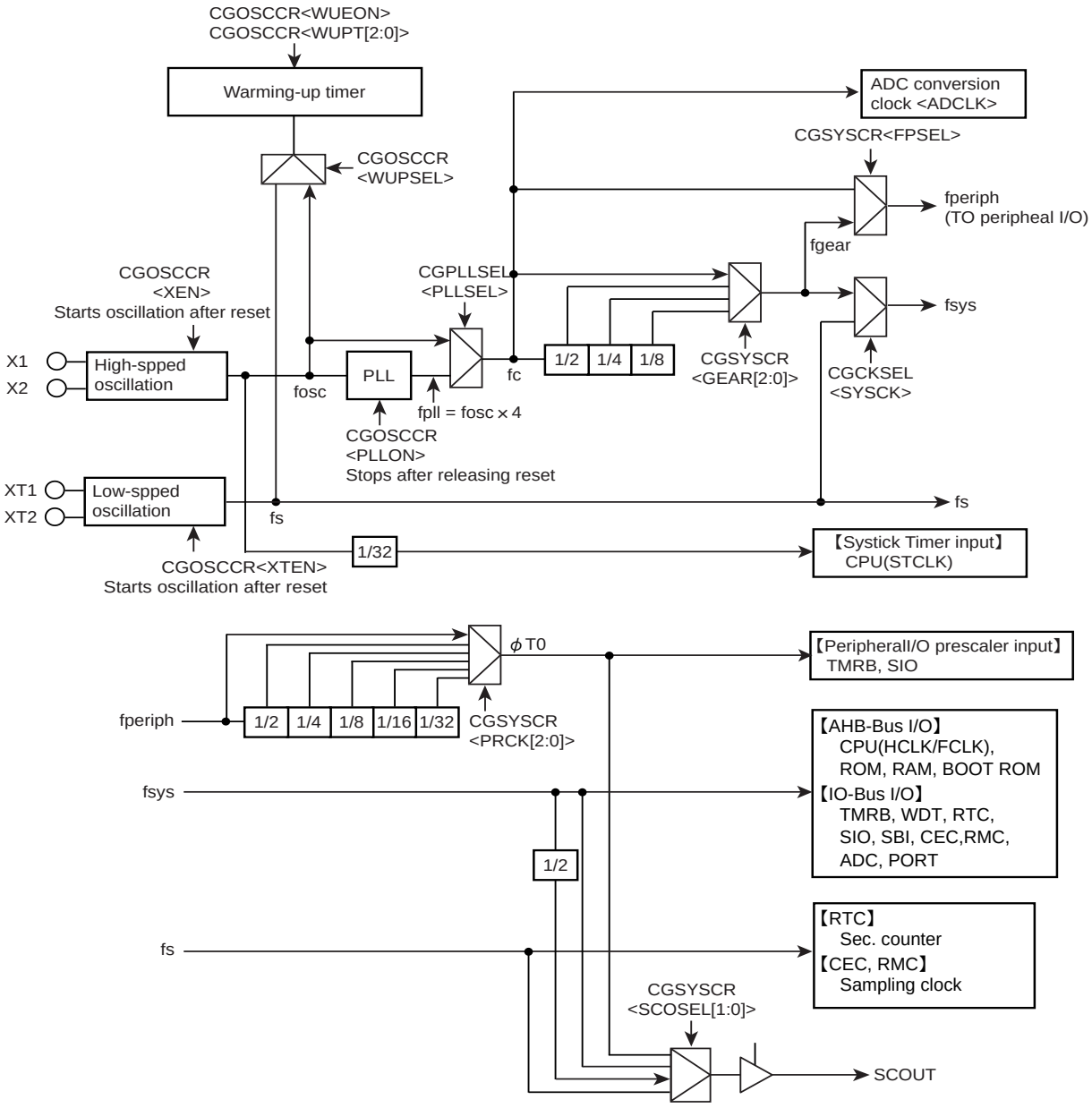


Figure 6-1 Clock Block Diagram

The input clocks to selector shown with an arrow are set as default after reset.

(2) Exception detection

If multiple exceptions occur simultaneously, the CPU takes the exception with the highest priority.

Table 7-1 shows the priority of exceptions. "Configurable" means that you can assign a priority level to that exception. Memory Management, Bus Fault and Usage Fault exceptions can be enabled or disabled. If a disabled exception occurs, it is handled as Hard Fault.

