Application Note

16

RL78 Window Watchdog Timer

16-bit Single-Chip Microcontroller

RL78/G12
RL78/G13
RL78/F12
RL78/D1A

Renesas Electronics
www.renesas.com

R01AN0603ED0100
April 5, 2011

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   – The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at power-on

   The state of the product is undefined at the moment when power is supplied.

   – The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

     In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

     In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited.

   – The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   – When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between products

   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

   – The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Table of Contents

Chapter 1 Reason for a Window Watchdog

1.1 Reason and benefit of a watchdog in general

In an application, after reset, a microcontroller has to control the functionality of the system. Usually the system works properly up to the time, the device is switched off.

In rare cases, it might happen that the microcontroller is running out of control. Several reasons can be responsible for that phenomena and the effect of being out of control to the application can show a lot of different undesired behaviours.

One example for malfunction is a software bug, which guides the software into an endless loop. In this case, the microcontroller shows no defined reaction or action at all for ever. So, the application/system is out of order. The user has to initiate a reset (e.g. by switching off and on the power) to reactivate the application.

A watchdog is an internal or external peripheral, which has to be restarted by the software sequential within a pre-defined timing range. If the software run fails (e.g. by stucking in an endless loop) and the watchdog is not restarted within the expected time, the watchdog generates a reset. Using such kind of security - peripheral, it is given that an application does not stuck for all time and the application restarts and recovers. This does not prohibit the malfunction, but it recovers the application.

1.2 Advantage of a window watchdog compared to a usual watchdog

An usual watchdog has only one (selectable) time period within the watchdog has to be restarted. The restart can be done at any time up to short before the selected watchdog time has elapsed. There is only one restriction for proper watchdog - timer handling: restart has to be done before watchdog - timer overflows.

If the microcontroller stucks in an endless loop in which restarting of the watchdog - timer is implemented, the microcontroller will also stuck for ever, as if no watchdog is implemented at all.

The window watchdog timer (WDT) has two time stamps within the restart is allowed: A dedicated selectable time after WDT start and the overflow time, the so called "open window". When a restart is triggered during this time frame, WDT restart is done. If the restart is triggered outside this window, a reset of the microcontroller is initiated.

The chance that the microcontroller will stuck just within an endless loop where a restart of the watchdog - timer is implemented might be very seldom. Stucking in an endless loop where a restart of the watchdog - timer is implemented just within the watchdog window open time frame might be nearly impossible.

So, the window watchdog increases the security level of the system, compared to a simple watchdog.

Chapter 2 Functionality of a Window Watchdog

2.1 Functions of Watchdog Timer in RL78 controllers

The watchdog timer (WDT) is clocked by the internal low-speed oscillation clock.

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases:

1. If the watchdog timer counter overflows
2. If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
3. If data other than “ACH” is written to the WDTE register
4. If data is written to the WDTE register during a window close period

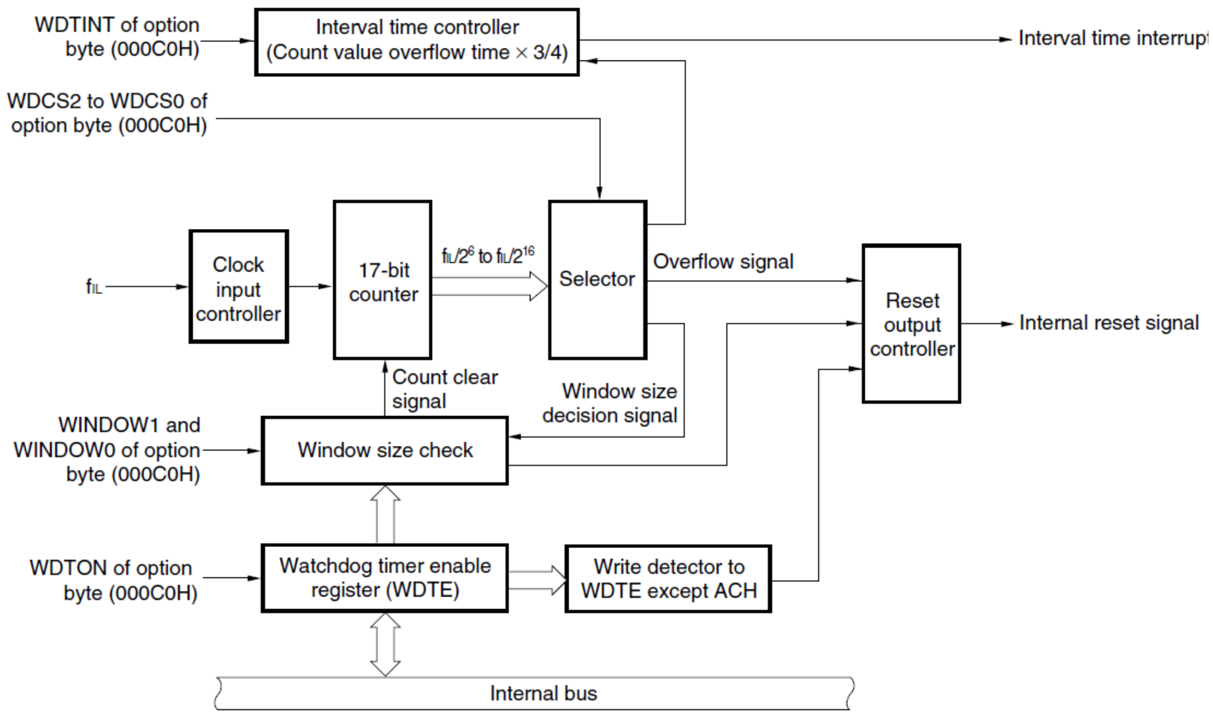
When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. So, after reset release, the software can recognize that the watchdog has triggered the reset before.

When 75% of the overflow time is reached, an interval interrupt can be generated.

