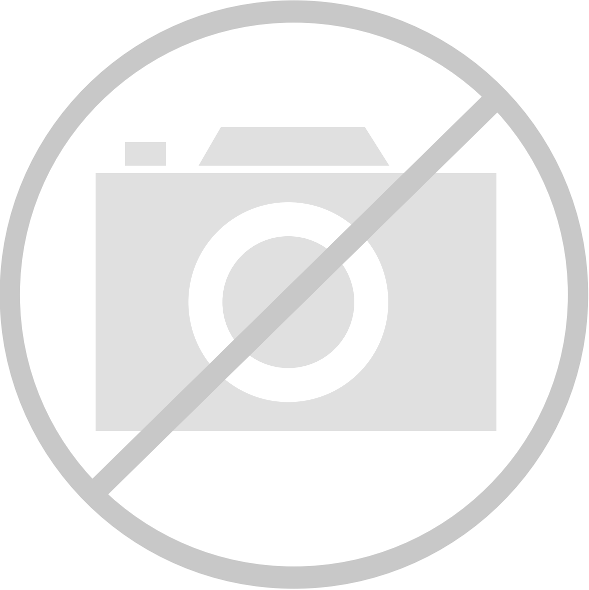
Renesas Electronics America Inc - UPD70F3424GJ(A)-GAE-AX Datasheet



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	Active
Core Processor	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3424gj-a-gae-ax

2.4.15 Port group 11

Port group 11 is an 8-bit port group. In alternative mode, it comprises pins for the following functions:

- Stepper Motor Controller/Driver outputs (SM11 to SM14, SM21 to SM24)
- Timer TMG2 channels (TOG21 to TOG24)
- Sound Generator outputs (SGO, SGOA)

Port group 11 includes the following pins:

Table 2-47 Port group 11: pin functions and port types

Pin functions in different modes				Pin function after reset	Port type
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
	Output mode (PMnm = 0)		Input mode (PMnm = 1)		
	PFCnm = 0 ALT1-OUT	PFCnm = 1 ALT2-OUT			
P110 (I/O)	SM11	TOG21	–	P110 (I)	M
P111 (I/O)	SM12	TOG22	–	P111 (I)	M
P112 (I/O)	SM13	TOG23	–	P112 (I)	M
P113 (I/O)	SM14	TOG24	–	P113 (I)	M
P114 (I/O)	SM21	SGO	–	P114 (I)	M
P115 (I/O)	SM22	SGOA	–	P115 (I)	M
P116 (I/O)	SM23		–	P116 (I)	M
P117 (I/O)	SM24		–	P117 (I)	M

Note 1. For pins that support only one alternative output mode, the PFCnm bit is not available.

2. Alternative *input* functions of TMG2 are provided on two pins each. Thus you can select on which pin the alternative function should appear. Refer to *“Alternative input selection“ on page 46*.

3. Port group 11 is equipped with high driver buffers for stepper motor control.

2.6 Pin Functions in Reset and Power Save Modes

The following table summarizes the status of the pins during reset and power save modes and after release of these operating states in normal operation mode, i.e. = 0.

The reset source makes a difference concerning the N-Wire debugger interface pins $\overline{\text{DRST}}$, DDI, DDO, DCK and DMS after reset release. An external $\overline{\text{RESET}}$ or an internal Power-on-clear switches all pins to input port mode, while all other internal reset sources make the pins available for the debugger.

In contrast to all other power save modes the HALT mode suspends only the CPU operation and has no effect on any pin status.