Table 7-1 Exception Types and Priority

No.	Exception type	Priority	Description
1	Reset	−3 (highest)	Reset pin, WDT or SYSRETREQ
2	Non-Maskable Interrupt	−2	$\overline{\text{NMI}}$ pin or WDT
3	Hard Fault	−1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled
4	Memory Management	Configurable	Exception from the Memory Protection Unit (MPU) (Note 1) Instruction fetch from the Execute Never (XN) region
5	Bus Fault	Configurable	Access violation to the Hard Fault region of the memory map
6	Usage Fault	Configurable	Undefined instruction execution or other faults related to instruction execution
7~10	Reserved	−	
11	SVCall	Configurable	System service call with SVC instruction
12	Debug Monitor	Configurable	Debug monitor when the CPU is not faulting
13	Reserved	−	
14	PendSV	Configurable	Pendable system service request
15	SysTick	Configurable	Notification from system timer
16~	External Interrupt	Configurable	External interrupt pin or peripheral function (Note 2)

Note 1: **This product does not contain the MPU.**

Note 2: **External interrupts have different sources and numbers in each product. For details, see "7.5.1.5 List of Interrupt Sources".**

(3) Priority setting

- Priority levels

  The external interrupt priority is set to the interrupt priority register and other exceptions are set to <PRI_n> bit in the system handler priority register.

  The configuration <PRI_n> can be changed, and the number of bits required for setting the priority varies from 3 bits to 8 bits depending on products. Thus, the range of priority values you can specify is different depending on products.

  In the case of 8-bit configuration, the priority can be configured in the range from 0 to 255. The highest priority is "0". If multiple elements with the same priority exist, the smaller the number, the higher the priority becomes.

Note: **<PRI_n> bit is defined as a 3-bit configuration with this product.**

- Priority grouping

  The priority group can be split into groups. By setting the <PRIGROUP> of the application interrupt and reset control register, <PRI_n> can be divided into the pre-emption priority and the sub priority.

  A priority is compared with the pre-emption priority. If the priority is the same as the pre-emption priority, then it is compared with the sub priority. If the sub priority is the same as the priority, the smaller the exception number, the higher the priority.

Bit	Bit Symbol	Type	Function
11-10	EMST9[1:0]	R	active level of INT6 standby clear request. 00: − 01: Rising edge 10: Falling edge 11: Both edges
9	−	R	Reads as undefined.
8	INT9EN	R/W	INT6 clear input 0:Disable 1: Enable
7	−	R	Read as 0.
6-4	EMCG8[2:0]	R/W	active level setting of INTRTC standby clear request. **Set it as shown below.** 010: Falling edge
3-2	EMST8[1:0]	R	active level of INTRTC standby clear request. 00: − 01: Rising edge 10: Falling edge 11: Both edges
1	−	R	Reads as undefined.
0	INT8EN	R/W	INTRTC clear input 0:Disable 1: Enable

Note 1: <EMSTx> is effective only when <EMCGx[2:0]> is set to "100" for both rising and falling edge. The active level used for the reset of standby can be checked by referring <EMSTx>. If interrupts are cleared with the CGICRCG register, <EMSTx> is also cleared.

Note 2: Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

8.2.6.6 PFFR2(Port F function register 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	PF2F2	-	-
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-3	−	R	Read as 0.
2	PF2F2	R/W	0: PORT 1: $\overline{CTS2}$
1-0	−	R	Read as 0.

8.2.6.7 PFOD (Port F open drain control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PF7OD	PF6OD	PF5OD	PF4OD	PF3OD	PF2OD	PF1OD	PF0OD
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PF7OD-PF0OD	R/W	0: CMOS 1: Open-drain

8.2.10.6 PJPUP (Port J pull-up control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7UP	PJ6UP	PJ5UP	PJ4UP	PJ3UP	PJ2UP	PJ1UP	PJ0UP
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7UP-PJ0UP	R/W	Pull-up 0: Disable 1: Enable

8.2.10.7 PJIE (Port J input control register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PJ7IE	PJ6IE	PJ5IE	PJ4IE	PJ3IE	PJ2IE	PJ1IE	PJ0IE
After reset	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	−	R	Read as 0.
7-0	PJ7IE-PJ0IE	R/W	Input 0: Disable 1: Enable

8.4.6 Port F Setting

Table 8-11 Port Setting List (Port F)

