Using the Watchdog Timer in the HT47R20A-1

D/N：HA0028E

Introduction

The main purpose of this Application Note is to show how to use the Watchdog Timer and special notes that should be taken into consideration.

The HT47 series MCU has a single internal watchdog timer circuit. The watchdog timer is provided to prevent software malfunctions or sequences from jumping to unknown locations with unpredictable results. 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 system clock divided by 4. One of the WDT clock source is the internal RC oscillator, which requires no external components. However, if the clock source selects the instruction clock and the ″HALT″ instruction is executed, the WDT may stop counting and lose its protecting purpose, in this case the device can only be restarted by external logic. When the device operates in a noisy environment, using the on-chip RC oscillator (WDT OSC) is strongly recommended, since the HALT instruction will stop the system clock.

The WDT overflow under normal operation will initialise a ″chip reset″ and set the status bit TO. Whereas in the HALT mode, the overflow will initialize a ″warm reset″ and only the PC and SP are reset to 0. To clear the contents of the WDT, three methods are adopted; external reset, software instruction, or a HALT instruction. The software instructions are of two types, the first is a single CLR WDT instruction and the other is a pair of instructions - CLR WDT1 and CLR WDT2. Of these two types of instruction, only one can be active depending on the configuration option – ″CLR WDT time selection option″. If the ″CLR WDT″ is selected (i.e., CLR WDT times equal one), any execution of the CLR WDT instruction will clear the WDT. In case ″CLR WDT1″ and ″CLR WDT2″ are chosen (i.e. CLR WDT times equal two), these two instructions must be executed to clear the WDT; otherwise, the WDT may reset the chip due to time-out.

The WDT time-out period is clock source fs/2^{15}. The following table shows the clock source and the WDT overflow period:

Clock Source	WDT Overflow Period
12kHz (WDTCLK)	2.73066s
32kHz (RTC)	1.00000s
System Frequency/4	0.27306s

Note: System frequency is 480kHz。

There are three ways in which a reset can occur.

- $\overline{\text{RES}}$ reset during normal operation
- $\overline{\text{RES}}$ reset during HALT
- WDT time-out reset during normal operation

The WDT time-out during a HALT is different from other chip reset conditions, since it can perform a warm reset that just resets the PC and SP leaving the other circuits in their original state. Some registers remain unchanged during other reset conditions. Most registers are reset to their initial condition when the reset conditions are met. By examining the PD and TO flags, the program can distinguish between different chip resets.

TO	PDF	RESET Conditions
0	0	$\overline{\text{RES}}$ reset during power-on
u	u	$\overline{\text{RES}}$ reset during normal operation
0	1	$\overline{\text{RES}}$ wake-up from HALT
1	u	WDT time-out reset during normal operation
1	1	WDT time-out during HALT

Note: “u” stands for unchanged

At the beginning of the program the following statements can be added, the reason for the reset can then be determined:

```
……
start：
     sz   to                  ;check flag state TO bit
     jmp  wdt_ov              ;WDT causes a system chip reset
     ……
wdt_ov： ……
```

Explanation

The ″CLR WDT″ instruction and ″CLR WDT1″ instruction codes are the same, that is, the two instructions have the same effects.

″CLR WDT1″ and ″CLR WDT2″ must simultaneously be executed to clear the watchdog timer, however if there's duplication, then the watchdog timer cannot be cleared. Example:

```
.....
CLR WDT1            ; clear watchdog timer
.....
CLR WDT2            ; clear watchdog timer
.....
CLR WDT1            ; clear watchdog timer
.....
CLR WDT1            ; can't clear watchdog timer
.....
CLR WDT2            ; clear watchdog timer
.....
CLR WDT2            ; can't clear watchdog timer
.....
CLR WDT1            ; clear watchdog timer
```

The purpose of having two instructions is that even if there is a clear watchdog timer instruction (for example CLR WDT1), there still remains the possibility that the program can enter a dead loop condition. The function of the commands ″CLR WDT1″ and ″CLR WDT2″ is to avoid this kind of situation. The following shows an example of a program within a loop:

```
LOOP :
    .....
    CLR  WDT1       ;loop, clear WDT1 to ″0″
    .....
    SZ   .....
    JMP  LOOP
    CLR  WDT2       ;after loop, clear WDT2 to ″0″
    .....
```

During the first execution of the ″CLR WDT1″ instruction, the WDT will be cleared. If however there is a loop, then it returns in a circular loop process to execute again the ″CLR WDT1″ instruction. The execution of the second ″CLR WDT1″ instruction, will now have no effect and a WDT overflow occurs, the system resets, and jumps out of the dead loop condition.