Table 2-61 Pin functions during and after reset / power save modes

Operating status		Pin status
Power-On-Clear	during	• P05/$\overline{\text{DRST}}$: P05 port input with internal pull-down resistor • all other pins: Hi-Z (3-state)
	after	• all ports Pnm: input port mode • A[23:0], D[15:0], $\overline{\text{BE}}$[1:0], $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$: input
external $\overline{\text{RESET}}$	during	• P05/$\overline{\text{DRST}}$: P05 port input with internal pull-down resistor • all other pins: Hi-Z (3-state)
	after	• P05/$\overline{\text{DRST}}$: $\overline{\text{DRST}}$ input with internal pull-down resistor • P52/DDI, P54/DCK, P55/DMS: DDI, DCK, DMS inputs • P53/DDO: DDO output • all ports Pnm: input port mode • A[23:0], D[15:0], $\overline{\text{BE}}$[1:0], $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$: input
all other reset sources	during	• P05/$\overline{\text{DRST}}$: P05 port input with internal pull-down resistor • all other pins: Hi-Z (3-state)
	after	• P05/$\overline{\text{DRST}}$, P52/DDI, P54/DCK, P55/DMS, P53/DDO: no change. same function as before reset • all ports Pnm: input port mode • A[23:0], D[15:0], $\overline{\text{BE}}$[1:0], $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$: input
HALT mode	during	same as before HALT mode
	after	
IDLE, WATCH, Sub-WATCH, STOP mode	during	same as before power save mode: • Output signals are valid and output levels are remained. • Input signals with wake-up capability[a] are valid. • Input signals without wake-up capability are ignored.
	after	same as before power save mode

[a] Inputs with wake-up capability: external interrupts (INTP0 to INTP7, NMI) and CAN receive data (CRXD0, CRXD1)

Note For information about the status of the external memory I/F pins refer to *Chapter 7 on page 255*.

If flash programming mode is enabled by FLMD0 = 1, P07 is used as FLMD1 pin in input port mode during and after reset.

(4) SCPS - SSCG post scaler control register

The 8-bit SCPS register controls the two independent SSCG post scalers (frequency dividers) for the CPU system clock VBCLKand for the modulated peripheral clocks SPCLK.

Access This register can be read/written in 8-bit or 1-bit units.

Address FFFF F830$_H$.

Initial Value 21$_H$. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	SPSPS2	SPSPS1	SPSPS0	0	VBSPS2	VBSPS1	VBSPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 4-13 SCPS register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6 to 4</td><td>SPSPS[2:0]</td><td>SSCG clock divider selection for generating SPCLK0:
<table>
<tr><th>SPSPS2</th><th>SPSPS1</th><th>SPSPS0</th><th>Clock divider setting</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>SPCLK0 = SSCG out frequency / 1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>SPCLK0 = SSCG out frequency / 2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>SPCLK0 = SSCG out frequency / 3</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>SPCLK0 = SSCG out frequency / 4</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>not supported</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>SPCLK0 = SSCG out frequency / 6</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>not supported</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>SPCLK0 = SSCG out frequency / 8</td></tr>
</table></td></tr>
<tr><td>2 to 0</td><td>VBSPS[2:0]</td><td>SSCG clock divider selection for generating VBCLK:
<table>
<tr><th>VBSPS2</th><th>VBSPS1</th><th>VBSPS0</th><th>Clock divider setting</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>VBCLK = SSCG out frequency / 1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>VBCLK = SSCG out frequency / 2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>VBCLK = SSCG out frequency / 3</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>VBCLK = SSCG out frequency / 4</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>not supported</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>VBCLK = SSCG out frequency / 6</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>not supported</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>VBCLK = SSCG out frequency / 8</td></tr>
</table></td></tr>
</table>

Note This register can only be written when the SSCG enable bit CKC.SCEN is cleared (SSCG switched off).

(5) ICC - IIC clock control register

The 8-bit ICC register determines the I^{2}C clock source and the clock divider setting for IICLK.

Access This register can be read/written in 8-bit units.

Writing to this register is protected by a special sequence of instructions. Please refer to *“PHCMD - Command protection register“ on page 141* for details.

Address FFFF F838$_H$.

Initial Value 00$_H$. The register is cleared by any reset.

7	6	5	4	3	2	1	0
0	IICPS2	IICPS1	IICPS0	0	0	IICSEL1	IICSEL0
R[a]	R/W	R/W	R/W	R[a]	R[a]	R/W	R/W

[a] These bits may be written, but write is ignored.

