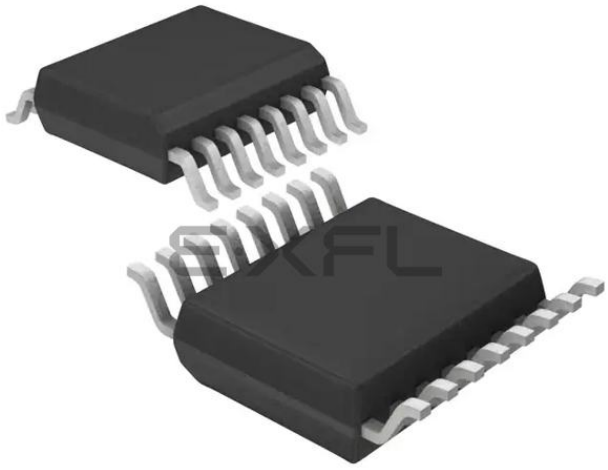
Renesas Electronics America Inc - UPD78F9211MA-FAA-AX Datasheet



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	Obsolete
Core Processor	78K0S
Core Size	8-Bit
Speed	10MHz
Connectivity	-
Peripherals	LVD, POR, PWM, WDT
Number of I/O	14
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-SSOP (0.173", 4.40mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f9211ma-faa-ax

2.2.2 P32 and P34 (Port 3)

P32 is a 1-bit I/O port. In addition to the function as an I/O port pin, this pin also has a function to input an external interrupt request signal.

P34 is a 1-bit input-only port. This pin is also used as a $\overline{\text{RESET}}$ pin, and when the power is turned on, this is the reset function.

For the setting method for pin functions, see **CHAPTER 15 OPTION BYTE**.

When P34 is used as an input port pin, connect the pull-up resistor.

P32 and P34 can be set to the following operation modes in 1-bit units.

(1) Port mode

P32 functions as a 1-bit I/O port. This pin can be set to the input or output mode by using port mode register 3 (PM3). In addition, an on-chip pull-up resistor can be connected to the port by using pull-up resistor option register 3 (PU3).

P34 functions as a 1-bit input-only port.

(2) Control mode

P32 functions as an external interrupt request input pin (INTP1) for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

2.2.3 P40 to P47 (Port 4)

P40 to P47 constitute an 8-bit I/O port. Each bit of this port can be set to the input or output mode by using port mode register 4 (PM4). In addition, an on-chip pull-up resistor can be connected to the port by using pull-up resistor option register 4 (PU4).

2.2.4 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal. When the power is turned on, this is the reset function, regardless of the option byte setting.

2.2.5 X1 and X2

These pins connect an oscillator to oscillate the X1 input clock.

X1 and X2 also function as the P23/ANI3 and P22/ANI2, respectively. For the setting method for pin functions, see **CHAPTER 15 OPTION BYTE**.

Supply an external clock to X1.

Caution The P22/X2/ANI2 and P23/X1/ANI3 pins are pulled down during reset.

2.2.6 V_{DD}

This is the positive power supply pin.

In μPD78F921x, V_{DD} functions alternately as the A/D converter reference voltage input. When using the A/D converter, stabilize V_{DD} at the supply voltage used (2.7 to 5.5 V).

2.2.7 V_{SS}

This is the ground pin.

In μPD78F921x, V_{SS} functions alternately as the ground potential of the A/D converter. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

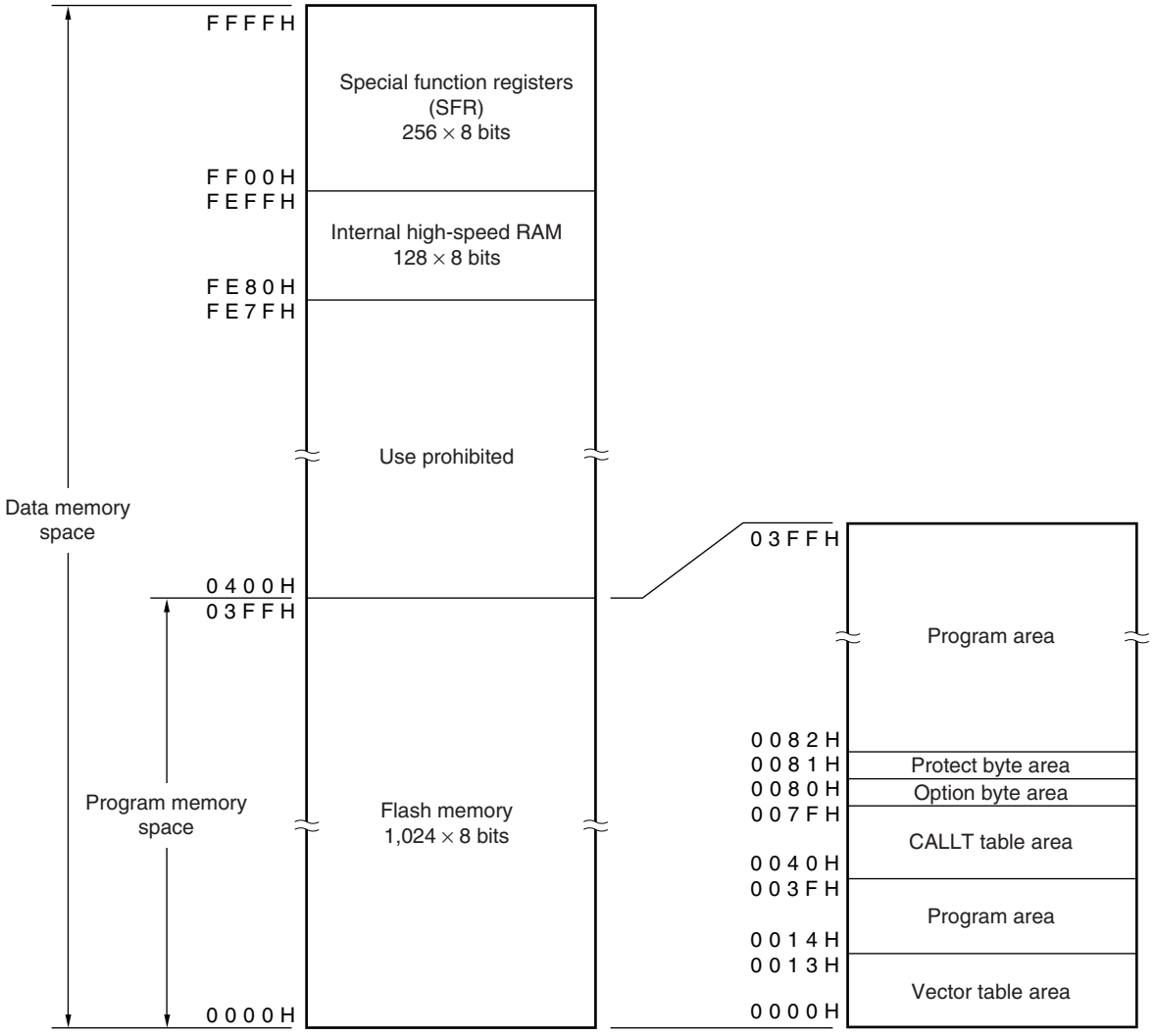
CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