2.2 Description of the watchdog timer in RL78

The block diagram of the watchdog timer in RL78 controllers is shown in *Figure 2-1*

Figure 2-1 Block diagram of RL78 watchdog timer


WDTINT of option byte (000C0H)
Interval time controller (Count value overflow time × 3/4)
Interval time interrupt
WDCS2 to WDCS0 of option byte (000C0H)
fIL
Clock input controller
17-bit counter
fIL/2^6 to fIL/2^16
Selector
Overflow signal
Reset output controller
Internal reset signal
Count clear signal
Window size decision signal
WINDOW1 and WINDOW0 of option byte (000C0H)
Window size check
WDTON of option byte (000C0H)
Watchdog timer enable register (WDTE)
Write detector to WDTE except ACH
Internal bus

Operation can be enabled by setting of bit WDTON = 1 in the user option byte 000C0H/010C0H. The window watchdog timer is clocked by the internal low-speed oscillator. The overflow time and open window time can also be selected in the user option byte 000C0H/010C0H. When enabled, the window watchdog timer has to be restarted regularly by software within the chosen open window time; otherwise an internal reset is generated.

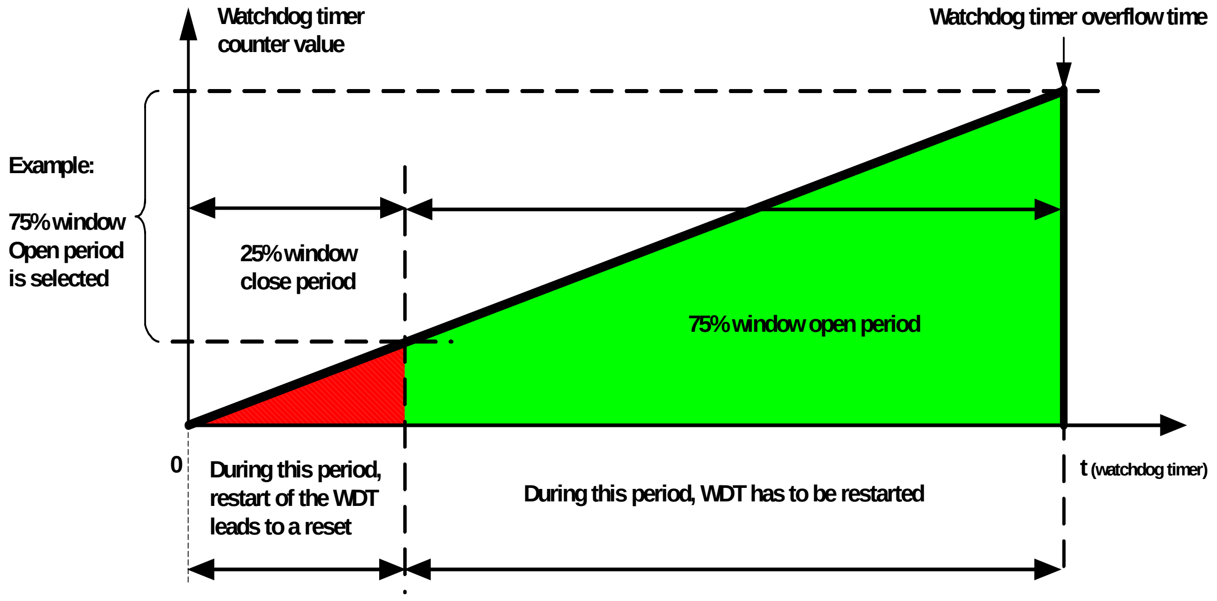
2.3 Window open / window close period

An usual watchdog timer has to be restarted during any time but before watchdog timer has elapsed to perform proper functionality of the application.

In opposite to a usual watchdog timer, the window watchdog timer has to be restarted during a specific time period, the so called window open period (see *Figure 2-2*). If there is no restart during the window open period, the watchdog overflows and generates an internal reset, same as a usual watchdog timer.

If there is an attempt to restart the window watchdog timer before window open period begins, there is also an internal reset generated. This feature is not implemented in an usual watchdog timer.

Figure 2-2 Window close period / window open period


Watchdog timer counter value
Watchdog timer overflow time
Example:
75% window Open period is selected
25% window close period
75% window open period
0
During this period, restart of the WDT leads to a reset
During this period, WDT has to be restarted
t (watchdog timer)

The open window period can be selected within three steps: 50%, 75% and 100% of the watchdog timer overflow time.

When 100% window open period is selected, the functionality is the same like a simple watchdog timer, due to restart access can be done within any time below watchdog timer overflow.