Pin	Port Type	Function	After re-set	PFCR	PFFR1	PFFR2	PFOD	PFPUP	PFIE
PF0	T10	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		TXD2(Output)		1	1	0	x	x	0
PF1	T4	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		RXD2(Input)		0	1	0	x	x	1
PF2	T16	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SCLK2(Input)		0	1	1	x	x	1
		SCLK2(Output)		1	1	0	x	x	0
		$\overline{\text{CTS2}}$(Input)		0	0	0	x	x	1
PF3	T4	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		RXIN1(Input)		0	1	0	x	x	1
PF4	T13	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SO1(Output)		1	1	0	x	x	0
		SDA1(Input/Output)		1	1	0	1	x	1
PF5	T13	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SI1(Input)		0	1	0	x	x	1
		SCL1(Input/Output)		1	1	0	1	x	1
PF6	T13	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		SCK1(Input)		0	1	0	x	x	1
		SCK1(Output)		1	1	0	x	x	0
PF7	T8	Input Port		0	0	0	x	x	1
		Output Port		1	0	0	x	x	0
		INT5(Input)		0	1	0	x	x	1

9.4.5 TBxMOD(Mode register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	TBCP	TBCPM		TBCLE	TBCLK	
After reset	0	0	1	0	0	0	0	0

Bit	Bit Symbol	Type	Function
31-7	−	R	Read as 0.
6	−	R/W	Write 0.
5	TBCP	W	Capture control by software 0: Capture by software 1: Don’t care When "0" is written, the capture register 0 (TBxCP0) takes count value. Read as 1.
4-3	TBCPM[1:0]	R/W	Capture timing 00: Disable Capture timing 01: TBxIN0↑ TBxIN1↑ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN0 pin input. Takes count values into capture register 1 (TBxCP1) upon rising of TBxIN1 pin input. 10: TBxIN0↑ TBxIN0↓ Takes count values into capture register 0 (TBxCP0) upon rising of TBxIN0 pin input. Takes count values into capture register 1 (TBxCP1) upon falling of TBxIN0 pin input. 11: TBxOUT↑ TBxOUT↓ Takes count values into capture register 0 (TBxCP0) upon rising of 16-bit timer match output (TBxOUT) and into capture register 1 (TBxCP1) upon falling of TBxOUT. (TMRB0 and TMRB1:TB7OUT, TMRB2 through TMRB4:TB8OUT, TMRB5 and TMRB6:TB9OUT).
2	TBCLE	R/W	Up-counter control 0: Disables clearing of the up-counter. 1: Enables clearing of the up-counter. Clears and controls the up-counter. When "0" is written, it disables clearing of the up-counter. When "1" is written, it clears up counter when there is a match with Timer Regsiter1 (TBxRG1).
1-0	TBCLK[1:0]	R/W	Selects the TMRBx source clock. 00: TBxIN0 pin input 01: φT1 10: φT4 11: φT16

9.7 Applications using the Capture Function

The capture function can be used to develop many applications, including those described below:

1. One-shot pulse output triggered by an external pulse
2. Frequency measurement
3. Pulse width measurement
4. Time difference measurement

9.7.1 One-shot pulse output triggered by an external pulse

One-shot pulse output triggered by an external pulse is carried out as follows:

The 16-bit up-counter is made to count up by putting it in a free-running state using the prescaler output clock. An external pulse is input through the TBxIN0 pin. A trigger is generated at the rising of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TBxCP0).

The CPU must be programmed so that an interrupt INTCAPx0 is generated at the rising of an external trigger pulse. This interrupt is used to set the timer registers (TBxRG0) to the sum of the TBxCP0 value (c) and the delay time (d), (c + d), and set the timer registers (TBxRG1) to the sum of the TBxRG0 values and the pulse width (p) of one-shot pulse, (c + d + p).[TBxRG1 change must be completed before the next match.]

In addition, the timer flip-flop control registers(TBxFFCR<TBE1T1, TBE0T1>) must be set to "11". This enables triggering the timer flip-flop (TBxFF0) to reverse when TBxUC matches TBxRG0 and TBxRG1. This trigger is disabled by the INTTBx interrupt after a one-shot pulse is output.

Symbols (c), (d) and (p) used in the text correspond to symbols c, d and p in "Figure 9-5 One-shot Pulse Output (With Delay)".

