Version 1.1

eRIC Watch Dog Timer:

Watch Dog Timer by default works in interval mode. This means, watchdog timer interrupt flag is set or runs the watch dog timer interrupt vector at the expiration of the selected time interval. No PUC is generated in this mode.

Refer eRIC_eROS_Developers_Manual_x.x on wards for complete list of WDT definitions. Refer SLAU259E document by Texas Instruments to use core registers.

Setting WatchDogTimer:

eRIC_WDT_Setup(Modebits): This sets the watch dog timer with clocksource and no of intervals needed. Modebits can be the sum of clocksource and interval modes as explained below:

There are three clocksources available.

a) eRICWDT_Cs_CPU :This can be used to choose SMCLK(upto 20Mhz frequency set by eRIC_SetCpuFrequency()) as clock source for WDT.
b) eRICWDT_Cs_32k :The WDT clock is sourced by 32768Hz clock
c) eRICWDT_Cs_10k : The WDT clock is sourced by 10000Hz clock

There are eight interval modes available.

a) eRICWDT_Interval_64 :WDT triggers after 64 clock cycles
b) eRICWDT_Interval_512 :WDT triggers after 512 clock cycles
c) eRICWDT_Interval_8192 :WDT triggers after 8192 clock cycles
d) eRICWDT_Interval_32768 :WDT triggers after 32768 clock cycles
e) eRICWDT_Interval_524288 :WDT triggers after 524288 clock cycles
f) eRICWDT_Interval_8388608 :WDT triggers after 8388608 clock cycles
g) eRICWDT_Interval_134217728 :WDT triggers after 134217728 clock cycles
h) eRICWDT_Interval_2147483648 :WDT triggers after 2147483648 clock cycles

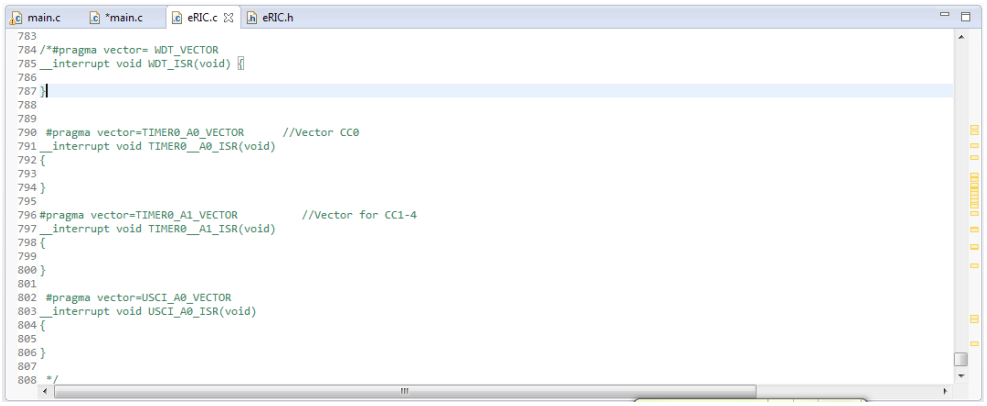
For example:

a) eRIC_WDT_Setup(eRICWDT_Cs_32k+ eRICWDT_Interval_32768); will set the WDT with 32k as clock source and triggers an interrupt after every 32768 cycles, which would be 1second on 32k clock.

b) eRIC_WDT_Setup(eRICWDT_Cs_10k + eRICWDT_Interval_32768); will set the WDT with 10k as clock source and triggers an interrupt after every 32768 cycles, which would be 3.276seconds on 10k clock.

eRIC_WDT_Start(); This will start the watch dog timer
eRIC_WDT_Stop(); This will stop the watch dog timer
eRIC_WDT_Reset(); This will reset the watch dog timer
eRIC_WDT_InterruptEnable(); This will enable WDT interrupt when interrupt mode is used.
eRIC_WDT_InterruptDisable(); This disabled WDT interrupt.
eRIC_WDT_ClearInterruptFlag(); This clears WDT interrupt flag
eRIC_WDT_HasInterrupted(); This is used to check if flag is set

Note: If Watch dog timer interrupt vector is used in main, it needs to be removed in eRIC.c as shown below:



SAMPLE CODE: The code below toggles Pin19 every second using WDT

```
1)  #include<cc430f5137.h>
2)  #include "eRIC.h"
3)  int main(void)
4)  {
5)  eRIC_GlobalInterruptDisable();                  //Global interrupts disabled
6)  Pin19_SetAsOutput();                            //Pin19 set as output
7)  eRIC_WDT_Setup(eRICWDT_Cs_32k+eRICWDT_Interval_32768); //Sets up WDT
8)  eRIC_WDT_Start();                               //Starts WDT
9)  eRIC_WDT_InterruptEnable();                     //WDT interrupt is enabled
10) eRIC_GlobalInterruptEnable();                   //Global interrupts enabled
11) }
12) #pragma vector= WDT_VECTOR
13) __interrupt void WDT_ISR(void)                  //WDT interrupt vector
14) {
15) eRIC_WDT_Stop();                               //Stops WDT
16) Pin19_Toggle();                                //Toggles Pin19 every second
17) eRIC_WDT_Reset();                              //Resets WDT timer
18) eRIC_WDT_Start();                              //Starts WDT
19) }
20) void eRIC_RfDataReceivedInterrupt()   //Add code here to deal with
21) available received data..This is triggered when interrupt is enabled and a
22) packet is received
23) {
24) }
```

Line1 and line2 includes cc430F5137 and eRIC.h which is must for any program code. Main starts at line3.

Global interrupts are disabled at Line5. Pin19 is set as output at Line6.Watchdog timer is setup with 32k clock source and 32768 intervals at Line7. WDT is started at Line8. WDT and global interrupt are enabled at Line9 and 10.

Interrupt vector is defined at Line12. And Pin is toggled whenever interrupt is triggered at Line16.

eRIC_RfDataReceivedInterrupt() is copied from eRIC.c which is removed and pasted in main at line 20-24. This has no effect in this example as there is no receiver enabled. But this code is needed for compiler to compile without error or un-remove this same code in eRIC.c.