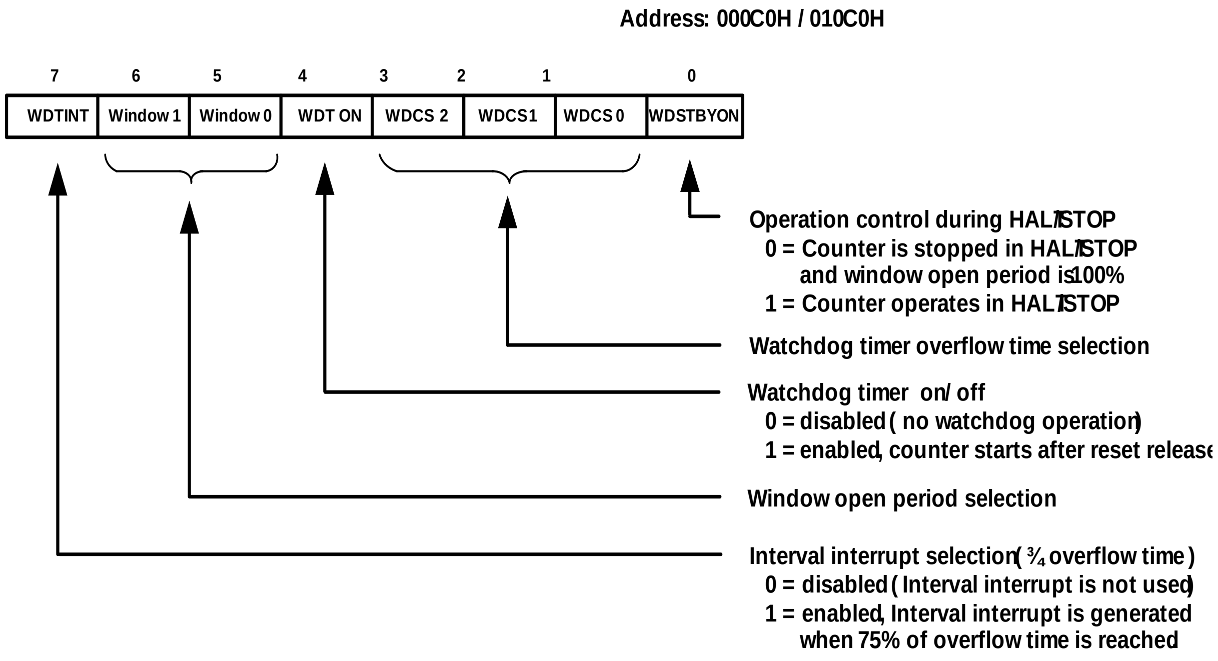
2.4 User Option Byte 000C0H/010C0H

In case of RL78, the fundamental setting of the watchdog timer (WDT) is done within setting of the option byte 000C0H/010C0H. The option byte selects if the watchdog timer is enabled or not. When it is enabled, the watch timer overflow time is valid, set with bits WDCS2 - WDCS0. Furthermore, the open window time-frame can be fixed in three levels. Selection can be done if an interrupt occurs when 75% of the watchdog time has elapsed.This interrupt might be useful for proper watchdog timer restart.

Due to the option byte is located within the code flash area, its setting can not be changed by software. This is an additional security item.

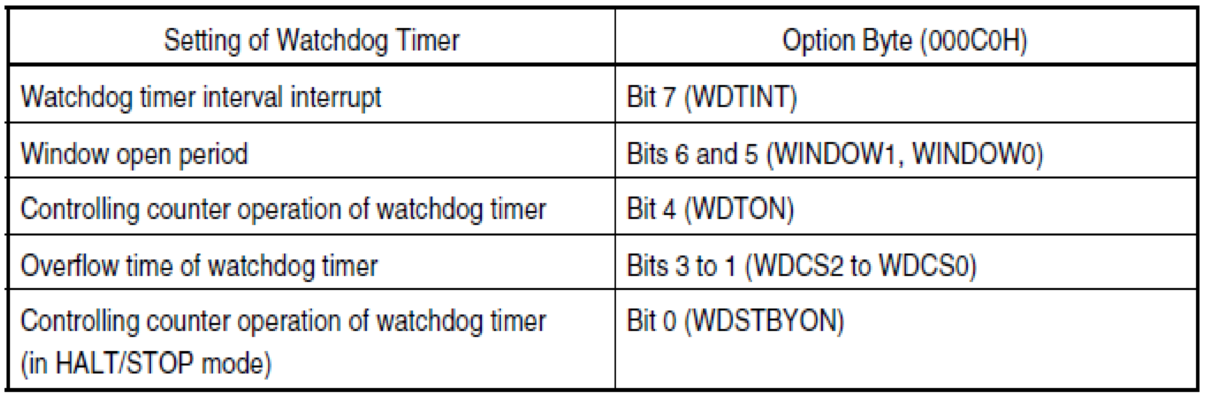
The content of the option byte is valid just after reset release. If the WDT is enabled, restart handling of the WDT has to be active just after reset release.

Figure 2-3 User Option byte 000C0H/010C0H for watchdog timer settings


Address: 000C0H / 010C0H
7
6
5
4
3
2
1
0
WDTINT
Window 1
Window 0
WDT ON
WDCS 2
WDCS1
WDCS 0
WDSTBYON
Operation control during HALT/STOP
0 = Counter is stopped in HALT/STOP and window open period is 100%
1 = Counter operates in HALT/STOP
Watchdog timer overflow time selection
Watchdog timer on/ off
0 = disabled (no watchdog operation)
1 = enabled, counter starts after reset release
Window open period selection
Interval interrupt selection (¾ overflow time)
0 = disabled (Interval interrupt is not used)
1 = enabled, Interval interrupt is generated when 75% of overflow time is reached

Caution Set the same value in 000C0H as in 010C0H when the boot swap operation is used because during boot swap operation 000C0H is replaced by 010C0H.