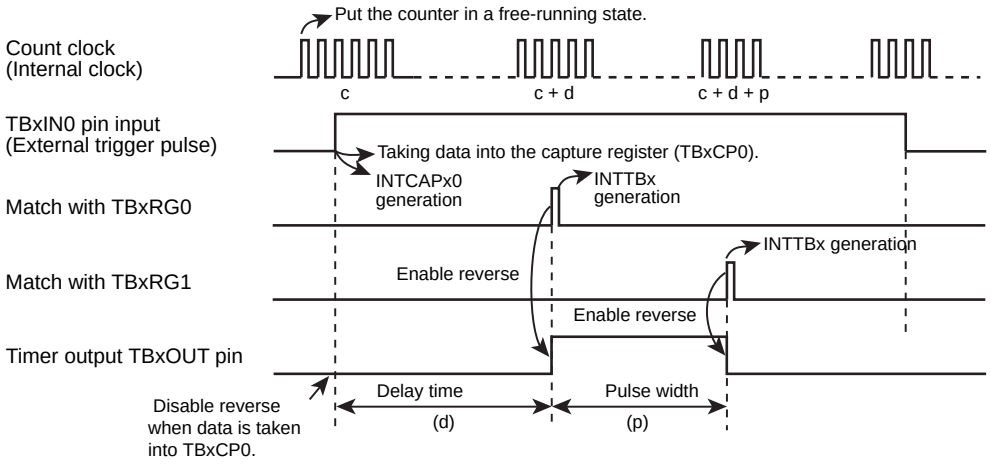


Figure 9-5 One-shot Pulse Output (With Delay)