Table 4-18 ICC register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6 to 4</td><td>IICPS[2:0]</td><td>Divider setting for IICLK:
<table>
<tr><th>IICPS2</th><th>IICPS1</th><th>IICPS0</th><th>Clock divider setting</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1 / 3.5</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1 / 4.5</td></tr>
<tr><td colspan="3">other settings</td><td>not supported</td></tr>
</table></td></tr>
<tr><td>1 to 0</td><td>IICSEL[1:0]</td><td>Clock source for IICLK:
<table>
<tr><th>IICSEL1</th><th>IICSEL0</th><th>Clock source</th></tr>
<tr><td>0</td><td>0</td><td>Main oscillator</td></tr>
<tr><td>0</td><td>1</td><td>SSCG / 2</td></tr>
<tr><td>1</td><td>x</td><td>PLL</td></tr>
</table></td></tr>
</table>

Note

1. On release of WATCH, Sub-WATCH and STOP mode or when the SDC.SDCR bit is set, IICSEL[1:0] is cleared—the main oscillator is selected as the IIC clock source.
   Pay attention if PSM.OSCDIS = 1 before entering any of the above power save modes, because the main oscillator will be disabled. Therefore the I^{2}C interface will have no clock supply after power save mode release.

2. The connected I^{2}C interfaces must be disabled before switching IICPS[2:0]. To switch the IICPS bits, first disable the I^{2}C interface by clearing the enable bit in the IIC control register, then switch IICPS[2:0] and finally re-enable the IIC interface.

4.2.5 Clock monitor registers

The following registers are used to control the monitor circuits of the main oscillator clock and the sub oscillator clock.

Please refer to *“Operation of the Clock Monitors“ on page 191* for supplementary information.

(1) CLMM - Main oscillator clock monitor mode register

The 8-bit CLMM register is used to enable the monitor for the main oscillator clock.

Access This register can be read/written in 8-bit or 1-bit units.

Writing to this register is protected by a special sequence of instructions. Please refer to *“PRCMDCMM - CLMM write protection register“ on page 167* for details.

Address FFFF F870$_H$.

Initial Value 00$_H$. This register is cleared by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CLMEM
R	R	R	R	R	R	R	R/W

Table 4-22 CLMM register contents

Bit position	Bit name	Function
0	CLMEM	Clock monitor enable: 0: Clock monitor for main oscillator disabled. 1: Clock monitor for main oscillator enabled. This bit can only be cleared by reset.

Note CLMM.CLMEM can be set at any time. However, the clock monitor is only activated after the main oscillator has stabilized, indicated by CGSTAT.OSCSTAT = 1.

Chapter 5 Interrupt Controller (INTC)

This controller is provided with a dedicated Interrupt Controller (INTC) for interrupt servicing and can process a large amount of maskable and two non-maskable interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution. Generally, an exception takes precedence over an interrupt.