The 78K0S/KY1+ can access up to 64 KB of memory space. Figures 3-1 to 3-3 show the memory maps.

<R>

Figure 3-1. Memory Map (μPD78F9210, 78F9510)



Remark The option byte and protect byte are 1 byte each.

<R> **Table 3-3. Special Function Registers (2/3)**

Address	Symbol	Bit No.								R/W	Number of Bits Manipulated Simultaneously			After Reset	Reference page
		7	6	5	4	3	2	1	0		1	8	16		
FF50H	LVIM	<LVION>	0	0	0	0	0	<LVIMD>	<LVIF>		√	√	–	00H Note 1	210
FF51H	LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0		–	√	–	00H Note 1	211
FF52H, FF53H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF54H	RESF	0	0	0	WDTRF	0	0	0	LVIRF	R	–	√	–	00H Note 2	204
FF55H to FF57H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF58H	LSRCM	0	0	0	0	0	0	0	<LSRSTOP>	R/W	√	√	–	00H	77
FF59H to FF5FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF60H	TMC00	0	0	0	0	TMC003	TMC002	TMC001	<OVF00>	R/W	√	√	–	00H	95
FF61H	PRM00	ES110	ES100	ES010	ES000	0	0	PRM001	PRM000		√	√	–	00H	99
FF62H	CRC00	0	0	0	0	0	CRC002	CRC001	CRC000		√	√	–	00H	97
FF63H	TOC00	0	<OSPT00>	<OSPE00>	TOC004	<LVS00>	<LVR00>	TOC001	<TOE00>		√	√	–	00H	98
FF64H to FF6FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF70H	TMHMD1	<TMHE1>	CKS12	CKS11	CKS10	TMMD11	TMMD10	<TOLEV1>	<TOEN1>	R/W	√	√	–	00H	134
FF71H to FF7FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF80H	ADM Note 3	<ADCS>	0	FR2	FR1	FR0	0	0	<ADCE>	R/W	√	√	–	00H	161
FF81H	ADS Note 3	0	0	0	0	0	0	ADS1	ADS0		√	√	–	00H	164
FF82H, FF83H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF84H	PMC2 Note 3	0	0	0	0	PMC23	PMC22	PMC21	PMC20	R/W	√	√	–	00H	69, 100, 136, 165
FF85H to FF9FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Notes 1. Retained only after a reset by LVI.

2. Varies depending on the reset cause.

3. μPD78F921x only

Remark For a bit name enclosed in angle brackets (<>), the bit name is defined as a reserved word in the RA78K0S, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0S.

<R>

Table 3-3. Special Function Registers (3/3)

Address	Symbol	Bit No.								R/W	Number of Bits Manipulated Simultaneously			After Reset	Reference page
		7	6	5	4	3	2	1	0		1	8	16		
FFA0H	PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	W	–	√	–	Undefined	237
FFA1H	PFS	0	0	0	0	0	WEPRERR	VCERR	FPRERR	R/W	√	√	–	00H	237
FFA2H	FLPMC	0	PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	0	FLSPM		–	√	–	Undefined	236
FFA3H	FLCMD	0	0	0	0	0	FLCMD2	FLCMD1	FLCMD0		√	√	–	00H	239
FFA4H	FLAPL	FLAP7	FLAP6	FLAP5	FLAP4	FLAP3	FLAP2	FLAP1	FLAP0		√	√	–	Undefined	240
FFA5H	FLAPH	0	0	0	0	FLAP11	FLAP10	FLAP9	FLAP8		√	√	–		240
FFA6H	FLAPHC	0	0	0	0	FLAPC11	FLAPC10	FLAPC9	FLAPC8		√	√	–	00H	240
FFA7H	FLAPLC	FLAPC7	FLAPC6	FLAPC5	FLAPC4	FLAPC3	FLAPC2	FLAPC1	FLAPC0		√	√	–		240
FFA8H	FLW	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0		–	√	–		241
FFA9H to FFDFH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFE0H	IF0	<ADIF> Note 1	<TMIF010>	<TMIF000>	<TMIFH1>	<PIF1>	<PIF0>	<LVIIF>	0	R/W	√	√	–	00H	180
FFE1H to FFE3H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFE4H	MK0	<ADMK> Note 1	<TMMK010>	<TMMK000>	<TMMKH1>	<PMK1>	<PMK0>	<LVIMK>	1	R/W	√	√	–	FFH	181
FFE5H to FFEBH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFECH	INTM0	0	0	ES11	ES10	ES01	ES00	0	0	R/W	–	√	–	00H	181
FFEDH to FFF2H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFF3H	PPCC	0	0	0	0	0	0	PPCC1	PPCC0	R/W	√	√	–	02H	76
FFF4H	OSTS	0	0	0	0	0	0	OSTS1	OSTS0		–	√	–	Undefined Note 2	78, 189
FFF5H to FFFAH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFFBH	PCC	0	0	0	0	0	0	PCC1	0	R/W	√	√	–	02H	76
FFFCH to FFFFH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Notes 1. μPD78F921x only

2. The oscillation stabilization time that elapses after release of reset is selected by the option byte. For details, refer to **CHAPTER 15 OPTION BYTE**.