Table 2-1 Bits of Option Byte 000C0H/010C0H for Watchdog timer



Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

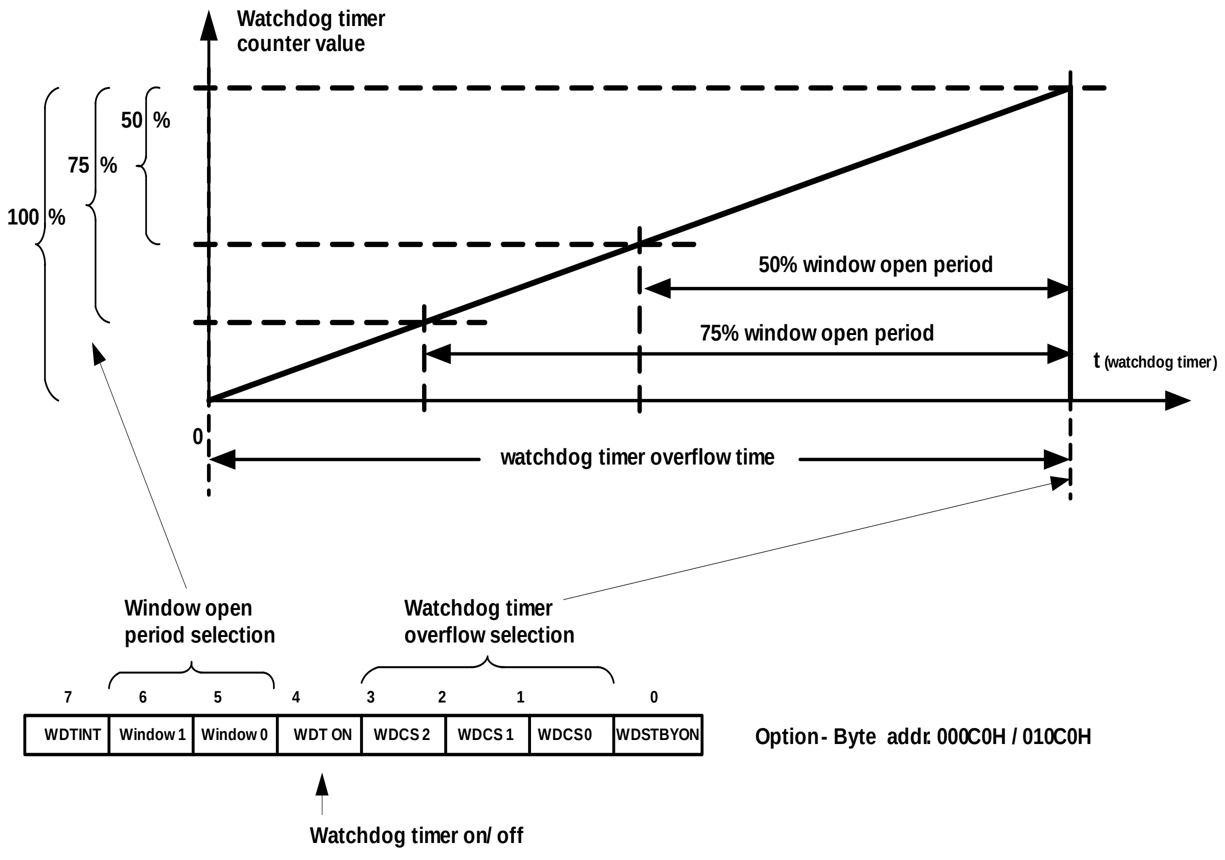
Notes

1. WDTINT = 1 generates an interrupt when 75% of the WDT time has elapsed. This might be useful for restarting the watchdog. When using this feature a separate supervision of the window close period by an additional timer is not necessary.
2. Window open period (Window1, Window0):
   This is the save period, the software has to restart the watchdog. Take care that the window open period is wide enough for usual software handling including WDT restart and possible interrupt handling, even when CPU is under highest load. If uncertain, select a wider window period.
3. WDTON: Watchdog timer enable / disable
   When WDT is enabled, be aware that the watchdog timer starts operation just after reset release. For proper functionality of the application, you have to take care to restart the WDT periodical.
4. WDCS2-0: Selection of watchdog timer overflow time
   Within this time, the watchdog has to be restarted. This period has to be defined seriously, it shouldn't be set too short. When WDT is enabled, the watchdog timer starts operation just after reset release. Take the main oscillators stabilization time into account, when the WDT is restarted when the CPU is clocked by external crystal.
   The watchdog timer continues its operation during self-programming of the flash memory and EEPROM^{TM} emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.
5. WDSTBYON controls if WDT is stopped during HALT/STOP or not.
   When WDTON = 1, WUTMMCK0 = 0 and WDSTBYON = 0, oscillation of the internal low-speed oscillator stops if the HALT or STOP instruction is executed. This reduces the power consumption during standby mode.
   When bit WDSTBYON = 0, the window open period is 100% regardless of of WINDOW1 and WINDOW0 bit setting.

   When WDTON = 1 and WDSTBYON = 1, oscillation of the internal low-speed oscillator and the watchdog timer keeps running during HALT- and STOP- mode. The CPU has to be waked up sequentially to restart the watchdog. This increases the standby current.

The correspondence between bit - setting in option byte and timing is described in *Figure 2-4*.

Figure 2-4 Timing selection of the window watchdog



When 100% window open period is selected, the functionality is the same as in usual watchdog behaviour. In this case, restart of the watchdog timer can be done at any time during watchdog timer runs. Restart has to be done before watchdog timer overflows.

Keep in mind: When watchdog counter is stopped during HALT/STOP - mode is selected (WDSTBYON = 0), the window open period is 100%, regardless of WINDOW1, WINDOW0 setting.

Chapter 3 Window Watchdog in the Application

3.1 Supervision for watchdog restart condition

When the watchdog timer starts to run, the watchdog timer must not be restarted during the window close time, otherwise a watchdog - reset is generated. Therefore, the window close period has to be supervised.

3.1.1 Supervision of watchdog timer using 75% interrupt

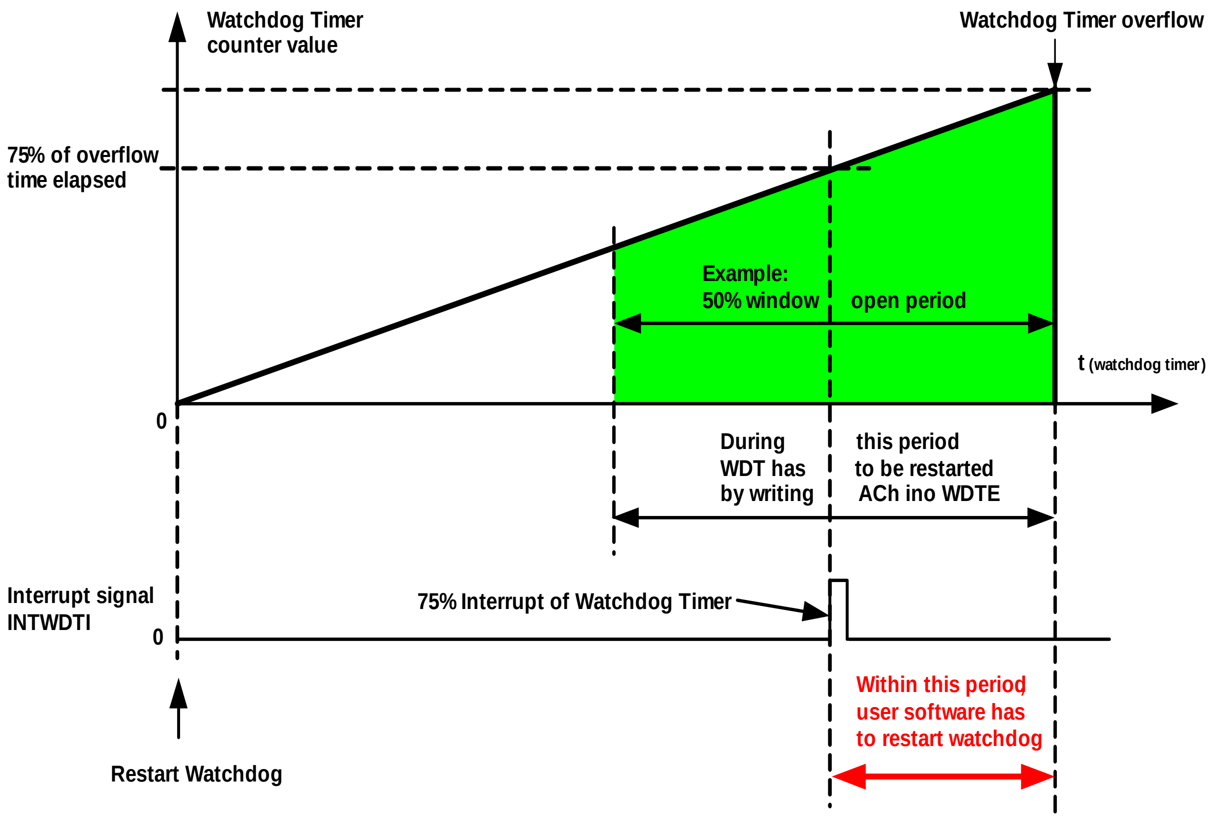
RL78 window watchdog timer is equipped with an 75% interrupt. When 3/4 of the watchdog timer period has elapsed, this interrupt is generated when enabled with option byte 000C0H/010C0H bit WDTINT = 1.