This controller can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

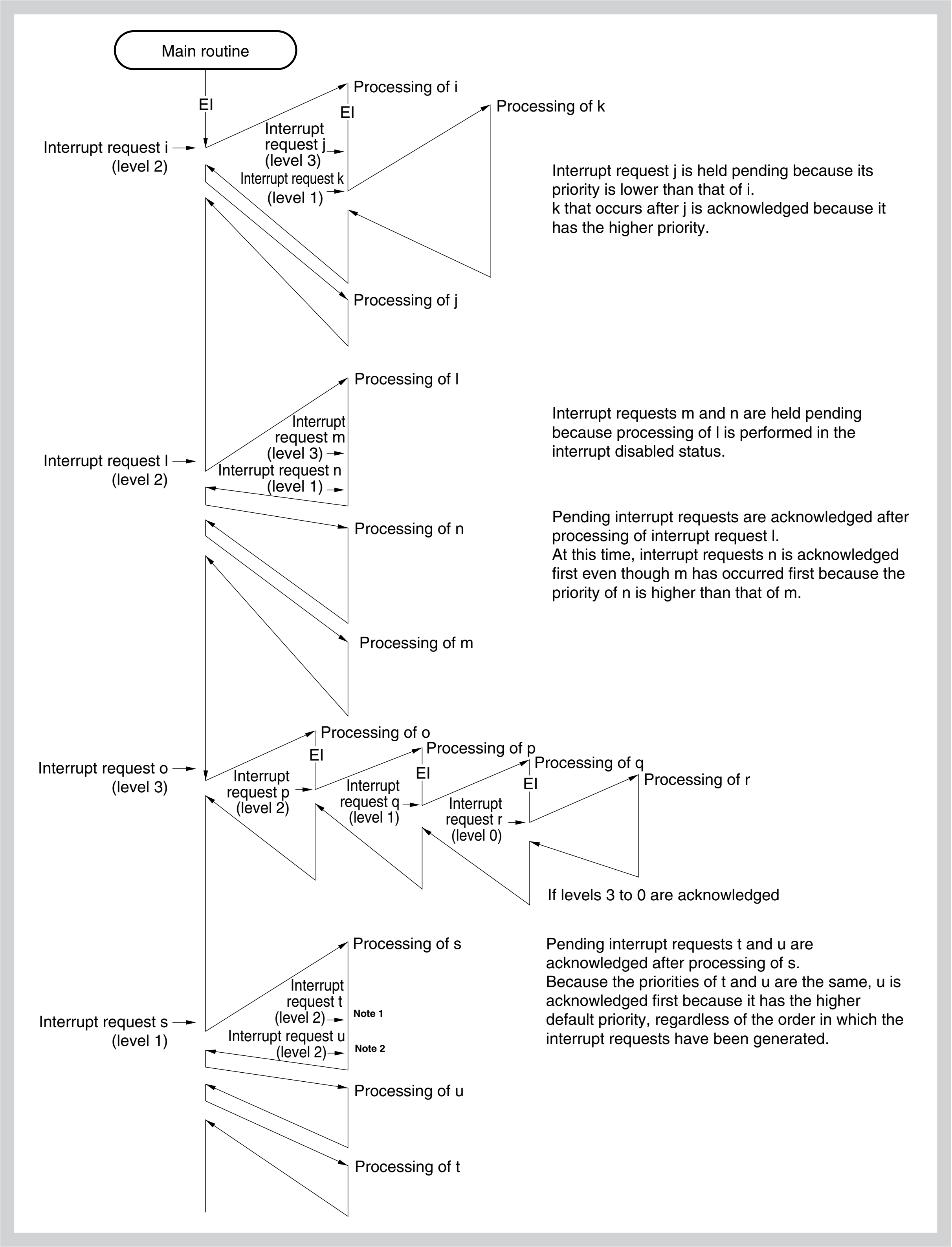
Eight levels of software-programmable priorities can be specified for each interrupt request. Starting of interrupt servicing takes no fewer than 5 system clocks after the generation of an interrupt request.

5.1 Features

- Interrupts
  - Non-maskable interrupts: 2 sources
  - Maskable interrupts:

Interrupt source	µPD70F3421, µPD70F3422, µPD70F3423	µPD70F3424, µPD70F3425, µPD70F3426A, µPD70F3427
Internal peripherals	69	85
External	7	8
Software	2	2

  - 8 levels of programmable priorities (maskable interrupts)
  - Multiple interrupt control according to priority
  - Masks can be specified for each maskable interrupt request
  - Noise elimination, edge detection and valid edge specification, level detection for external interrupt request signals
  - Wake-up capable
    (analogue noise elimination for external interrupt request signals)
  - NMI and INTP0 share the same pin
- Exceptions
  - Software exceptions: 2 channels with each 16 sources
  - Exception traps: 2 sources (illegal opcode exception and debug trap)



Interrupt request j is held pending because its priority is lower than that of i.
k that occurs after j is acknowledged because it has the higher priority.

