MicroConverter® Technical Note – uC018
Uses of the Time Interval Counter

Introduction

A Time Interval Counter is provided on-chip for counting longer intervals than the standard 8051-compatible timers are capable of. The TIC is capable of timeout intervals ranging for 1/128th second to 255 hours.

The TIC is present on the following MicroConverter parts

- ADuC814
- ADuC816
- ADuC824
- ADuC831
- ADuC832
- ADuC834

The TIC is clocked by the 32.768KHz watch crystal directly on the ADuC814, ADuC816, ADuC824, ADuC832,ADuC834 and by a r/c oscillator on the ADuC831. Since the TIC is clocked from a source independent of the master clock it can remain active during powerdown. The TIC can then be used to generate an interrupt after a specified interval and 'wake up' the MicroConverter from powerdown.

This technote describes how to configure the Time Interval Counter to 'wake up' the microconverter every 10 minutes and perform an ADC conversion.

Time Interval Counter SFR's

Six SFRs are associated with the time interval counter. TIMECON is the control register. A brief summary of the TIMECON sfr is given below, further details can be seen in the relevant datasheet.

TIMECON.7	RESERVED
TIMECON.6	This bit must contain '1'
TIMECON.5	Timebase selection, ITS1,ITS0 [128th sec, one sec,
TIMECON.4	one min, one hour]
TIMECON.3	Single Time Interval control bit (0=reload & restart)
TIMECON.2	Time interval interrupt bit 'TII'
TIMECON.1	Time interval enable bit (0=disable & clear)
TIMECON.0	Time clock enable bit (0=disable)

The other SFRs associated with the Time Interval Counter are

INTVAL	User required time interval
HTHSEC	Increments in 1/128 intervals in clock mode
SEC	Increments in seconds in clock mode
MIN	Increments in minutes in clock mode
HOUR	Increments in hours in clock mode

Time Interval Counter Operation

Depending on the configuration of the timebase selection bits (ITS1 and ITS0) the selected time counter register overflow will clock the interval counter. When this counter is equal to the time interval value loaded in the INTVAL SFR, the TII bit is set and generates an interrupt if enabled.

A block diagram of this is shown below

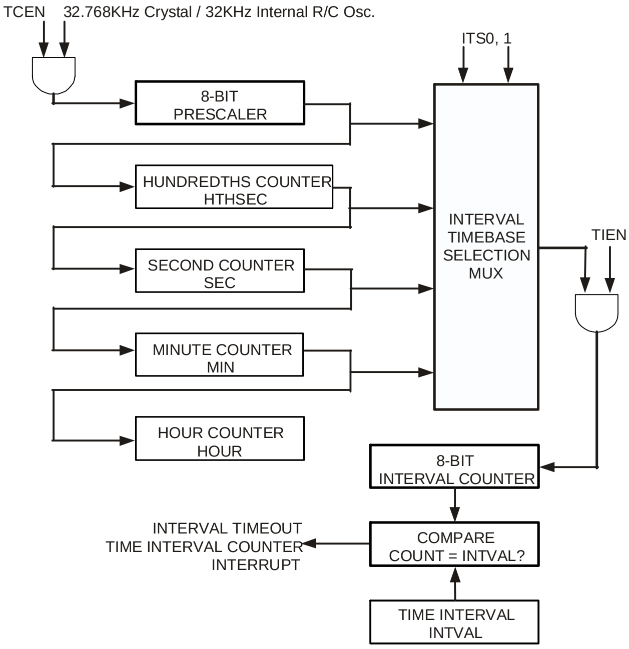


Figure 1. Block Diagram of the TIC structure

In clock mode the HTHSEC overflows after 127 counts and increments the MIN SFR, and so on. The timebase SFRs can be written initially with the current time facilitating the implementation of a real time clock.