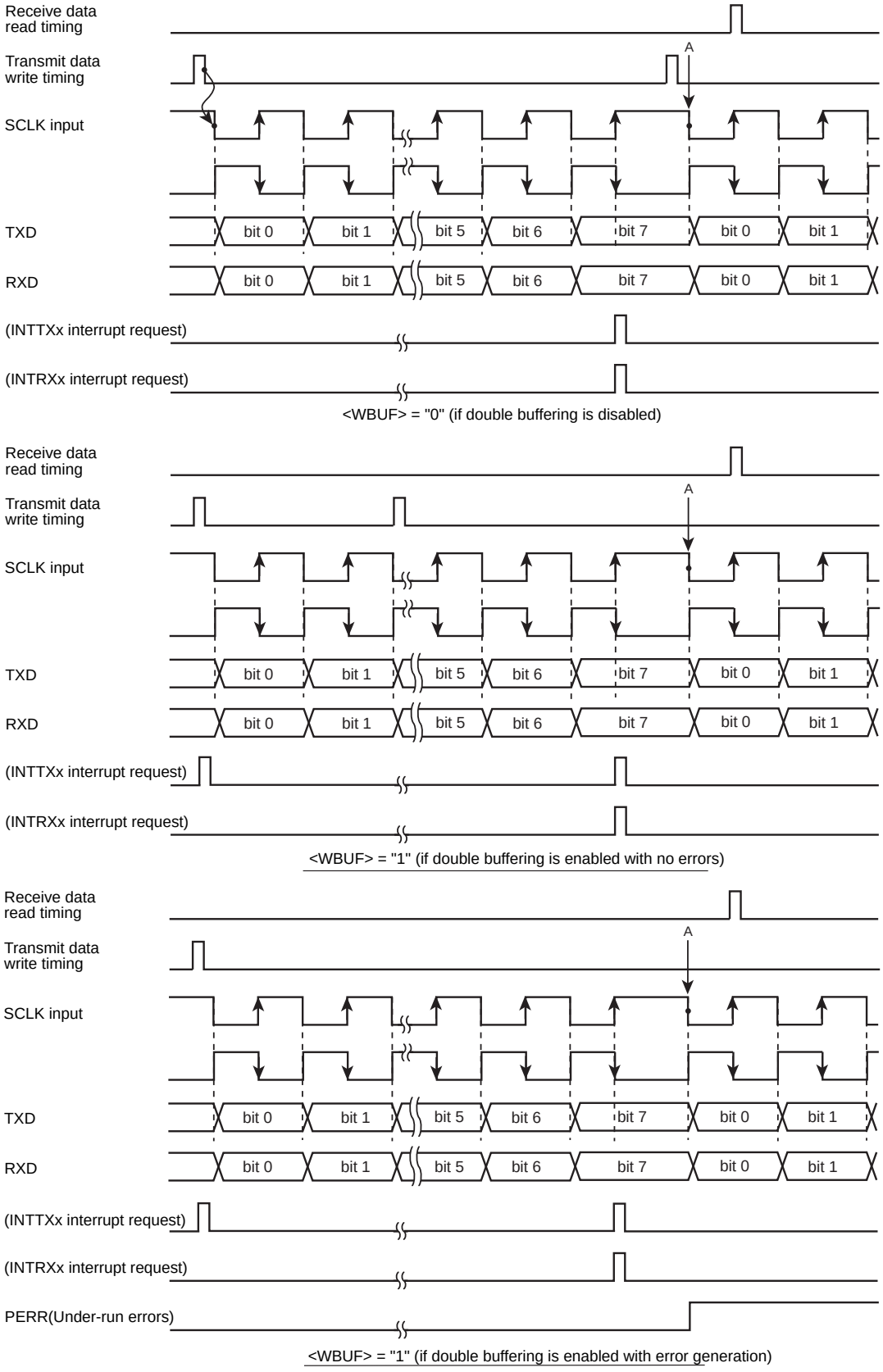


Figure 10-17 Transmit/Receive Operation in the I/O Interface Mode (SCLK Input Mode)

11.4.3 SBIxCR2(Control register 2)

This register serves as SBIxSR register by reading it.

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
After reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MST	TRX	BB	PIN	SBIM		SWRST	
After reset	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	Function
31-8	-	R	Read as 0.
7	MST	W	Select master/slave 0: Slave mode 1: Master mode
6	TRX	W	Select transmit/ receive 0: Receive 1: Transmit
5	BB	W	Start/stop condition generation 0: Stop condition generated 1: Start condition generated
4	PIN	W	Clear INTSBIx interrupt request 0: − 1: Clear interrupt request
3-2	SBIM[1:0]	W	Select serial bus interface operating mode (Note) 00: Port mode (Disables a serial bus interface output) 01: SIO mode 10: I2C bus mode 11: Reserved
1-0	SWRST[1:0]	W	Software reset generation Write "10" followed by "01" to generate a reset.

Note: **Make sure that modes are not changed during a communication session.Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "High" level before switching the operating mode from the port mode to the I2C bus or clock-synchronous 8-bit SIO mode.**

Bit	Bit Symbol	Type	Function
4	CECBRD	R/W	Broadcast transmission 0: Not broadcast transmission 1: Broadcast transmission Set this bit to "1" when transmitting a broadcast message.
3-0	CECFREE[3:0]	R/W	Time of bus to be free 0000: 1bit cycle 0001: 2bit cycle 0010: 3bit cycle 0011: 4bit cycle 0100: 5bit cycle 0101: 6bit cycle 0110: 7bit cycle 0111: 8bit cycle 1000: 9bit cycle 1001: 10bit cycle 1010: 11bit cycle 1011: 12bit cycle 1100: 13bit cycle 1101: 14bit cycle 1110: 15bit cycle 1111: 16bit cycle Specifies time of a bus to be free that checked before transmission. Start transmission after checking the CEC line kept inactive during the specified cycles.

Note: <CECDTRS> must be used under the same setting as CECRCR1<CECLNC>.

14.4.5.5 Reactivating normal AD conversion

To reactivate normal AD conversion while the conversion is underway, a software reset (ADMOD3<ADRST>) must be performed before starting AD conversion. The H/W activation method must not be used to reactivate normal AD conversion.

14.4.5.6 Conversion completion

(1) Normal AD conversion completion

When normal AD conversion is completed, the AD conversion completion interrupt (INTAD) is generated. The result of AD conversion is stored in the storage register, and two registers change: the register ADMOD0<EOCFN> which indicates the completion of AD conversion and the register ADMOD0<ADBFN>.

Interrupt request, conversion register storage register and <EOCFN><ADBFN> change with a different timing according to a mode selected.

In mode other than fixed-channel repeat conversion mode, conversion results are stored in AD conversion result registers (ADREG08 through ADRG7F) corresponding to a channel.

In fixed-channel repeat conversion mode, the conversion results are sequentially stored in storage registers ADREG08 through ADREG7F. However, if interrupt setting on <ITM> is set to be generated each time one AD conversion is completed, the conversion result is stored only in ADREG08. If interrupt setting on <ITM> is set to be generated each time four AD conversions are completed, the conversion results are sequentially stored in ADREG08H through ADREG3B.

Interrupt requests, flag changes and conversion result registers in each mode are as shown below.