Interrupt requests m and n are held pending because processing of l is performed in the interrupt disabled status.

Pending interrupt requests are acknowledged after processing of interrupt request l.
At this time, interrupt requests n is acknowledged first even though m has occurred first because the priority of n is higher than that of m.

Pending interrupt requests t and u are acknowledged after processing of s.
Because the priorities of t and u are the same, u is acknowledged first because it has the higher default priority, regardless of the order in which the interrupt requests have been generated.

Figure 5-8 Example of processing in which another interrupt request is issued while an interrupt is being processed (2/2)

Note To initialize an external memory area after a reset, register PRC has to be set if page ROM mode is selected. Do not change this register after initialization. Do not access external page ROM devices before initialization is finished.

Caution To initialize an external memory area after a reset, this register has to be set. Do not access external devices before initialization is finished. Do not change this register while an external device is accessed.

Caution Special care must be taken when using the internal RAM as DMA source or destination. Refer to *“Simultaneous program execution and DMA transfer with internal RAM“ on page 342*.

- DMA transfer completion flag
- Automatic restart function
- Forcible DMA termination by NMI

Caution These registers cannot be accessed during DMA operation.

(6) CORVALnH - VFB flash/ROM “Data Replacement” ROM correction value register

These registers hold the upper 16 bit of the value that shall replace the original value from the VFB flash/ROM.

Access These registers can be read/written in 16-bit units.

Address CORVAL0H: FFFF F932$_H$
CORVAL1H: FFFF F936$_H$
CORVAL2H: FFFF F93A$_H$
CORVAL3H: FFFF F93E$_H$
CORVAL4H: FFFF F942$_H$
CORVAL5H: FFFF F946$_H$

Initial Value 0000$_H$

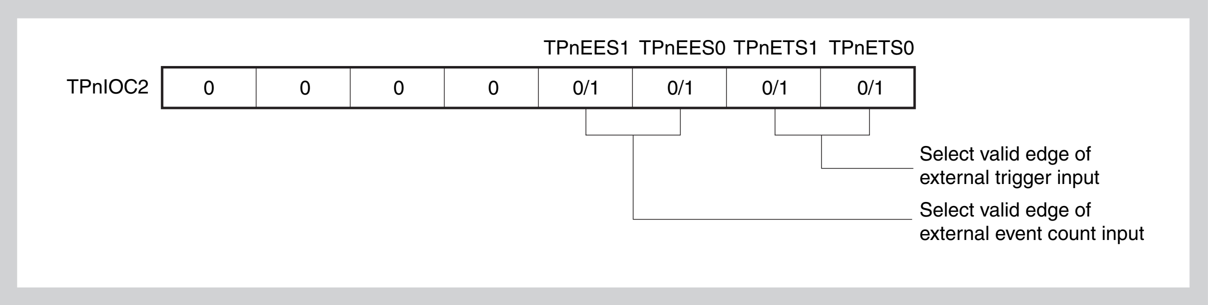
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORVALn[31:16]															
R/W															

Table 9-7 CORVALnH register contents

Bit Position	Bit Name	Function
15 to 0	CORVALn [31:16]	Upper 16 bit of the correction value to replace the ROM contents.

Note CORVALnH shall only be changed when the corresponding channel is disabled (CORCTL0.CORCENn = 0).

(d) TMPn I/O control register 2 (TPnIOC2)



					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1	0/1	0/1	0/1

Select valid edge of external trigger input

Select valid edge of external event count input

(e) TMPn counter read buffer register (TPnCNT)

The value of the 16-bit counter can be read by reading the TPnCNT register.

(f) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

If D_0 is set to the TPnCCR0 register and D_1 to the TPnCCR1 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1)$ × Count clock cycle

Active level width = D_1 × Count clock cycle

Note TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the external trigger pulse output mode.