Remark For a bit name enclosed in angle brackets (<>), the bit name is defined as a reserved word in the RA78K0S, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0S.

6.2 Configuration of 16-bit Timer/Event Counter 00

16-bit timer/event counter 00 consists of the following hardware.

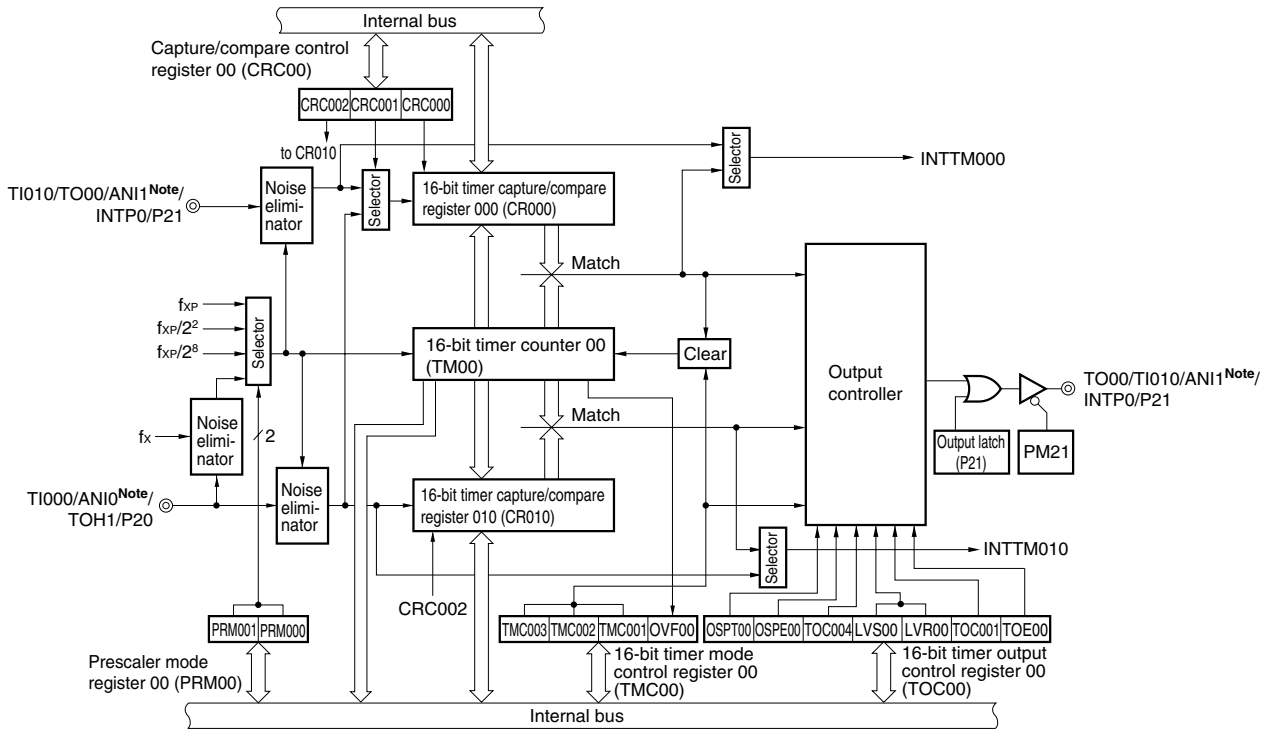
Table 6-1. Configuration of 16-bit Timer/Event Counter 00

Item	Configuration
Timer counter	16-bit timer counter 00 (TM00)
Register	16-bit timer capture/compare registers 000, 010 (CR000, CR010)
Timer input	TI000, TI010
Timer output	TO00, output controller
Control registers	16-bit timer mode control register 00 (TMC00) Capture/compare control register 00 (CRC00) 16-bit timer output control register 00 (TOC00) Prescaler mode register 00 (PRM00) Port mode register 2 (PM2) Port register 2 (P2) Port mode control register 2 (PMC2) (μPD78F921x only)

<R>

Figure 6-1 shows a block diagram of these counters.

Figure 6-1. Block Diagram of 16-bit Timer/Event Counter 00



<R> **Note** μPD78F921x only

Caution 6. Changing the CR010 setting during TM00 operation may cause a malfunction. To change the setting, refer to 6.5 Cautions Related to 16-bit Timer/Event Counter 00 (17) Changing compare register during timer operation.

6.3 Registers to Control 16-bit Timer/Event Counter 00

The following seven types of registers are used to control 16-bit timer/event counter 00.