When this interrupt occurs, 1/4 of the watchdog timer period is left to restart the watchdog.

If this remaining time is sufficient for restart under all conditions, the usage of this interrupt might be the most convenient one to handle watchdog timer restart:

- There is no additional timer necessary
- There is no additional timer handling necessary
- The tolerance of the internal low-speed OSC is eliminated

Figure 3-1 75% interrupt, generated by the watchdog timer



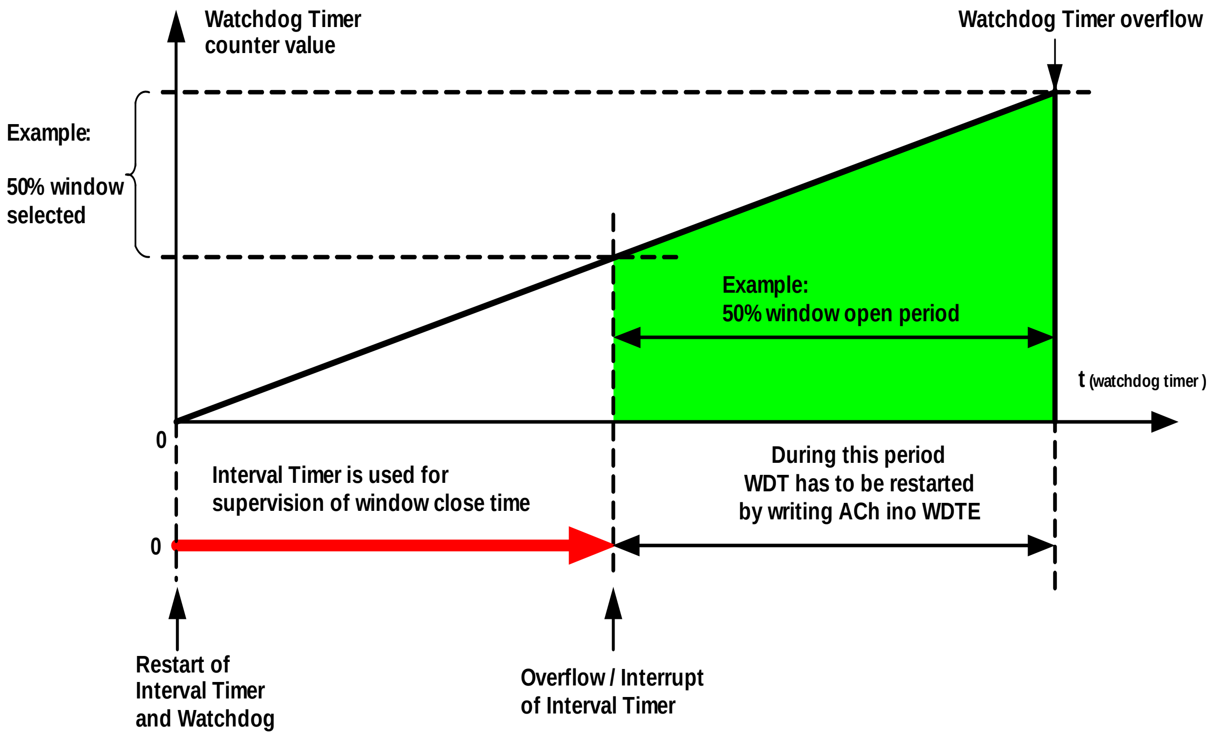
Watchdog Timer counter value
Watchdog Timer overflow
75% of overflow time elapsed
Example: 50% window
open period
t (watchdog timer)
0
During WDT has by writing
this period to be restarted ACh ino WDTE
Interrupt signal INTWDTI
0
75% Interrupt of Watchdog Timer
Within this period user software has to restart watchdog
Restart Watchdog

3.1.2 Supervision of window close time using a timer

Supervision of the window close time can be realized using a timer.

The window watchdog timer is clocked with the internal low-speed oscillator. This internal low-speed oscillator has a tolerance. To eliminate this tolerance, the timer which supervises the window watchdog timer should be clocked with the same clock in best case. So, the interval timer is the optimal candidate to be used for the supervision of the window - close time, due to the same clock source is used (see: *Figure 3-2*). Of course, when you take the tolerance of the internal low-speed oscillator into account, any other time base is also suitable for use.

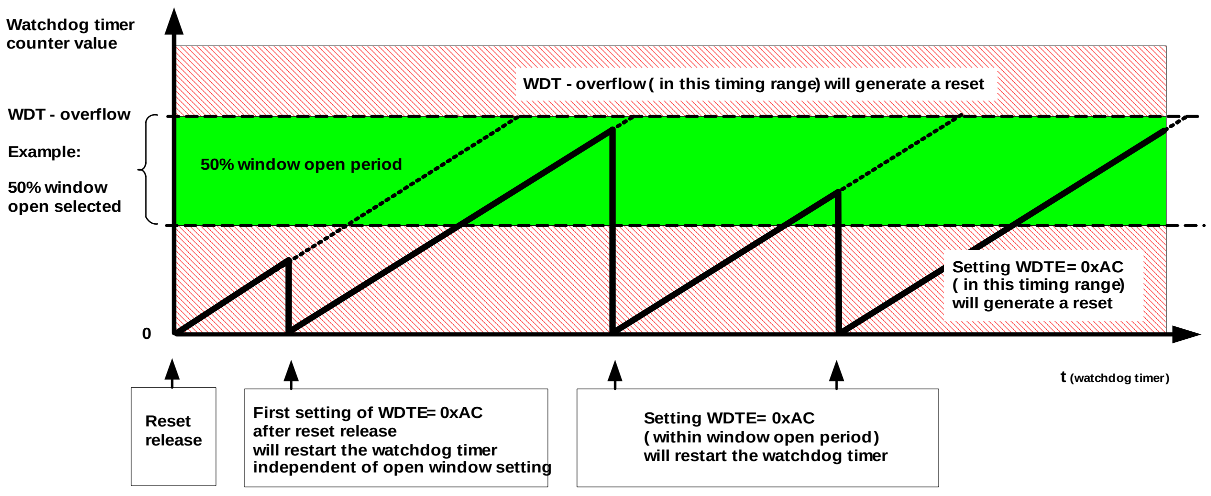
Figure 3-2 Interval Timer supervises the window close time


