

Using the Watchdog Timer in the HT48 MCU Series

D/N : HA0019E

Introduction

Each device in the HT48 series of microcontrollers contains a watchdog timer. The purpose of the watchdog timer is to prevent problems should the device jump to an erroneous program location or if the device should enter an endless loop. In this application note, although the HT48R10A-1 device is used to demonstrate how to use the watchdog timer, the operation of the WDT in other microcontrollers is similar.

Usage

The WDT prescaler register has the name WDTS and is located at the Special Purpose Data Memory address 09H. When the WDT overflows the TO flag will be set. When the "CLR WDT" instruction is executed the TO flag will be cleared. The "CLR WDT" instruction will also clear the PD flag at the same time. The WDT may be enabled or disabled by a configuration option. If disabled any instructions related to the WDT will be ignored.

The watchdog timer clock source is determined by a configuration option, which in the case of this chosen device can be either the internal WDT clock (WDTCLK), the real time clock (RTC) or the instruction clock ($f_{SYS}/4$). When the WDT counter overflows, the device will be reset, the effects of which are shown in the reset tables in the respective datasheet. However if the device is already in the low power mode, achieved by the previous execution of a HALT instruction, when a WDT overflow occurs only a warm reset will be implemented. In this case only the PC and SP flags will be reset to 0.

Within the HT-IDE3000 options, choose the OSC option first to enable the corresponding clock source to appear as an WDT option. After this ensure that the correct Clear WDT option is selected.

The RTC clock can only be used if the internal RC oscillator is selected via configuration option as the system oscillator. Once the internal RC oscillator is selected as the system oscillator the RTC can be chosen as the WDT clock source. If the watchdog clock source is selected to be the instruction clock, it is important to note that in the HALT mode, the WDT will stop counting and lose its protective function. In this case, the system can only be restarted by external logic using the wake-up pins. For this reason it is recommended to use either the integrated RC oscillator (WDT oscillator) or the 32kHz RTC as the clock source.

To prevent the WDT from resetting the device to allow normal MCU operation to continue the WDT counter must be cleared before it overflows and issues a reset. There are three ways in which the contents of the WDT can be cleared, via an external reset, a software instruction or a HALT instruction. The software instructions include "CLR WDT", "CLR WDT1" and "CLR WDT2". Only one can be chosen by configuration option according to the needs of the application. If the "CLR WDT" is chosen, the contents of the WDT will be cleared only by a "CLR WDT" instruction. If "CLR WDT1" and "CLR WDT2" are chosen, then both instructions have to be executed in turn to clear the WDT contents. Otherwise the WDT will reset the system when it overflows.

The WDT overflow time is determined by the WDT clock source, which can be divided by a WDT prescaler, the value of which is determined by the WDTS register as shown in the following table. As only three bits are used in this register the remaining unused bits in the WDTS register can be accessed by the programmer for general purpose use. Note that in the case of the HT48R10A-1 device the WDT clock source is first divided by 256 before being further divided by the WDT prescaler.

WS2	WS1	WS0	Division Ratio
0	0	0	1 : 1
0	0	1	1 : 2
0	1	0	1 : 4
0	1	1	1 : 8
1	0	0	1 : 16
1	0	1	1 : 32
1	1	0	1 : 64
1	1	1	1 : 128

WDT Prescaler Division Ratio

The overflow formula is:

$TOVER = \text{WDT clock source time unit} \times 256 \times \text{prescaler division factor}$ (determined by WDTS). As an example, the longest overflow period can be calculated as follows. Taking the slowest clock source, which is the internal WDT oscillator with a maximum period of 130 μ s. Therefore the longest WDT overflow period will be given by 130 μ s (largest time unit) \times 256 (fixed value) \times 128 (largest division factor) = 4.26s

The shortest overflow period is when the fastest 8MHz system crystal oscillator is used. This gives a WDT clock time unit of 0.5 μ s. Therefore the shortest time out period is given by 0.5 μ s \times 256 (fixed) \times 1 (smallest division factor) = 128 μ s

The longest and shortest WDT overflow times for the various clock sources is shown below:

Clock Source	Longest Overflow Period	Smallest Overflow Period
WDTOSC (RC)	4.26s	8.19ms
32kHz (RTC)	1.00s	7812.5 μ s
System Frequency/4	327.68ms	128 μ s

As the frequency of the WDT oscillator varies with system voltage, the above table shows the parameters at $V_{DD}=5V$. The longest overflow period uses the biggest time unit of the correspondent clock source with a division ratio of 128. The shortest overflow period uses the fastest correspondent clock source with a division option of 1. Temperature effects are not considered.