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn bit = 0 (after restart, address match)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1		▲2					▲3	▲4		Δ5

▲1: IICSn register = 0010X010B

▲2: IICSn register = 0010X000B

▲3: IICSn register = 0001X110B

▲4: IICSn register = 0001X000B

Δ 5: IICSn register = 00000001B

Remarks 1. ▲: Always generated
Δ: Generated only when SPIEn bit = 1
X: don’t care
2. n = 0 to 2

<2> When WTIMn bit = 1 (after restart, address match)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP	
			▲1	▲2		▲3				▲4		▲5	Δ6

▲1: IICSn register = 0010X010B

▲2: IICSn register = 0010X110B

▲3: IICSn register = 0010XX00B

▲4: IICSn register = 0001X110B

▲5: IICSn register = 0001XX00B

Δ 6: IICSn register = 00000001B

Remarks 1. ▲: Always generated
Δ: Generated only when SPIEn bit = 1
X: don’t care
2. n = 0 to 2

(3) CnGMABT - CANn global automatic block transmission control register

The CnGMABT register is used to control the automatic block transmission (ABT) operation.

Access This register can be read/written in 16-bit units.

Address <CnRBaseAddr> + 006_H

Initial Value 0000_H. The register is initialized by any reset.

(a) CnGMABT read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	ABTCLR	ABTTRG

ABTCLR	Automatic block transmission engine clear status bit
0	Clearing the automatic transmission engine is completed.
1	The automatic transmission engine is being cleared.

Note
1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared to 0.
   The operation is not guaranteed if the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1.
2. When the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared to 0 as soon as the requested clearing processing is complete.

ABTTRG	Automatic block transmission status bit
0	Automatic block transmission is stopped.
1	Automatic block transmission is under execution.

Caution
1. Do not set the ABTTRG bit (1) in the initialization mode. If the ABTTRG bit is set in the initialization mode, the operation is not guaranteed after the CAN module has entered the normal operation mode with ABT.
2. Do not set the ABTTRG bit (1) while the CnCTRL.TSTAT bit is set (1). Confirm TSTAT = 0 directly in advance before setting ABTTRG bit.

CCERC	Error counter clear bit
0	The CnERC and CnINFO registers are not cleared in the initialization mode.
1	The CnERC and CnINFO registers are cleared in the initialization mode.

Note
1. The CCERC bit is used to clear the CnERC and CnINFO registers for re-initialization or forced recovery from the bus-off state. This bit can be set to 1 only in the initialization mode.
2. When the CnERC and CnINFO registers have been cleared, the CCERC bit is also cleared to 0 automatically.
3. The CCERC bit can be set to 1 at the same time as a request to change the initialization mode to an operation mode is made.
4. The CCERC bit is read-only in the CAN sleep mode or CAN stop mode.
5. The receive data may be corrupted in case of setting the CCERC bit to (1) immediately after entering the INIT mode from self-test mode.

AL	Bit to set operation in case of arbitration loss
0	Re-transmission is not executed in case of an arbitration loss in the single-shot mode.
1	Re-transmission is executed in case of an arbitration loss in the single-shot mode.

Note The AL bit is valid only in the single-shot mode.

VALID	Valid receive message frame detection bit
0	A valid message frame has not been received since the VALID bit was last cleared to 0.
1	A valid message frame has been received since the VALID bit was last cleared to 0.

Note
1. Detection of a valid receive message frame is not dependent upon storage in the receive message buffer (data frame) or transmit message buffer (remote frame).
2. Clear the VALID bit (0) before changing the initialization mode to an operation mode.
3. If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in the normal mode and the other in the receive-only mode, the VALID bit is not set to 1 before the transmitting node enters the error passive state, because in receive-only mode no acknowledge is generated.
4. To clear the VALID bit, set the Clear VALID bit to 1 first and confirm that the VALID bit is cleared. If it is not cleared, perform clearing processing again.

Set RDY	Clear RDY	Setting of RDY bit
0	1	RDY bit is cleared to 0.
1	0	RDY bit is set to 1.
Other than above		RDY bit is not changed.