Watchdog Timer counter value
Watchdog Timer overflow
Example:
50% window selected
Example:
50% window open period
t (watchdog timer)
0
Interval Timer is used for supervision of window close time
During this period WDT has to be restarted by writing ACh ino WDTE
0
Restart of Interval Timer and Watchdog
Overflow / Interrupt of Interval Timer

3.2 WDT restart during window open period

To avoid that the window watchdog generates an undesired reset, it has to be restarted cyclic during window open time only. If the watchdog timer overflows, a reset is generated. If the software attempts to restart the WDT below the window open period, a reset is also generated. See *Figure 3-3* for proper restart handling of the WDT.

Figure 3-3 WDT restart during open window time range

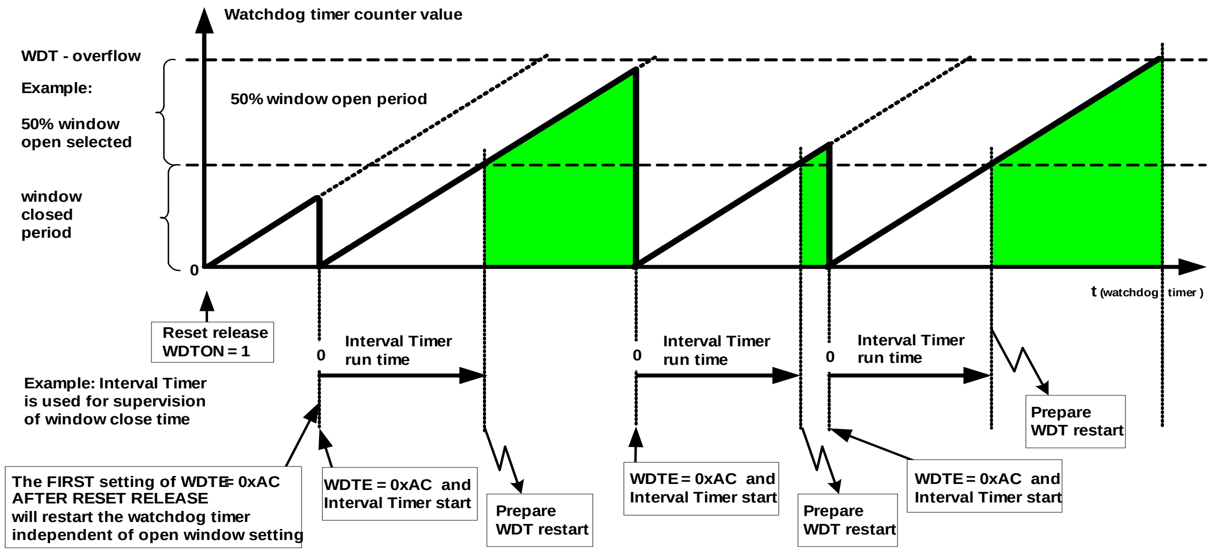

Watchdog timer counter value
WDT - overflow
Example:
50% window open selected
0
WDT - overflow (in this timing range) will generate a reset
50% window open period
Setting WDTE= 0xAC (in this timing range) will generate a reset
t (watchdog timer)
Reset release
First setting of WDTE= 0xAC after reset release will restart the watchdog timer independent of open window setting
Setting WDTE= 0xAC (within window open period) will restart the watchdog timer

Note There is one exception for restart - timing. Just after reset release, restarting the watchdog timer is independent from the selected window open time. So, after reset release the first restart access can be done within any time but before watchdog timer overflow.

3.3 Interval timer handling for window close period supervision

Figure 3-4 shows the proper restart handling of the window watchdog timer, using interval timer for supervision of the window close time.

Figure 3-4 Sequential restart of the WDT using interval timer


Watchdog timer counter value
WDT - overflow
Example:
50% window open selected
50% window open period
window closed period
0
t (watchdog timer)
Reset release WDTON = 1
Example: Interval Timer is used for supervision of window close time
0
Interval Timer run time
The FIRST setting of WDTE = 0xAC AFTER RESET RELEASE will restart the watchdog timer independent of open window setting
WDTE = 0xAC and Interval Timer start
Prepare WDT restart
WDTE = 0xAC and Interval Timer start
Prepare WDT restart
WDTE = 0xAC and Interval Timer start
Prepare WDT restart

Note The maximum interval time of the interval timer (max. 4095 clocks of f_{IL}) is shorter than the maximum interval time of the watchdog timer (max. 65536 clocks of f_{IL}).