- 16-bit timer mode control register 00 (TMC00)
- Capture/compare control register 00 (CRC00)
- 16-bit timer output control register 00 (TOC00)
- Prescaler mode register 00 (PRM00)
- Port mode register 2 (PM2)
- Port register 2 (P2)
- <R> Port mode control register 2 (PMC2) (μPD78F921x only)

(1) 16-bit timer mode control register 00 (TMC00)

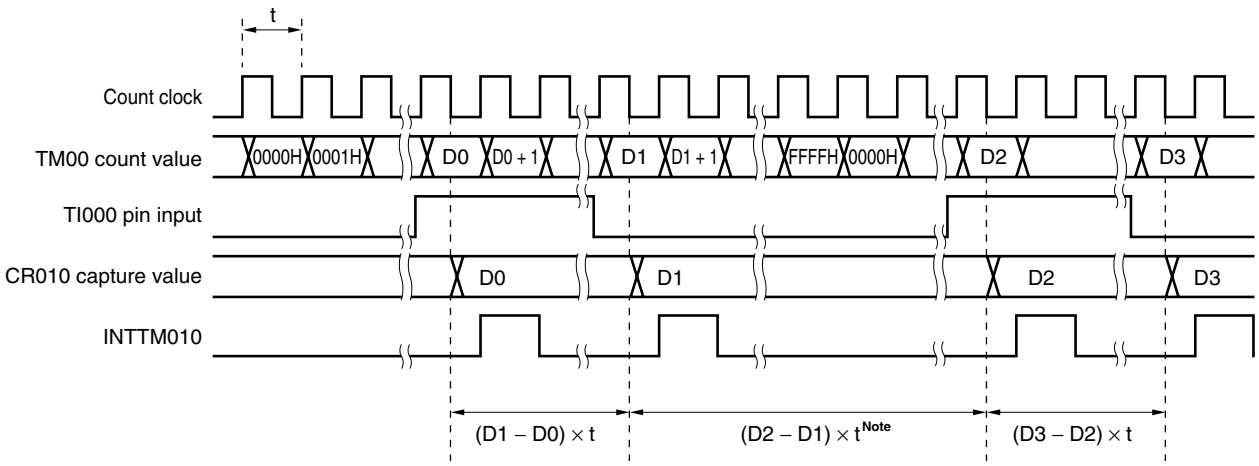
This register sets the 16-bit timer operating mode, the 16-bit timer counter 00 (TM00) clear mode, and output timing, and detects an overflow.

TMC00 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets the value of TMC00 to 00H.

Caution 16-bit timer counter 00 (TM00) starts operating as soon as values other than 0 and 0 (operation stop mode) are set to TMC002 and TMC003, respectively. Set TMC002 and TMC003 to 0 and 0 to stop operation.

Figure 6-21. Timing of Pulse Width Measurement Operation by Free-Running Counter and One Capture Register (with Both Edges Specified)



Note The carry flag is set to 1. Ignore this setting.

(2) Measurement of two pulse widths with free-running counter

When 16-bit timer counter 00 (TM00) is operated in free-running mode, it is possible to simultaneously measure the pulse widths of the two signals input to the TI000 pin and the TI010 pin.

Specify both the rising and falling edges as the valid edges of the TI000 and TI010 pins, by using bits 4 and 5 (ES000 and ES010) and bits 6 and 7 (ES100 and ES110) of PRM00.

When the valid edge specified by bits 4 and 5 (ES000 and ES010) of prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 010 (CR010) and an interrupt request signal (INTTM010) is set.

Also, when the valid edge specified by bits 6 and 7 (ES100 and ES110) of PRM00 is input to the TI010 pin, the value of TM00 is taken into 16-bit timer capture/compare register 000 (CR000) and an interrupt request signal (INTTM000) is set.

Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00), and a capture operation is only performed when a valid level of the TI000 or TI010 pin is detected twice, thus eliminating noise with a short pulse width.

Caution The measurable pulse width in this operation example is up to 1 cycle of the timer counter.

<3> Do not set the 16-bit timer capture/compare registers 000 and 010 (CR000 and CR010) to 0000H.

(12) One-shot pulse output with external trigger

<1> Do not input the external trigger again while the one-shot pulse is output.
To output the one-shot pulse again, wait until the current one-shot pulse output is completed.

<2> Do not set the 16-bit timer capture/compare registers 000 and 010 (CR000 and CR010) to 0000H.

(13) Operation of OVF00 flag

<1> The OVF00 flag is also set to 1 in the following case.
Either of the clear & start mode entered on a match between TM00 and CR000, clear & start at the valid edge of the TI000 pin, or free-running mode is selected.

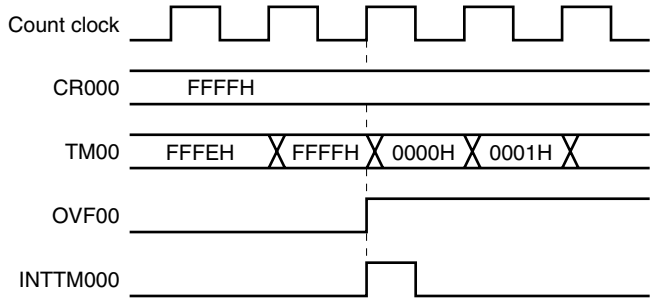
↓

CR000 is set to FFFFH.

↓

When TM00 is counted up from FFFFH to 0000H.

Figure 6-38. Operation Timing of OVF00 Flag



<2> Even if the OVF00 flag is cleared before the next count clock is counted (before TM00 becomes 0001H) after the occurrence of a TM00 overflow, the OVF00 flag is re-set and clearance becomes invalid.

(14) Conflicting operations

If the register read period and the input of the capture trigger conflict when CR000/CR010 is used as a capture register, the capture trigger input takes precedence and the read data is undefined. Also, if the count stop of the timer and the input of the capture trigger conflict, the captured data is undefined.

(2) Port mode register 2 (PM2) and port mode control register 2 (PMC2)[Note]

When using the P20/TOH1/TI000/ANI0 pin for timer output, clear PM20, the output latch of P20, and PMC20 to 0.

PM2 and PMC2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM2 to FFH, and clears PMC2 to 00H.