Set IE bit and RDY bit always separately.

Caution

1. Do not set the DN bit to 1 by software. Be sure to write 0 to bit 10.
2. Do not set the TRQ bit and the RDY bit (1) at the same time. Set the RDY bit (1) before setting the TRQ bit.
3. Do not clear the RDY bit (0) during message transmission. Follow the transmission abort process about clearing the RDY bit (0) for redefinition of the message buffer.
4. Clear again when RDY bit is not cleared even if this bit is cleared.
5. Be sure that RDY is cleared before writing to the other message buffer registers, by checking the status of the RDY bit.

20.8.5 Resetting error counter CnERC of CAN module

If it is necessary to reset the CAN module error counter CnERC and CAN module information register CnINFO when re-initialization or forced recovery from the bus-off status is made, set the CCERC bit of the CnCTRL register to 1 in the initialization mode. When this bit is set to 1, the CnERC and CnINFO registers are cleared to their default values.

24.4 Cautions

24.4.1 Polling of LBCTL0.TPF0 flag may indicate wrong status

Though the LBCTL0.TPF0 flag is intented to determine the current status of the LCD bus data transfer, reading of this flag may indicate a wrong status by accident.

Therefore, instead of polling the LBCTL0.TPF0 flag it is recommended to use an interrupt procedure (INTLCD) or a DMA transfer to load new LCD data into the LCD bus interface data register (LBDATA0x).

Polling of the IF flag of the corresponding interrupt control register (LCDIC) is not affected and can be applied alternatively.

24.4.2 Writing to the LBDATA0W/ LBDATA0/ LBDATA0L register

When writing to the LBDATA0x register while a transfer on the LCD data bus is ongoing a corrupt data transfer may be the result. The critical situation can occur under certain clock constellations.

To avoid the critical situation one of the following measures must be applied.

(1) Avoidance of simultaneous write to LBDATA0x register and LCD data bus transfer

To ensure that LBDATA0x register is not written while a transfer is ongoing, the LBDATA0x register should be operated upon the occurrence of the LCD Bus Interface interrupt (INTLCD) with LBCTL0.TCIS0 set to 1.

(2) Avoidance of crictical clock constellations

Use one of the following clock settings depending on the product. For other combinations a critical clock constellation cannot be excluded, particularly if the CPU is supplied by SSCG and the LCD bus interface by PLL. When a combination other then below is used the measure (1) above has to be applied.

(a) CPU system clock (VBCLK) and LCD bus clock are supplied by PLL

Conditions for all products

PLLCLK = 32 MHz

SCC.SPSEL[1:0] = 01B
(SPCLK0 = PLLCLK/2, SPCLK1 = PLLCLK/4)

μPD70F3421
μPD70F3422
μPD70F3423
μPD70F3424

PCC.CKS[1:0] = 11B
(CPU system clock (VBCLK) = PLLCLK= 32 MHz)

LBCTL0.LBC0[1:0] = 00B
(LCD bus clock = SPCLK0 = 16 MHz)

27.1.1 General reset performance

The following figure shows the signals involved in the reset function:

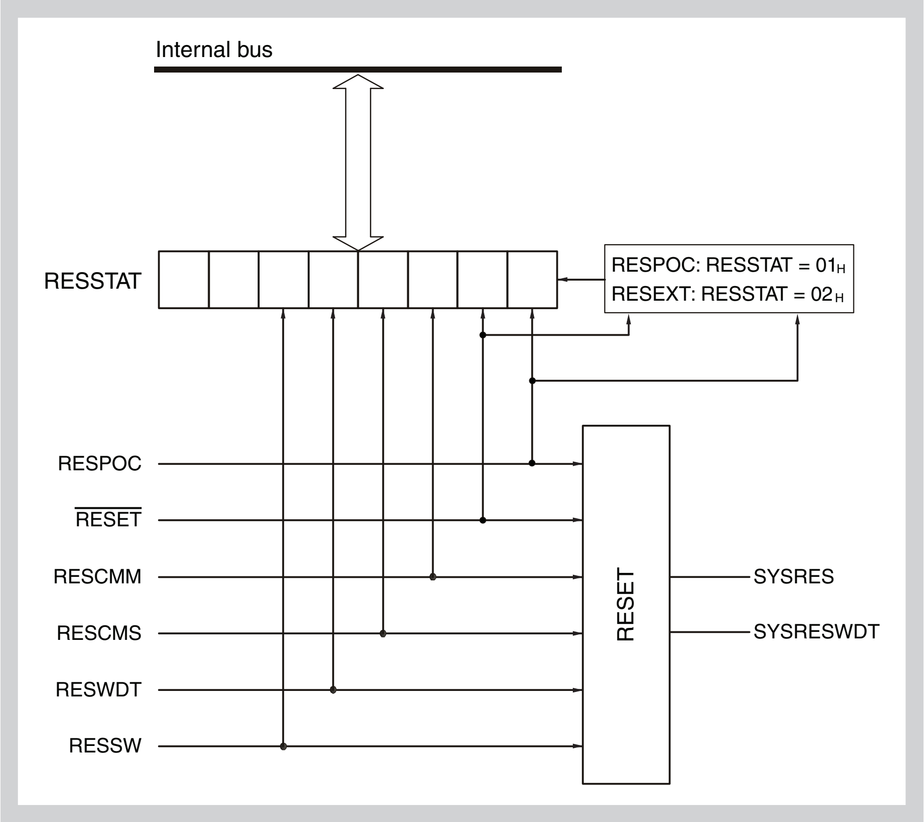


Figure 27-1 Reset function signal diagram

All resets are applied asynchronously. That means, resets are not synchronized to any internal clock. This ensures that the microcontroller can be kept in reset state even if all internal clocks fail to operate.

The reset function provides two internal reset signals:

- System reset SYSRES
  SYSRES is activated by all reset sources.
- Watchdog reset SYSRESWDT
  SYSRESWDT is activated by Power-On-Clear and external $\overline{\text{RESET}}$ only.

Both resets provoke different reset behaviour of the Watchdog Timer. For details refer to the *“Watchdog Timer (WDT)“ on page 527*.

(1) Variable reset vector

The flash memory devices allow to program the start address of the user’s program, instead of starting at address 0000 0000_H. The variable reset vector is stored in the extra area of the flash memory and can be written by an external flash programmer or in self-programming mode.

(8) Debug monitor function

A memory space for debugging that is different from the user memory space is used during debugging (background monitor mode). The user program can be executed starting from any address.

While execution of the user program is aborted, the user resources (such as memory and I/O) can be read and written, and the user program can be downloaded.

(9) Mask function

Each of the following signals can be masked. That means these signals will not be effective during debugging.

The correspondence with the mask functions of the debugger (ID850NWC) for the N-Wire emulator (IE-V850E1-CD-NW) is shown below.

– NMI0 mask function:	NMI pin
– NMIWDT mask function:	Watchdog Timer interrupt NMIWDT
– Reset mask function:	all reset sources

(10) Timer function

The execution time of the user program can be measured.

(11) Peripheral macro operation/stop selection function during break

Depending on the debugger to be used, certain peripheral macros can be configured to continue or to stop operation upon a breakpoint hit.

- Functions that are always stopped during break
  - Watchdog Timer
- Functions that can operate or be stopped during break (however, each function cannot be selected individually)
  - A/D Converter
  - All timers P
  - All timers Z
  - Timer Y
  - Watch Timer
- Peripheral functions that continue operating during break (functions that cannot be stopped)
  - Peripheral functions other than above

(12) Function during power saving modes

When the device is set into a power saving mode, debug operation is not possible. When exiting the power save mode, the on-chip debug unit continues operation.

The N-Wire interface is still accessible during power saving modes:

- N-Wire emulator can get status information from the on-chip debug unit.
- Stop mode can be released by the N-Wire emulator.