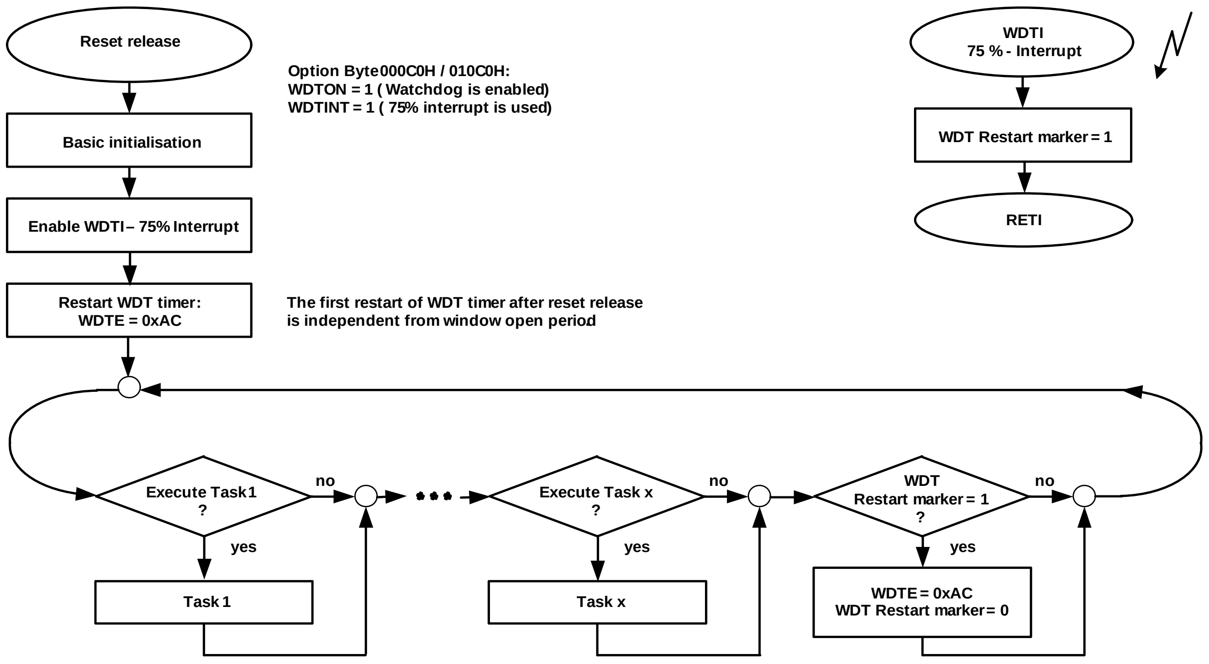
If the selected interval time of the watchdog timer is longer than the interval timer ones, several interval timer overflows may have to be counted before the restart instruction of the watchdog timer is executed.

Chapter 4 Handling of WDT restart

The software has to be prepared to restart the window watchdog timer just during the window open period.

Figure 4-1 shows an example of a simplified operating system, focused on the window watchdog timer handling for proper WDT restart using 75% watchdog timer interrupt.

Figure 4-1 Simplified operating system, handling of watchdog timer restart



In the example above, watchdog timer and usage of 75% interrupt is enabled by option byte. After reset release, the WDT - Interrupt has to be enabled for usage and WDT has to be restarted. The first restart of WDT timer after reset release is independent from window open period. There is no necessity to take care for window open period. Nevertheless, the restart has to be below watchdog timer overflow time.

A 75% watchdog timer interrupt service routine is implemented to set a marker bit that window close time has been elapsed for further action in the operating system.

In the endless loop of the operating system, one of its tasks is to detect if the marker bit is set. If the marker bit is set, the task has to restart the WDT and reset the marker bit.

Under these conditions, user has to take care that the microcontroller is able to restart the WDT within the remaining 25% of the WDT overflow time, before watchdog timer overflows, even at the highest load of the CPU. Don't forget that e.g. interrupts can lengthen the operation time.

The watchdog timer continues its operation during self-programming of the flash memory and EEPROM^{TM} emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

If the expected time might exceed the overflow time under any condition, a longer WDT overflow time has to be selected.

Caution It is not recommended to implement the WDT restart within the 75% interrupt service routine: If the CPU stucks in a task in an unexpected endless loop, WDT restart would be done in time within the interrupt service routine.

So, although the CPU is stucked, there is no watchdog timer - reset.

Therefore restarting of the WDT should be implemented within the usual operation flow.

Chapter 5 WDT operating conditions

5.1 Internal low-speed oscillation clock

The clock for the watchdog timer is the internal low-speed oscillator.

This circuit oscillates a clock of f_{IL} = 15 kHz (typical). Due to its nature the tolerance of this oscillator is relative high.

Only the following peripheral hardware runs on the internal low-speed oscillation clock.

- Watchdog timer
- Real-time clock
- Interval timer

The internal low-speed oscillator operates when

- Option byte 000C0H/010C0H bit 4: WDTON = 1 or
- Operation speed mode control register (OSMC) bit 4: WUTMMCK0 = 1 or
- WDTON = 1 and WUTMMCK0 = 1

5.2 Internal Low-speed oscillator and WDT during HALT/STOP- mode

WDSTBYON controls if WDT is stopped during HALT/STOP or not.

When WUTMMCK0 = 0 and option byte (000C0H/010C0H): WDTON = 1 and WDSTBYON = 0, oscillation of the internal low-speed oscillator and the watchdog timer stops if the HALT or STOP instruction is executed.

This reduces the power consumption during standby mode.

After standby release, the internal low-speed oscillator starts running and the watchdog timer is clocked again, starting at counter value = 00 automatically.

Note When WDSTBYON = 0, the window open period is 100%, regardless the setting of the WINDOW1 and WINDOW0 bits.

When option byte (000C0H/010C0H): WDTON = 1 and WDSTBYON = 1, oscillation of the internal low-speed oscillator and the watchdog timer keeps running during HALT- and STOP- mode.

The CPU has to be waked up sequentially to restart the watchdog. The 75% WDT interrupt can be used as well as the interval timer for sequential wake up of the CPU, re-triggering the WDT and entering standby mode again.

This increases the standby current, noticeable especially during STOP - mode.