- Fixed-channel single conversion mode

  After AD conversion completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is cleared to "0", and the interrupt request is generated.

  Conversion results are stored a conversion result register correspond to a channel.

- Channel scan single conversion mode

  After the channel scan conversion is completed, ADMOD0<EOCFN> is set to "1", ADMOD0<ADBFN> is set to "0", and the interrupt request INTAD is generated.

  Conversion results are stored a conversion result register correspond to a channel.

- Fixed-channel repeat conversion mode

  ADMOD0<ADBFN> is not cleared to "0". It remains at "1". The timing with which the interrupt request INTAD is generated can be selected by setting ADMOD0<ITM> to an appropriate setting. ADMOD0<EOCFN> is set with the same timing as this interrupt INTAD is generated.

  a. One conversion

     With <ITM[1:0]> set to "00", an interrupt request is generated each time one AD conversion is completed. In this case, the conversion results are always stored in the storage register ADREG08. After the conversion result is stored, <EOCFN> changes to "1".

  b. Four conversions

     With <ITM[1:0]> set to "01", an interrupt request is generated each time four AD conversions are completed. In this case, the conversion results are sequentially stored in the storage register ADREG08 through ADREG3B. After the conversion result is stored in ADREG3B, <EOCFN> is set to "1", and the storage of subsequent conversion results starts from ADREG08.

17.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 17-1 Operation Modes

Operation mode		Operation details
Single chip mode		After reset is cleared, it starts up from the internal flash memory.
	Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's card, are defined. The former is referred to as "normal mode" and the latter "user boot mode.
	User boot mode	The user can uniquely configure the system to switch between these two modes. For example, the user can freely design the system such that the normal mode is selected when the port "A0" is set to "1" and the user boot mode is selected when it is set to "0." The user should prepare a routine as part of the application program to make the decision on the selection of the modes.
Single boot mode		After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user's set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the above table, the User Boot mode and the Single Boot mode are the programmable modes. These two modes, the User Boot mode and the Single Boot mode, are referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's card.

Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\text{BOOT}}$ (PH0) pin while the device is in reset status.

Table 17-2 Operation Mode Setting

Operation mode	Pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$ (PH0)
Single chip mode	0 → 1	1
Single boot mode	0 → 1	0

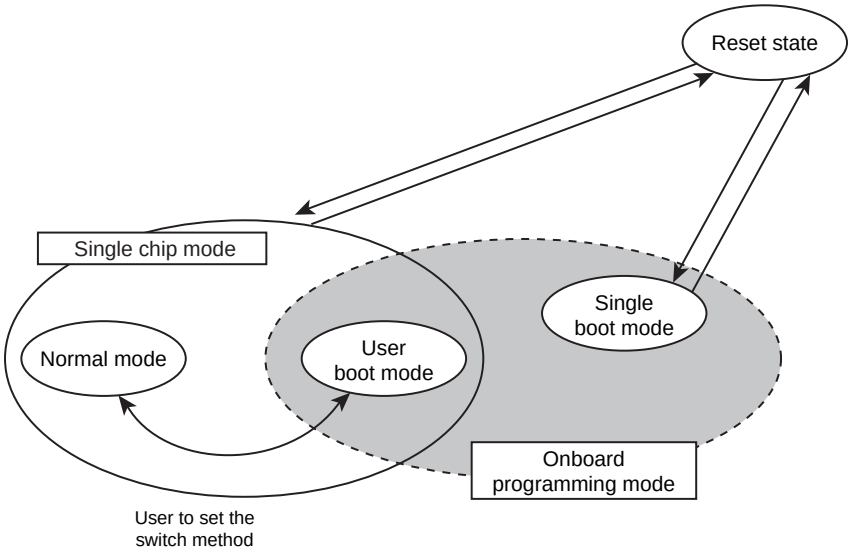
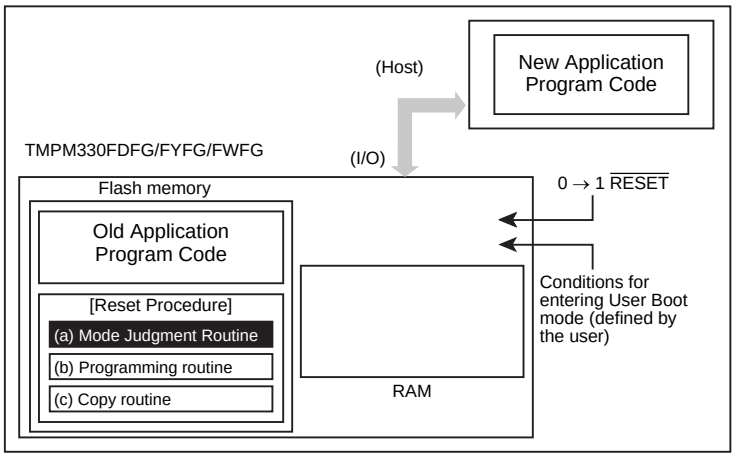