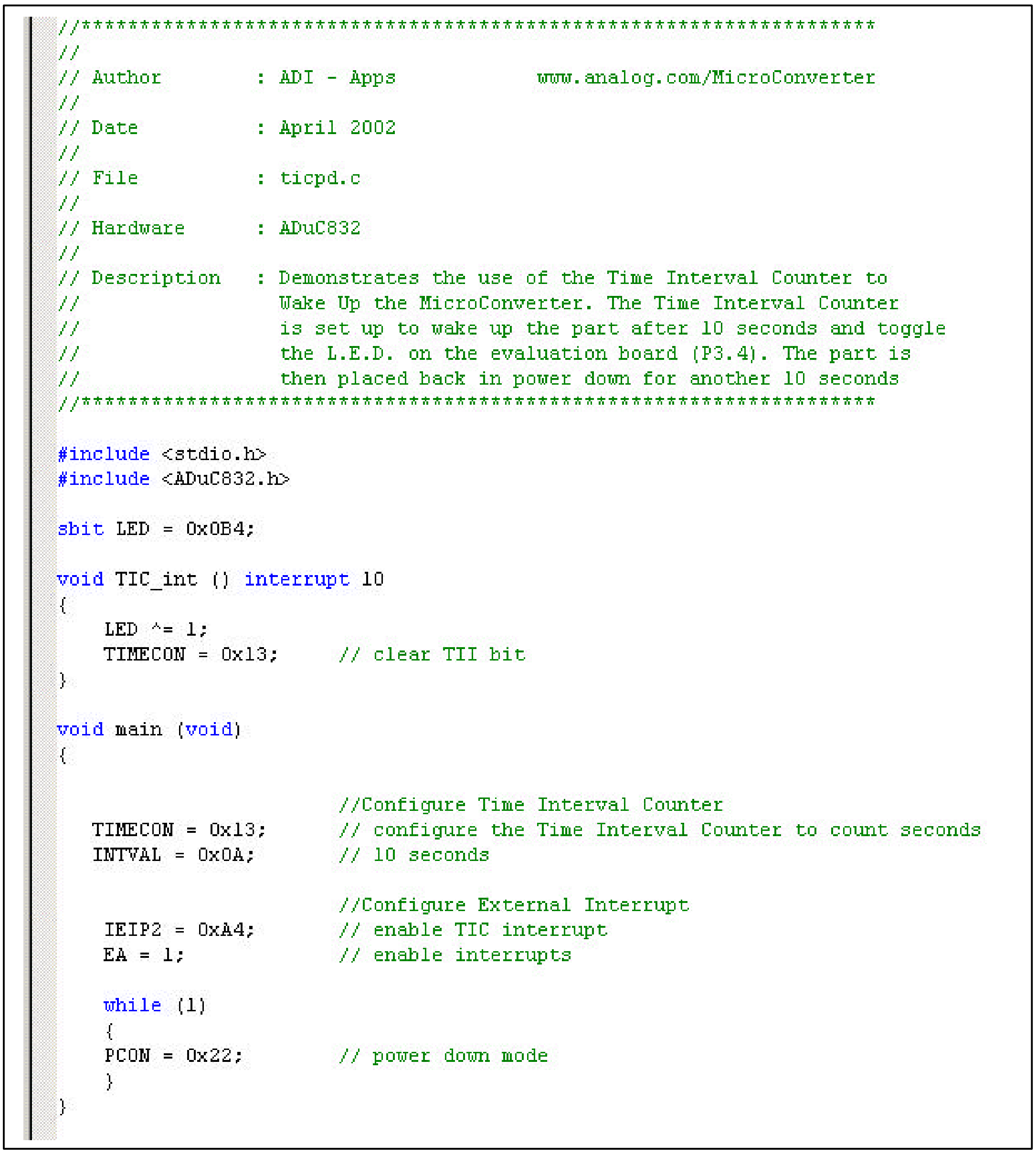
If the Timer Interval Counter interrupt is enabled via the IEIP2 register the program counter will vector to the ISR and execute the code there. If the part is in power down with the Time Interval Counter enabled when the interrupt occurs the part will 'wake up' and continue executing code, starting with the TIC ISR.

This allows the part to be placed into powerdown and programmed to wake at specific intervals. This makes the MicroConverters ideal for remote battery-powered sensors where regular widely spaced readings are required.

The following examples show the various uses of the Time Interval Counter. The code in the following examples is written for the ADuC832, though it can be used with any of the MicroConverter parts with a TIC by adjusting the baudrate and including the correct header file.

Wake Up from Powerdown Example

In this example the TIC is configured to timeout after 10 seconds.
The part is placed in powerdown with the TIC running. After 10 seconds the TIC will powerup and will start executing code at the TIC interrupt service routine. This will complement the L.E.D. on P3.4. The part then enters powerdown again.