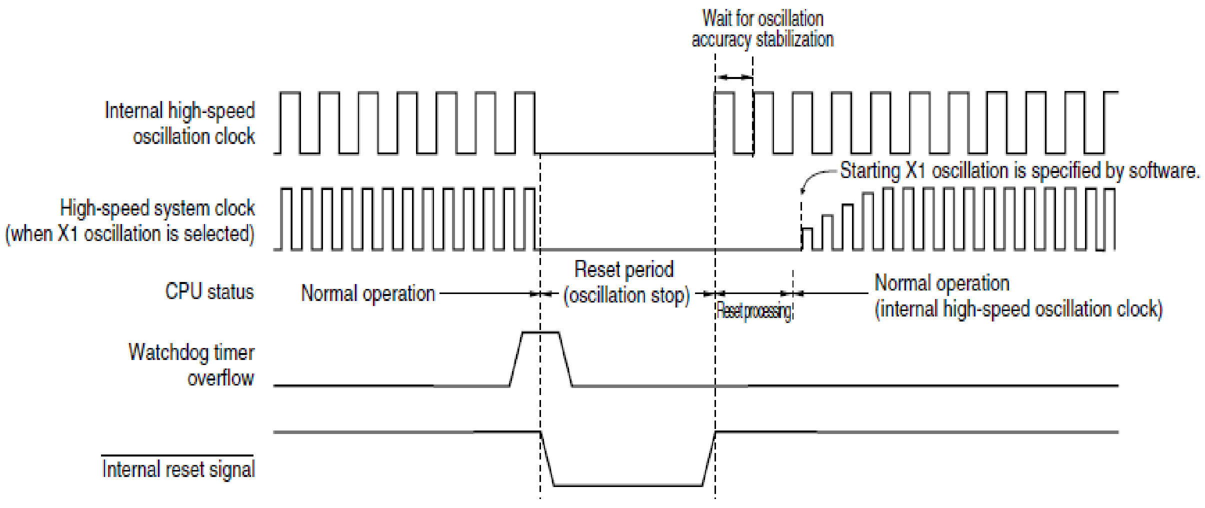
5.3 WDT reset generation

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

Figure 5-1 Timing of reset due to watchdog timer overflow


Wait for oscillation accuracy stabilization
Internal high-speed oscillation clock
Starting X1 oscillation is specified by software.
High-speed system clock (when X1 oscillation is selected)
CPU status
Normal operation
Reset period (oscillation stop)
Reset processing
Normal operation (internal high-speed oscillation clock)
Watchdog timer overflow
Internal reset signal

When a reset occurs triggered by the watchdog timer, the internal watchdog timer reset signal:

- sets the watchdog timer counter = 00H
- sets bit 4 of the reset control flag in register RESF to WDTRF = 1.

Figure 5-2 Reset Control Flag Register RESF, bit WDTRF

Address: FFFA8H After reset: 00H[Note 1] R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated by the watchdog timer.

Notes 1. The value after reset varies depending on the reset source.

After reset release the reset control flag register RESF can be tested. The reason for reset can be detected and the software can act accordingly.

Chapter 6 Option byte for window watchdog

6.1 Functions of Option Bytes in general

Addresses 000C0H to 000C3H of the flash memory of the RL78 form an option byte area.

Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).

Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set.

To use the boot swap operation during self programming, 000C0H - 000C3H are replaced by 010C0H - 010C3H.

Therefore, set the same values placed in 000C0H - 000C3H to 010C0H - 010C3H.

6.2 Settings of window watchdog option byte

located at address 000C0H/010C0H:

- Setting of 75% interval interrupt of watchdog timer
  - Used or not used
- Setting of window open period of watchdog timer
- Operation of watchdog timer
  - Operation is stopped or enabled.
- Setting of interval time of watchdog timer
- Operation of watchdog timer during standby
  - Operation is stopped or enabled in the HALT or STOP mode.

After reset release, the appropriate setting is valid for functionality.

6.3 Format of the User Option Byte 000C0H/010C0H

The selection for window watchdog / internal low-speed oscillator functionality is done by setting of the option byte at address 000C0H and 010C0H.

Figure 6-1 Format of watchdog option byte 000C0H/010C0H

Address: 000C0H/010C0H

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note]
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.19m s)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode[Note 2]
1	Counter operation enabled in HALT/STOP mode

Note The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

f_{IL}: Internal low-speed oscillation clock frequency

6.4 Setting of the Option Byte in the users source file

The option byte(s) has (have) a specific location within the microcontrollers address range. The address of the option byte is specified in the microcontrollers ".xcl - file", delivered by Renesas.

6.4.1 Setting of the Option Byte using relocatable segments

As an example, the following segment definition is included in a specific xcl - file for the watchdog timer - option byte.
The same has to be defined in address 010CxH, when boot swap is used in the application.

```
//-------------------------------------------------------------------------------------------------------------
// Allocate OPTION BYTE segment starting at address 000C0H, here 3 bytes specified
//-------------------------------------------------------------------------------------------------------------
-Z(CODE)OPTBYTE=000C0-000C2
```

(1) Example: Selection of the watchdogs behaviour, addressed by xcl - file

- In C - language

```
# pragma location = “OPTBYTE”        // Definition of the segment for the option byte ( .xcl – file)


__root const unsigned char myoptionbyte[ 3 ] = {0xB9, 0x16. 0xE8};

                                     // Watchdog timer 75% interrupt enabled
                                     // Window open period of watchdog timer: 50%
                                     // Watchdog timer operation enabled after reset release
                                     // Overflow time of watchdog timer: 2^11/fIL,
                                     // Watchdog timer active during HALT/STOP - mode
                                     // Second and third value just as example
```

- In assembler (IAR - Workbench)

```
RSEG   OPTBYTE       ; Reference to .xcl – file, address 0x000C0
DB     10111001B     ; Watchdog timer 75% interrupt enabled
                     ; Window open period of watchdog timer: 50%
                     ; Watchdog timer operation enabled after reset release
                     ; Overflow time of watchdog timer: 2^11/fIL,
                     ; Watchdog timer active during HALT/STOP - mode
DB     00010110B     ; Second value as example
DB     11101000B     ; Third value as example
```

6.4.2 Setting of the Option Byte using absolute addressing

- In C - language

```
__root const unsigned char myoptionbyte @ 0x000C0 = 0xB9;

                        // Absolute addressing at address 0x000C0
                        // Watchdog timer 75% interrupt enabled
                        // Window open period of watchdog timer: 50%
                        // Watchdog timer operation enabled after reset release
                        // Overflow time of watchdog timer: 2^11/fIL,
                        // Watchdog timer active during HALT/STOP - mode
```

- In assembler

```
ASEG
ORG     0x000C0         ; Absolute addressing at address 0x000C0
DB      10111001B       ; Watchdog timer 75% interrupt enabled
                        ; Window open period of watchdog timer: 50%
                        ; Watchdog timer operation enabled after reset release
                        ; Overflow time of watchdog timer: 2^11/fIL,
                        ; Watchdog timer active during HALT/STOP - mode
```

Chapter 7 Revision History

Table 7-1 Revision History

Version	Date	Change description
0100	5. April 2011	This is the initial version of the document.

RL78 Window Watchdog Application Note

Publication Date:	<Revision (previous)>	<Release Date (previous)>
		April 5, 2011

Published by: Renesas Electronics Corporation

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RL78 Window Watchdog

Renesas Electronics Corporation

R01AN0603ED0100
<DocumentNumber (previous)>