<R> **Note** μPD78F921x only

Figure 7-5. Format of Port Mode Register 2 (PM2)

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

PM2n	P2n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 7-6. Format of Port Mode Control Register 2 (PMC2) (μPD78F921x only)

Address: FF84H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	PMC23	PMC22	PMC21	PMC20

PMC2n	Specification of operation mode (n = 0 to 3)
0	Port/Alternate-function (except A/D converter) mode
1	A/D converter mode

7.4 Operation of 8-bit Timer H1

7.4.1 Operation as interval timer/square-wave output

When 8-bit timer counter H1 and compare register 01 (CMP01) match, an interrupt request signal (INTTMH1) is generated and 8-bit timer counter H1 is cleared to 00H.

Compare register 11 (CMP11) is not used in interval timer mode. Since a match of 8-bit timer counter H1 and the CMP11 register is not detected even if the CMP11 register is set, timer output is not affected.

By setting bit 0 (TOEN1) of timer H mode register 1 (TMHMD1) to 1, a square wave of any frequency (duty = 50%) is output from TOH1.

9.5 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 9-13. Overall Error

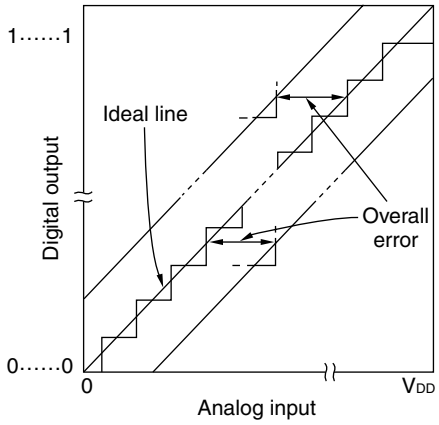
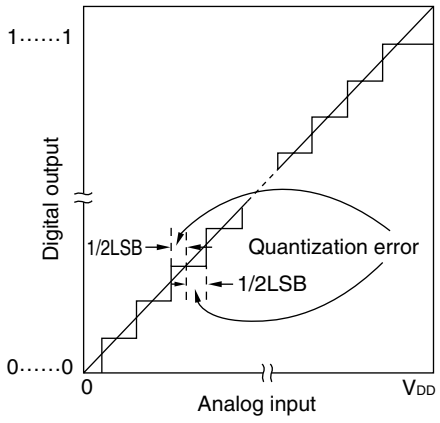


Figure 9-14. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

13.4 Cautions for Power-on-Clear Circuit

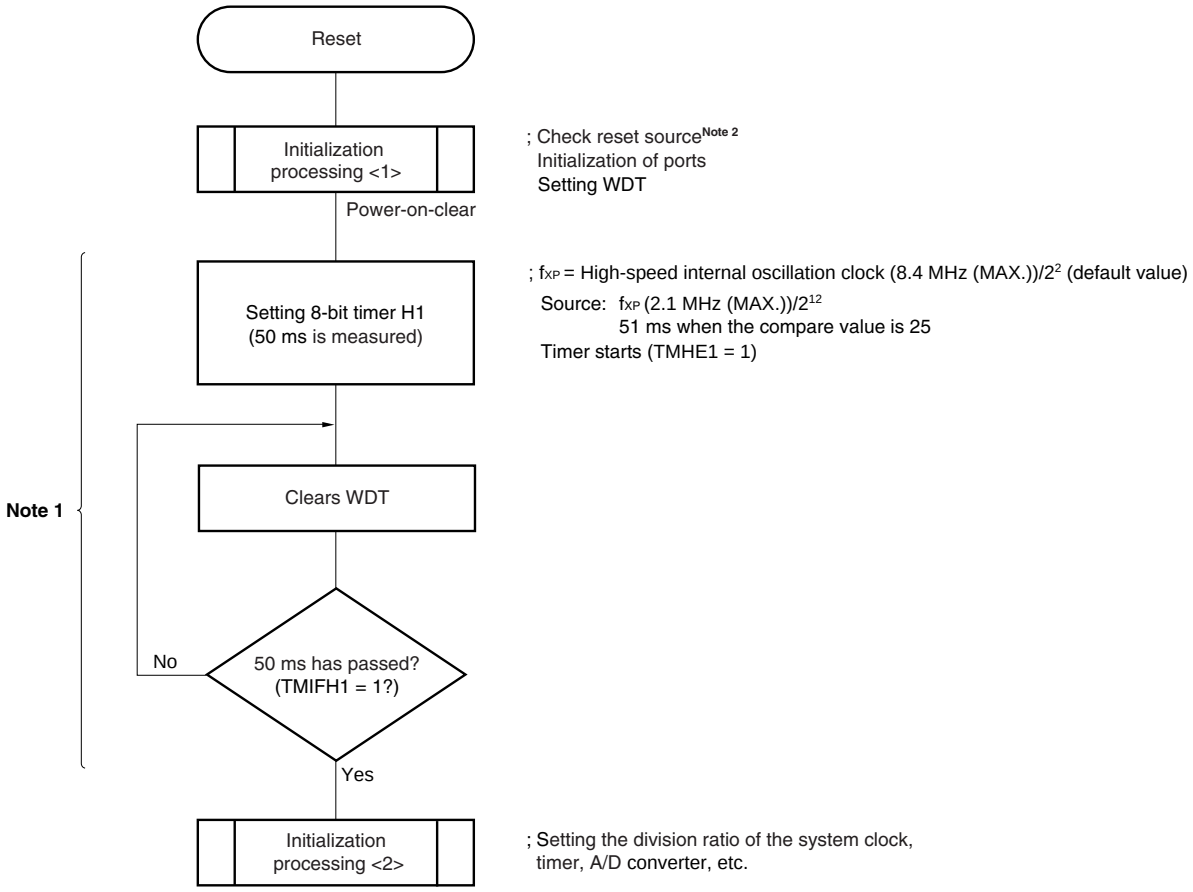
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 13-3. Example of Software Processing After Release of Reset (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage



Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

16.8.2 Cautions on self programming function

- No instructions can be executed while a self programming command is being executed. Therefore, clear and restart the watchdog timer counter in advance so that the watchdog timer does not overflow during self programming. Refer to Table 16-10 for the time taken for the execution of self programming.
- Interrupts that occur during self programming can be acknowledged after self programming mode ends. To avoid this operation, disable interrupt servicing (by setting MK0 to FFH, and executing the DI instruction) when shifting from the normal mode to the self programming mode with a specific sequence.
- RAM is not used while a self programming command is being executed.
- If the supply voltage drops or the reset signal is input while the flash memory is being written or erased, writing/erasing is not guaranteed.
- The value of the blank data set during block erasure is FFH.
- Set the CPU clock beforehand so that it is 1 MHz or higher during self programming.
- Execute self programming after executing the NOP and HALT instructions immediately after executing a specific sequence to set self programming mode. At this time, the HALT instruction is automatically released after 10 μs (MAX.) + 2 CPU clocks (f_{CPU}).
- If the clock of the oscillator or an external clock is selected as the system clock, execute the NOP and HALT instructions immediately after executing a specific sequence to set self programming mode, wait for 8 μs after releasing the HALT status, and then execute self programming.
- Check FPRERR using a 1-bit memory manipulation instruction.
- The state of the pins in self programming mode is the same as that in HALT mode.
- Since the security function set via on-board/off-board programming is disabled in self programming mode, the self programming command can be executed regardless of the security function setting. To disable write or erase processing during self programming, set the protect byte.
- Be sure to clear bits 4 to 7 of flash address pointer H (FLAPH) and flash address pointer H compare register (FLAPHC) to 0 before executing the self programming command. If self programming is executed with these bits set to 1, the device may malfunction.
- Clear the value of the FLCMD register to 00H immediately before setting to self programming mode and normal mode.

16.8.3 Registers used for self programming function

The following registers are used for the self programming function.

- Flash programming mode control register (FLPMC)
- Flash protect command register (PFCMD)
- Flash status register (PFS)
- Flash programming command register (FLCMD)
- Flash address pointers H and L (FLAPH and FLAPL)
- Flash address pointer H compare register and flash address pointer L compare register (FLAPHC and FLAPLC)
- Flash write buffer register (FLW)

The 78K0S/KY1+ has an area called a protect byte at address 0081H of the flash memory.

(2) Flash protect command register (PFCMD)

If the application system stops inadvertently due to malfunction caused by noise or program hang-up, an operation to write the flash programming mode control register (FLPMC) may have a serious effect on the system. PFCMD is used to protect FLPMC from being written, so that the application system does not stop inadvertently.

Writing FLPMC is enabled only when a write operation is performed in the following specific sequence.

<1> Write a specific value to PFCMD (A5H)

<2> Write the value to be set to bit 0 (FLSPM) of the FLPMC (writing in this step is invalid)

<3> Write the inverted value of the value to be set to bit 0 (FLSPM) of the FLPMC (writing in this step is invalid)

<4> Write the value to be set to bit 0 (FLSPM) of the FLPMC (writing in this step is valid)

Caution Interrupt servicing cannot be executed in self programming mode. Disable interrupt servicing (by executing the DI instruction while MK0 = FFH) between the points before executing the specific sequence that sets self programming mode and after executing the specific sequence that changes the mode to the normal mode.

This rewrites the value of the register, so that the register cannot be written illegally.

Occurrence of an illegal write operation can be checked by bit 0 (FPRERR) of the flash status register (PFS). Check FPRERR using a 1-bit memory manipulation instruction.

A5H must be written to PFCMD each time the value of FLPMC is changed.

PFCMD can be set by an 8-bit memory manipulation instruction.

Reset signal generation makes PFCMD undefined.

Figure 16-11. Format of Flash Protect Command Register (PFCMD)

Address: FFA0H After reset: Undefined W

Symbol	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) Flash status register (PFS)

If data is not written to the flash programming mode control register (FLPMC), which is protected, in the correct sequence (writing the flash protect command register (PFCMD)), FLPMC is not written and a protection error occurs. If this happens, bit 0 of PFS (FPRERR) is set to 1.

When FPRERR is 1, it can be cleared to 0 by writing 0 to it.

Errors that may occur during self programming are reflected in bit 1 (VCERR) and bit 2 (WEPRERR) of PFS. VCERR or WEPRERR can be cleared by writing 0 to them.

All the flags of the PFS register must be pre-cleared to 0 to check if the operation is performed correctly.

PFS can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears PFS to 00H.

Caution Check FPRERR using a 1-bit memory manipulation instruction.

16.8.6 Example of block erase operation in self programming mode

An example of the block erase operation in self programming mode is explained below.

<1> Set 03H (block erase) to the flash program command register (FLCMD).
<2> Set the block number to be erased, to flash address pointer H (FLAPH).
<3> Set flash address pointer L (FLAPL) to 00H.
<4> Write the same value as FLAPH to the flash address pointer H compare register (FLAPHC).
<5> Set the flash address pointer L compare register (FLAPLC) to 00H.
<6> Clear the flash status register (PFS).
<7> Write ACH to the watchdog timer enable register (WDTE) (clear and restart the watchdog timer counter)[Note 1].
<8> Execute the HALT instruction then start self programming. (Execute an instruction immediately after the HALT instruction if self programming has been executed.)
<9> Check if a self programming error has occurred using bit 1 (VCERR) and bit 2 (WEPRERR) of PFS[Note 2].
Abnormal → <10>
Normal → <12>
<R> <10> If the number of times the erase command can be executed has not been exceeded, return to step <6> and re-execute the command. If the number of times the erase command can be executed has been exceeded, block erasure ends abnormally.
<11> Block erase processing is abnormally terminated.
<12> Block erase processing is normally terminated.

Notes 1. This setting is not required when the watchdog timer is not used.
<R> **2.** Separately check the WEPRERR bit to check for errors in executing the erase command on a write-prohibited area.