Figure 17-2 Mode Transition Diagram

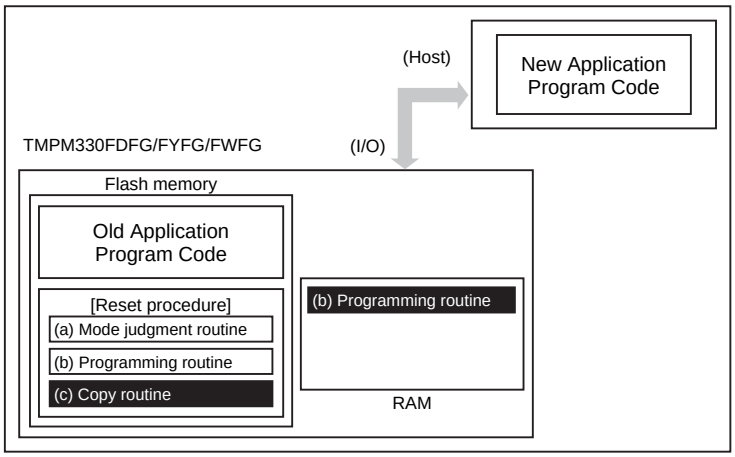
(2) Step-2

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMPM330FDFG/FYFG/FWFG flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode.)



(3) Step-3

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to the TMPM330FDFG/FYFG/FWFG on-chip RAM.



17.2.2.2 (1-B) Method 2: Transferring a Programming Routine from an External Host

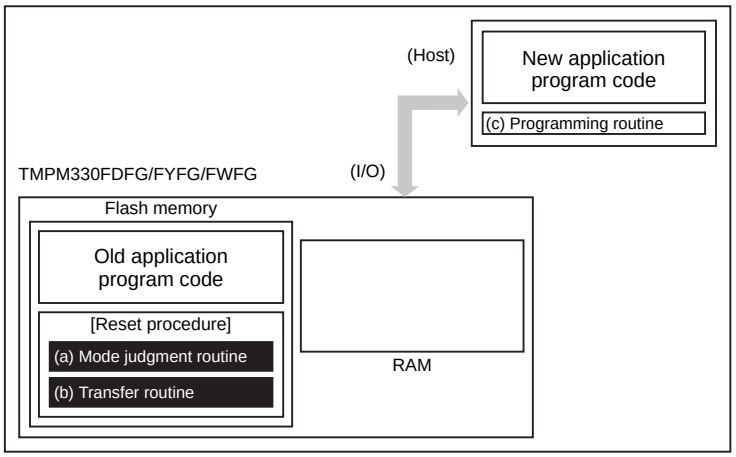
(1) Step-1

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMPM330FDFG/FYFG/FWFG on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode
(b) Transfer routine: Code to download new program code from a host controller

Also, prepare a programming routine shown below on the host controller:

(c) Programming routine: Code to download new program code from an external host controller and re-program the flash memory



2. The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Flash Memory Sum command is 0x20.

3. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

   If the 3rd byte is equal to any of the command codes listed in Table 17-4, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x20 and then branches to the Show Flash Memory Sum routine. If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the command wait state (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. The Show Flash Memory Sum routine adds all the bytes of the flash memory together. The 5th and 6th bytes, transmitted from the target board to the controller, indicate the upper and lower bytes of this total sum, respectively. For details on sum calculation, see a later section "17.2.10.8 Calculation of the Show Flash Memory Sum Command".

5. The 7th byte is a checksum value for the 5th and 6th bytes. To calculate the checksum value, add the 5th and 6th bytes together, drop the carry and take the two’s complement of the sum. Transmit this checksum value from the controller to the target board.

6. The 8th byte is the next command code.

17.2.10.3 Show Product Information Command

See Table 17-8 for the transfer format of this command.