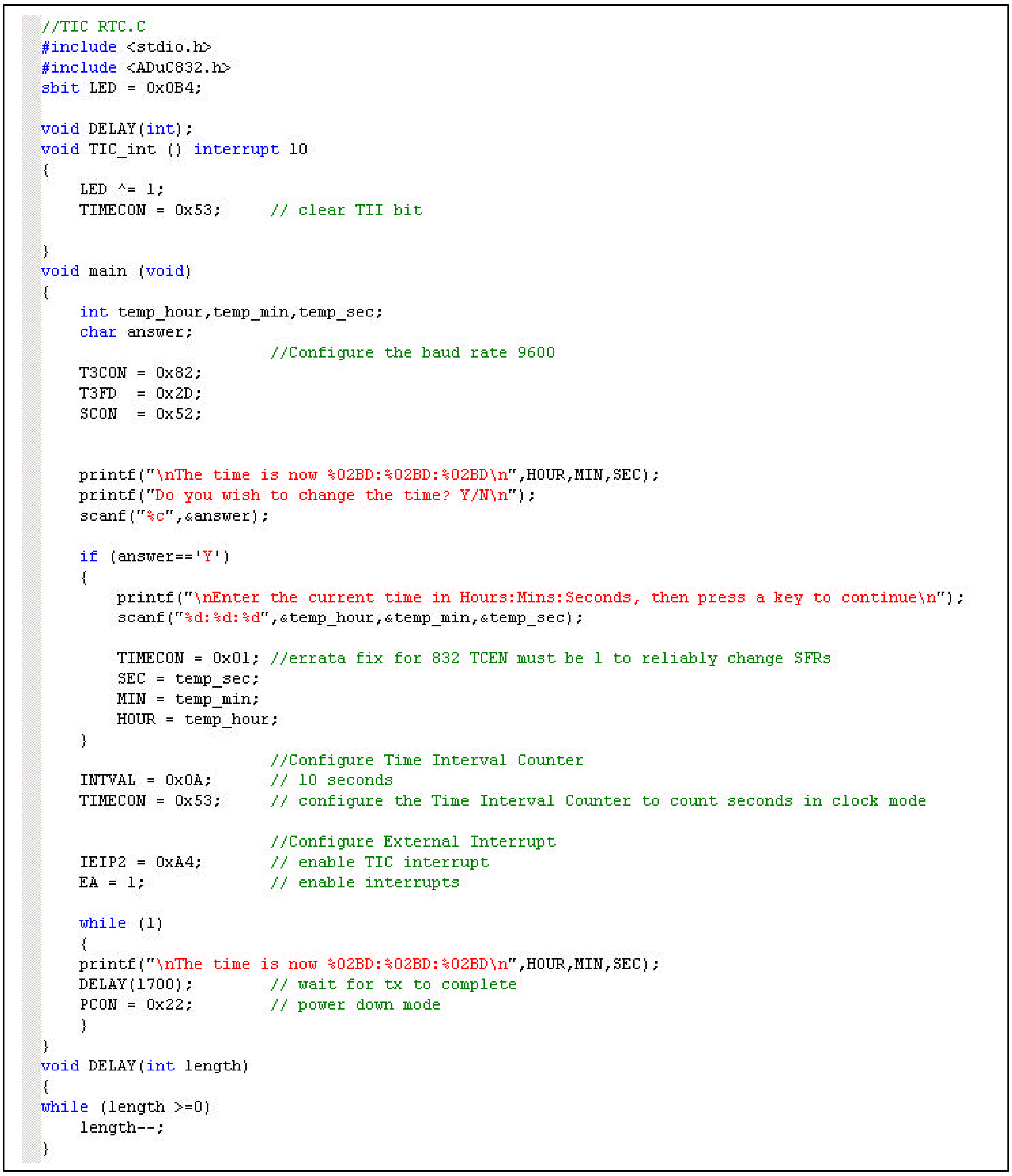


```
//********************************************************************
//
// Author        : ADI - Apps            www.analog.com/MicroConverter
//
// Date          : April 2002
//
// File          : ticpd.c
//
// Hardware      : ADuC832
//
// Description   : Demonstrates the use of the Time Interval Counter to
//                 Wake Up the MicroConverter. The Time Interval Counter
//                 is set up to wake up the part after 10 seconds and toggle
//                 the L.E.D. on the evaluation board (P3.4). The part is
//                 then placed back in power down for another 10 seconds
//********************************************************************

#include <stdio.h>
#include <ADuC832.h>

sbit LED = 0x0B4;

void TIC_int () interrupt 10
{
    LED ^= 1;
    TIMECON = 0x13;     // clear TII bit
}

void main (void)
{

                        //Configure Time Interval Counter
   TIMECON = 0x13;      // configure the Time Interval Counter to count seconds
   INTVAL = 0x0A;       // 10 seconds

                        //Configure External Interrupt
    IEIP2 = 0xA4;       // enable TIC interrupt
    EA = 1;             // enable interrupts

    while (1)
    {
    PCON = 0x22;        // power down mode
    }
}
```