An example of a program that performs a block blank check in self programming mode is shown below.

```
;----------------------------
;START
;----------------------------

FlashBlockBlankCheck:
       MOV      FLCMD,#04H      ; Sets flash control command (block blank check)
       MOV      FLAPH,#07H      ; Sets number of block for blank check (block 7 is specified
                                ; here)
       MOV      FLAPL,#00H      ; Fixes FLAPL to "00H"
       MOV      FLAPHC,#07H     ; Sets blank check block compare number (same value as that of
                                ; FLAPH)
       MOV      FLAPLC,#0FFH    ; Fixes FLAPLC to "FFH"

       MOV      PFS,#00H        ; Clears flash status register
       MOV      WDTE,#0ACH      ; Clears & restarts WDT
       HALT                     ; Self programming is started
       MOV      A,PFS
       MOV      CmdStatus,A     ; Execution result is stored in variable
                                ; (CmdStatus = 0: normal termination, other than 0: abnormal
                                ; termination)

;----------------------------
;END
;----------------------------
```

(2) Write to internal verify

<1> Mode is shifted from normal mode to self programming mode (<1> to <7> in **16.8.4**)

<2> Specification of source data for write

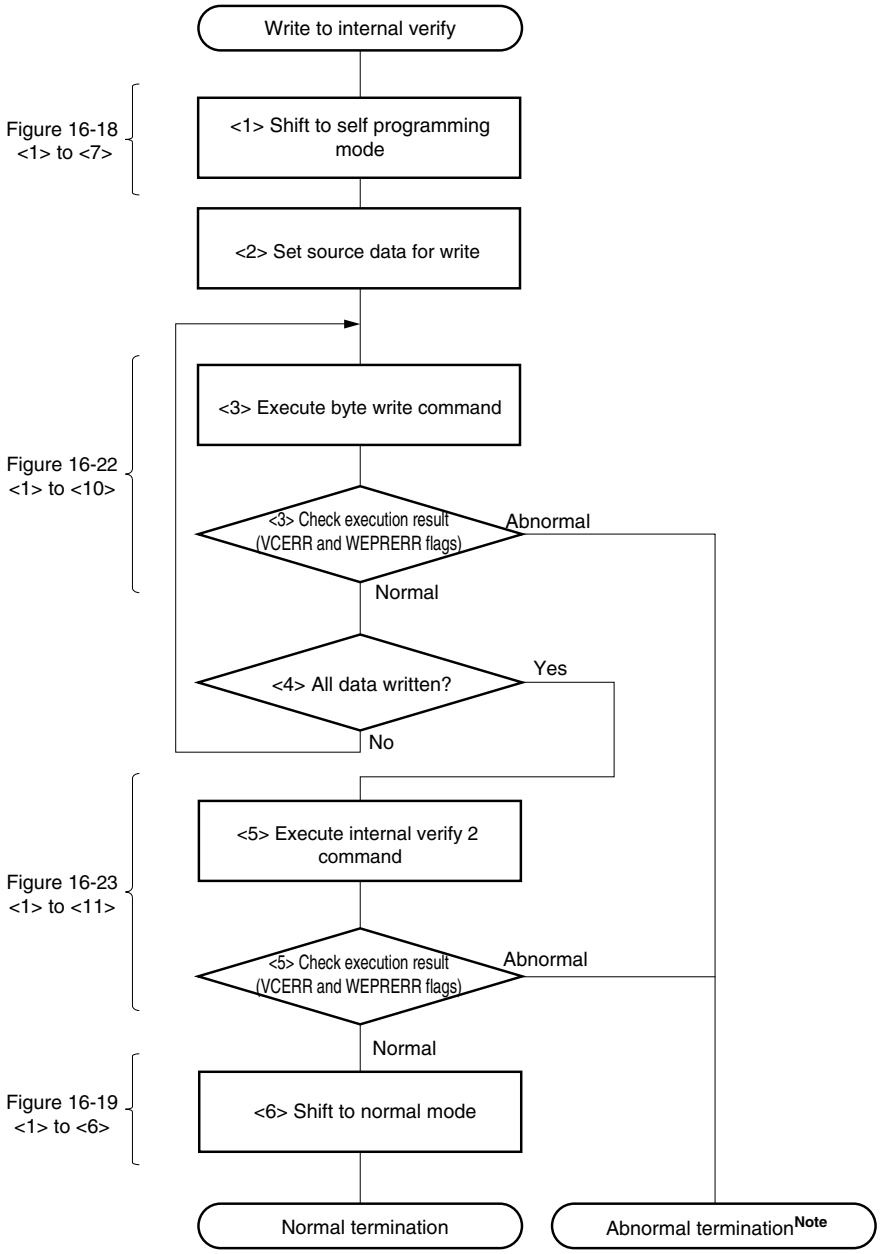
<3> Execution of byte write → Error check (<1> to <10> in **16.8.8**)

<4> <3> is repeated until all data are written.

<5> Execution of internal verify 2 → Error check (<1> to <11> in **16.8.9**)

<6> Mode is shifted from self programming mode to normal mode (<1> to <6> in **16.8.5**)

Figure 16-26. Example of Operation When Command Execution Time Should Be Minimized (from Write to Internal Verify)



Note Perform processing to shift to normal mode in order to return to normal processing.