1. The processing of the 1st and 2nd bytes are the same as for the RAM Transfer command.

2. The 3rd byte, which the target board receives from the controller, is a command. The code for the Show Product Information command is 0x30.

3. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits 0xX8 (bit 3) and returns to the command wait state again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

   If the 3rd byte is equal to any of the command codes listed in Table 17-4, the boot program echoes it back to the controller. When the Show Flash Memory Sum command was received, the boot program echoes back a value of 0x30 and then branches to the Show Flash Memory Sum routine. If the 3rd byte is not a valid command, the boot program sends back 0xX1 (bit 0) to the controller and returns to the state in which it waits for a command (the third byte) again. In this case, the upper four bits of the acknowledge response are undefined - they hold the same values as the upper four bits of the previously issued command.

4. The 5th to 8th bytes, transmitted from the target board to the controller, are the data read from addresses shown below in the flash memory. Software version management is possible by storing a software ID in these locations.

Product name	Area
TMPM330FDFG TMPM330FYFG	0x3F87_FF00 to 0x3F87_FF03
TMPM330FWFG	0x3F81_FF00 to 0x3F81_FF03

number of flash blocks available. Each product transmits own number shown below.

Product name	Number of flash blocks
TMPM330FDFG	0x06, 0x00
TMPM330FYFG	0x06, 0x00
TMPM330FWFG	0x04, 0x00

15. The 57th to 83rd bytes, transmitted from the target board to the controller, contain information about the flash blocks. Flash blocks of the same size are treated as a group. Information about the flash blocks indicate the start address of a group, the size of the blocks in that group (in halfwords) and the number of the blocks in that group. The 57th to 65th bytes are the information about the 16-kbyte blocks. The 66th to 74th bytes are the information about the 32-kbyte blocks. The 75th to 83rd bytes are the information about the 64-kbyte blocks. The 84th to 92nd bytes are the information about the 128-kbyte blocks. See Table 17-8 for the values of bytes transmitted.
16. The 66th byte, transmitted from the target board to the controller, is a checksum value for the 5th to 92nd bytes. The checksum value is calculated by adding all these bytes together, dropping the carry and taking the two’s complement of the total sum.
17. The 94th byte is the next command code.

19.9 Handling Precaution

19.9.1 Solderability

Test parameter	Test condition	Note
Solderability	Use of Sn-37Pb solder Bath Solder bath temperature = 230°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: solderability rate until forming ≥ 95%
	Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	

19.9.2 Power-on sequence

The power supply must be raised (from 0V to 2.7V) at a speed of 0.37ms/V or slower. The power-on sequence must consider the time for the internal regulator and oscillator to be stableIn the .In the TX03, the internal regulator requires at least 700 μs to be stable.

The time required to achieve stable oscillation varies with system. At cold reset, the external reset pin must be kept "Low" for a duration of time sufficiently long enough for the internal regulator and oscillator to be stable.

Figure 19-2 shows the power-on sequence.

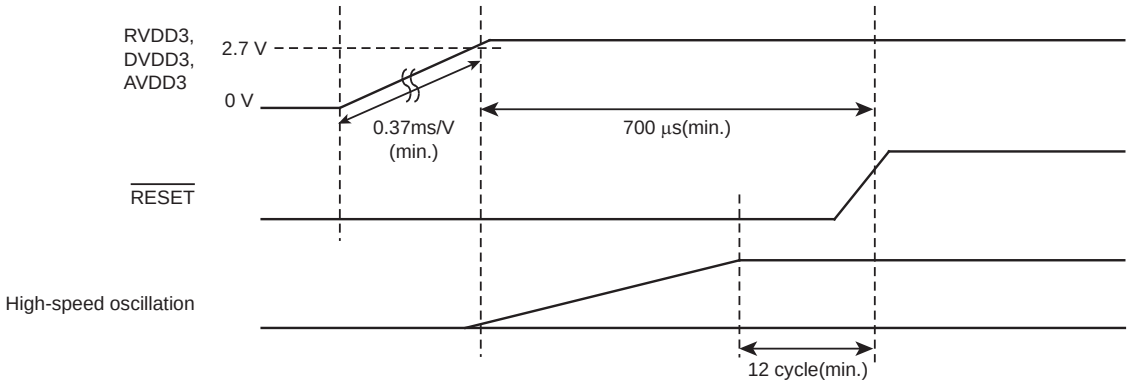


Figure 19-2 Power-on sequence