List 1. Wakeup from powerdown example code

Real Time Clock Example

In this example the Time Interval Counter is set up to function as a real time clock. The user is prompted to enter the current time, after that the time is displayed at ten second intervals.

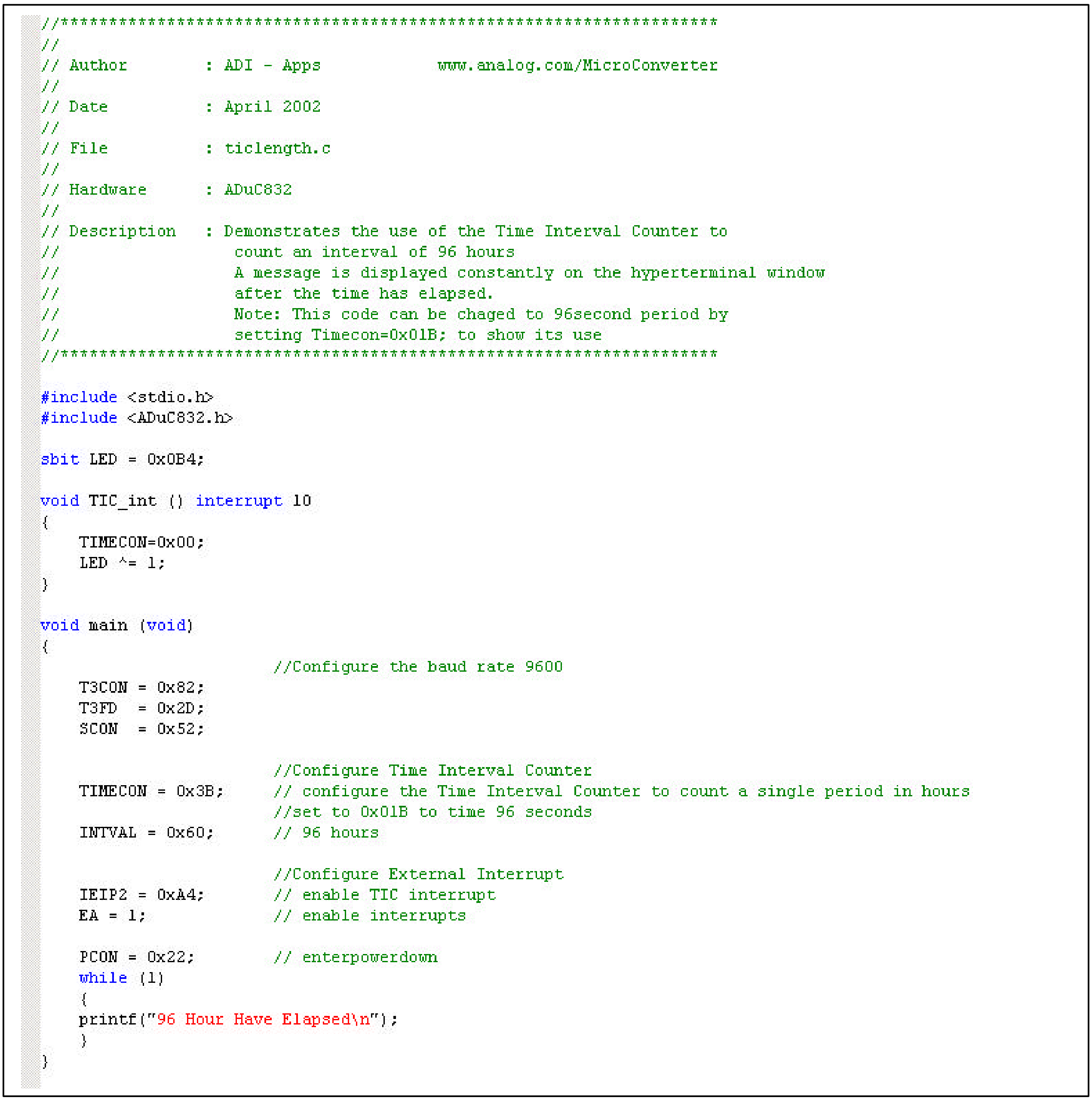


```c
//TIC RTC.C
#include <stdio.h>
#include <ADuC832.h>
sbit LED = 0x0B4;

void DELAY(int);
void TIC_int () interrupt 10
{
    LED ^= 1;
    TIMECON = 0x53;     // clear TII bit

}
void main (void)
{
    int temp_hour,temp_min,temp_sec;
    char answer;
                        //Configure the baud rate 9600
    T3CON = 0x82;
    T3FD  = 0x2D;
    SCON  = 0x52;


    printf("\nThe time is now %02BD:%02BD:%02BD\n",HOUR,MIN,SEC);
    printf("Do you wish to change the time? Y/N\n");
    scanf("%c",&answer);

    if (answer=='Y')
    {
        printf("\nEnter the current time in Hours:Mins:Seconds, then press a key to continue\n");
        scanf("%d:%d:%d",&temp_hour,&temp_min,&temp_sec);

        TIMECON = 0x01; //errata fix for 832 TCEN must be 1 to reliably change SFRs
        SEC = temp_sec;
        MIN = temp_min;
        HOUR = temp_hour;
    }
                        //Configure Time Interval Counter
    INTVAL = 0x0A;      // 10 seconds
    TIMECON = 0x53;     // configure the Time Interval Counter to count seconds in clock mode

                        //Configure External Interrupt
    IEIP2 = 0xA4;       // enable TIC interrupt
    EA = 1;             // enable interrupts

    while (1)
    {
    printf("\nThe time is now %02BD:%02BD:%02BD\n",HOUR,MIN,SEC);
    DELAY(1700);        // wait for tx to complete
    PCON = 0x22;        // power down mode
    }
}
void DELAY(int length)
{
while (length >=0)
    length--;
}
```