Remark <1> to <6> in Figure 16-26 correspond to <1> to <6> in **16.8.10 (2)** above.

```
        INCW    DE              ; Address at which data is to be written + 1
        BR      FlashWriteLoop

FlashVerify:
        MOVW    HL,#WriteAdr    ; Sets verify address

        MOV     FLCMD,#02H      ; Sets flash control command (internal verify 2)
        MOV     A,H
        MOV     FLAPH,A         ; Sets verify start address
        MOV     A,L
        MOV     FLAPL,A         ; Sets verify start address
        MOV     A,D
        MOV     FLAPHC,A        ; Sets verify end address
        MOV     A,E
        MOV     FLAPLC,A        ; Sets verify end address

        MOV     WDTE,#0ACH      ; Clears & restarts WDT
        HALT                    ; Self programming is started
        MOV     A,PFS
        CMP     A,#00H
        BNZ     $StatusError    ; Checks internal verify error
                                ; Performs abnormal termination processing when an error
                                ; occurs.

        MOV     FLCMD,#00H      ; Clears FLCMD register
ModeOffLoop:
        MOV     PFS,#00H        ; Clears flash status register
        MOV     PFCMD,#0A5H     ; PFCMD register control
        MOV     FLPMC,#00H      ; FLPMC register control (sets value)
        MOV     FLPMC,#0FFH     ; FLPMC register control (inverts set value)
        MOV     FLPMC,#00H      ; Sets normal mode via FLPMC register control (sets value)

        BT      PFS.0,$ModeOffLoop ; Checks completion of write to specific registers
                                   ; Repeats the same processing when an error occurs
                                   ; Restore the CPU clock to its setting before the self
                                   ; programming, after normal completion of the specific
                                   ; sequence
        MOV     MK0,#INT_MK0    ; Restores interrupt mask flag

        EI

        BR      StatusNormal

;---------------------------------------------------------------------
;END (abnormal termination processing); Perform processing to shift to
      normal mode in order to return to normal processing
;---------------------------------------------------------------------
StatusError:
```

Note 5. The INTP1 pin is used for communication between QB-MINI2 and the target device during debugging. When debugging is performed with QB-MINI2, therefore, the INTP1 pin and its alternate-function pin cannot be used. For INTP1 pin connection, refer to **17.1.1 Connection of INTP1 pin**.

Pins for communication depend on whether the monitor program has been written or not. (refer to **Table 17-1**) X1 and X2 pins can be used as I/O port pins or the pins for oscillation, after the monitor program has been written.

Table 17-1. Pins for communication with QB-MINI2

Before writing the monitor program	After writing the monitor program
X1, X2, $\overline{\text{RESET}}$, INTP1, V_{DD}, V_{SS}	$\overline{\text{RESET}}$, INTP1, V_{DD}, V_{SS}

17.1.1 Connection of INTP1 pin

The INTP1 pin is used only for communication between QB-MINI2 and the target device during debugging. Design circuits appropriately according to the relevant case among the cases shown below.

(1) INTP1 pin is not used in target system (as is illustrated in **Figure 17-1**. **Recommended Circuit Connection**)
→ See **Figure 17-2**.

(2) QB-MINI2 is used only for programming, not for debugging
→ See **Figure 17-3**.

(3) QB-MINI2 is used for debugging and debugging of the INTP1 pin is performed only with a real machine
→ See **Figure 17-4**.

Figure 17-2. Circuit Connection for the Case Where INTP1 Pin Is Not Used in Target System

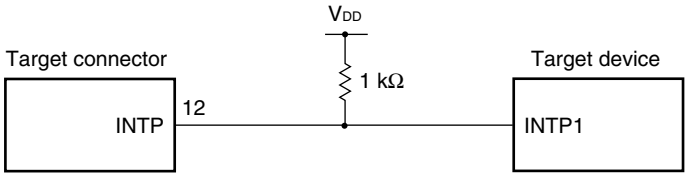
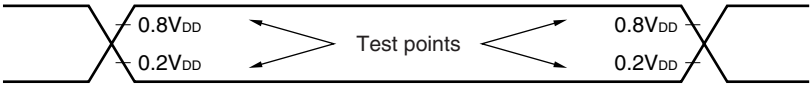


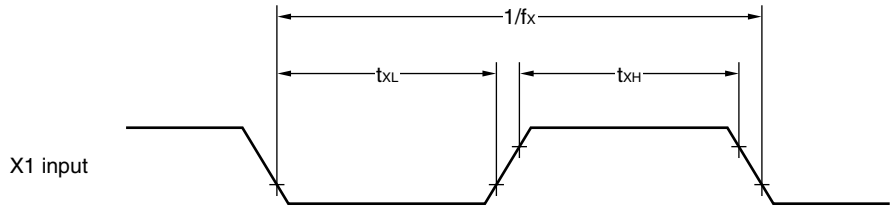
Figure 17-3. Circuit Connection for the Case Where QB-MINI2 Is Used Only for Programming

(A2) grade product T_A = –40 to +125°C

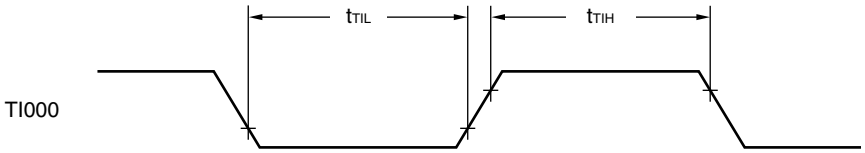
AC Timing Test Points (Excluding X1 Input)



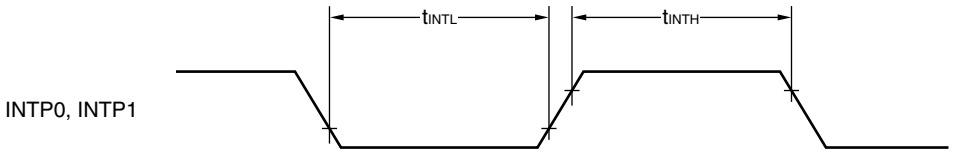
Clock Timing



TI000 Timing



Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing

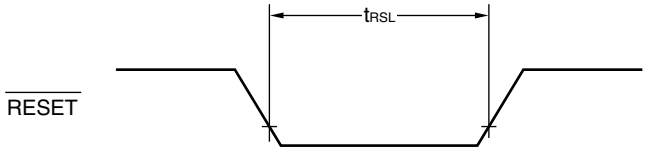


Figure B-2. When using the 78K0S/Kx1+ target cable (single track)