Program Example

Example 1

```

Program name : wdtimeover.asm
Writer : Shan-yun Huang
Purpose : an introduction to using the watchdog timer
Options : wdtinstr option choose one clear instruction
Program list :
INCLUDE HT48R10A-1.INC
data .section 'data'
count1 db ?
code .section 'code'
org 10h ;first program section
sz status.5 ;TO dead loop if no overflow
jmp $ ;wait if (1)overflows
jmp start2
start: nop
clr wdt ;prevent WDT overflow
start2: ;dead loop area
nop
mov a,03h ;set the division factor=128
;clr wdt ;system reset will fail
mov wdts,a ;set WDT time
jmp start2 ;set the dead loop

```

- Example 1 program instruction

The program explains how to use the WDT. After a period of operation, the program pointer will stop at (1) to indicate the avoidance of a WDT dead loop. If it contains only one WDT clear instruction in the dead loop, the program still cannot be reset normally.

Example 2

```

;Program name:wdtimeover1.asm
;Writer : Shan-yun Huang
;Purpose : program structure using the "CLR WDT" pair of instructions
;Options : wdtinstr option take two clear instruction
;Program list :
INCLUDE HT48R10A-1.INC
data .section 'data'
count1 db ?
code .section 'code'
org 10h ;first program section
sz status.5 ;TO dead loop if no overflow
jmp $ ;if (1) overflows then wait
jmp start2
start: nop
clr wdt2 ;(2)will prevent WDT from overflow
start2: ;dead loop area
nop

```

```

mov a,03H                ;set division factor 128
clr wdt1                 ;(3)dead loop includes "CLR WDT"
mov wdts,a               ;set WDT time
jmp start2;dead loop set by(4)

```

- Example 2 program instruction

The program set a dead loop at (4) that contains the WDT clear instruction "CLR WDT1" at (3). However, the "CLR WDT2" instruction at (2) isn't in the loop, the WDT counter will keep counting and the program will be reset by the WDT and finally stop at (1) to prevent action from the dead loop.

Note: When using a pair of clear WDT instructions, the correct "TWO CLEAR INSTRUCTIONS" configuration option must be selected as the option to clear the WDT, otherwise it will be cleared by one instruction. The program cannot return to the default (1).

After the WDT overflows, the program will start from 000H. If ORG is used for program division, note that it must be restarted from the first section.

In the program, when a "CLR WDT" is included in the dead loop, there is no way to reset unless two "CLR WDT" instructions are combined.

Example 3

```

Program name : wdtime.asm
;Writer : Shan-yun Huang
;Purpose : show the overflow time setting and measure
;Options : osc is 32khzRTC+BUILTIN RC, TMR_CLK is 32KHZRTC
;Program list:
include ht48r10a-1.inc
;-----
data .section 'data'
    count1 equ    [40h]
    count2 equ    [41h]
    temp    equ    [42h]
;-----
code .section at 0h 'code'
org 00h
    jmp start
org 08h
    jmp intstart
start:
    sz STATUS.5
    jmp $
    clr count1
    clr count2
    clr temp
    mov a,07H                ;set division factor to 128
    mov WDTS,a               ;set WDT time

```

```

main:
    mov a,84h                ;use fSYS=32k RTC
    mov tmrc,a
    set intc.0                ;start main halt
    set intc.2                ;start timer
    mov a,00h                ;initial value 0
    mov tmr,a
    set tmrc.4                ;counting allowed
$2:
    snz status.5              ;wait for WDT overflow
    jmp $2
    clr tmrc.4                ;stop counting
    jmp $
intstart:
    inc count1                ;count every 256MS
    sz count1
    jmp $1
    inc count2                ;count longer delay
    sz count2
    jmp $1
    jmp $
$1:
    reti
end

```

- Example 3 instruction

The program will estimate the different overflow time of each clock source. Resetting the WDT overflow does not influence the TMR register, so it is OK to measure the overflow time of each WDT mode using the timer. The clock source of the timer is the 32kHz RTC that will count every 1ms. When the WDT overflows, the TO flag will be set and repeatedly detected by the program. The timer will stop when the TO flag changes state.

When the clock source is chosen to be the internal WDT clock (normal condition) the RC oscillator frequency is 65μs/5V.

Theoretical value :

Clock Source	WDT Overflow Period	Options
WDTOSC	$65\mu s \times 256 \times 128 = 2.1299s$	WDT is RC OSC
32kHz (RTC)	$31.25\mu s \times 256 \times 128 = 1.024s$	WDT is 32kHz RTC
System Frequency/4	$1 \times 256 \times 128 = 0.032768s$	WDT is T1

The WDTS Division Ratio is Chosen to be 128

Note: System Frequency is 4000kHz

Measured Value

The formula :

$$TOVER = 1ms \times 256 \times COUNT1 + TMR$$

Clock source	WDT Time Period	Relative Error
WDTCLK	$9 \times 256 + 256 = 2.56S$	0.1%
32kHz (RTC)	$4 \times 256 + 1 = 1.025S$	0.1%
System Frequency/4	21H=33ms	0.7%

Note: The program execution error is not considered in this case. In addition, the time base signal waveform may also cause errors.