List 2. Real Time Clock example code

Timing Long Intervals

In this example the Time Interval Counter is configured to time a period of 96 hours. After this time has elapsed the part wakes up from powerdown and displays a message on the UART indicating the specified interval has elapsed.



```
//********************************************************************
//
// Author        : ADI - Apps           www.analog.com/MicroConverter
//
// Date          : April 2002
//
// File          : ticlength.c
//
// Hardware      : ADuC832
//
// Description   : Demonstrates the use of the Time Interval Counter to
//                  count an interval of 96 hours
//                  A message is displayed constantly on the hyperterminal window
//                  after the time has elapsed.
//                  Note: This code can be chaged to 96second period by
//                  setting Timecon=0x01B; to show its use
//********************************************************************

#include <stdio.h>
#include <ADuC832.h>

sbit LED = 0x0B4;

void TIC_int () interrupt 10
{
    TIMECON=0x00;
    LED ^= 1;
}

void main (void)
{
                        //Configure the baud rate 9600
    T3CON = 0x82;
    T3FD  = 0x2D;
    SCON  = 0x52;

                        //Configure Time Interval Counter
    TIMECON = 0x3B;     // configure the Time Interval Counter to count a single period in hours
                        //set to 0x01B to time 96 seconds
    INTVAL = 0x60;      // 96 hours

                        //Configure External Interrupt
    IEIP2 = 0xA4;       // enable TIC interrupt
    EA = 1;             // enable interrupts

    PCON = 0x22;        // enterpowerdown
    while (1)
    {
    printf("96 Hour Have Elapsed\n");
    }
}
```

List 3. Timing Long Intervals code example