Semiconductors

AN10382

Using the RTC efficiently in the LPC213x

Rev. 01 — 6 June 2005 Application note

Document information

Info	Content
Keywords	RTC, External 32.768 KHz crystal, LPC2000, LPC213x
Abstract	Provides information on reducing current consumption from V_{bat} while the ARM7 core is in the active mode of operation

PHILIPS

Revision history

Rev	Date	Description
01	20050606	Initial version

Contact information

For additional information, please visit: **http://www.semiconductors.philips.com**

For sales office addresses, please send an email to: **sales.addresses@www.semiconductors.philips.com**

1. Introduction

The Real Time Clock (RTC) is a set of counters for measuring time when system power is on, and optionally when it is off. It uses little power in Power-down mode. On the LPC213x, the RTC can be clocked by a separate 32.768 KHz oscillator, or by a programmable prescale divider based on the VPB clock. Also, the RTC is powered by its own power supply pin, V_{bat}, which can be connected to a battery or to the same 3.3 V supply used by the rest of the device.

In this application note, we discuss the most efficient way of using the RTC (active mode only) in the LPC213x family. It is recommended that the RTC should run from the external 32.768 KHz crystal connected to the RTCX1 and RTCX2 pins while the core is in the active mode of operation. Current is drawn from V_{bat} even if the RTC is running from V_{DD}. Less current is drawn from V_{bat} if the RTC uses the external crystal instead of VPB clock.

The RTC needs to be initially configured using the VPB clock. During the configuration, the clock source of the RTC can be selected to be the external 32.768 KHz crystal. Once the RTC time counters are enabled (using the Clock Control register), the VPB clock to the RTC can be disconnected using the PCONP register.

The VPB clock is only needed for initial configuration. Disconnecting the VPB clock does not affect the RTC functionality in any way. The RTC registers can still be read and also interrupts from the RTC can be serviced. The code example below verifies this observation.

2. RTC code example

The code has been compiled using the Keil ARM C compiler and is tested on the Keil MCB2130 board. In the C code provided, the following is done:

1. The RTC is configured using the internal VPB clock. It is configured to generate interrupts on counter (seconds, minutes, hours) increments.
2. Once the RTC is started, the VPB clock can be disconnected using the PCONP register.
3. On an IRQ, the Consolidated Timer Registers are read, the values of which are output on the UART. The interrupt is then cleared in the RTC Interrupt Service Routine (ISR).

2.1 C code

```
/* ---------------------------------------------------------------------------------------- */

/* LPC213x definitions  */

#include <LPC213x.H>
#include<stdio.h>
/* Mask values used while reading from Consolidated Time Registers */
#define MASKSEC     0x3f
#define MASKMIN    0x3f00
#define MASKHR      0x1f0000
/* Serial port Initializations are done in a different file */
extern void init_serial(void);
void read_rtc(void)__irq;
```

```
void Initialize(void);


/*  ------------------- Main -------------------  */
int main (void) {

int i;
/* Initialize System */
Initialize();
/* Initialize Serial port */
init_serial();
printf("\nRTC in 213x");

/* Start RTC */
CCR=0x11;
/* Disconnect pclk to RTC */
PCONP=0x1815BE;

/* Loop forever */
while (1)
 {
 }
}

/*  --------------------- Initialize ---------------------  */
void Initialize()
  {

/* VPBDIV setting */
VPBDIV=0x0;

/* Intialize RTC  */
CCR=0x2;  /* Reset the clock */
ILR=0x3; /* Clear the Interrupt Location Register */
HOUR=0x0;
SEC=0x0;
MIN=0x0;

/* Initialize Vectored Interrupt Controller (VIC) */
VICIntSelect=0x000;
VICIntEnable =0x2000;
VICVectCntl0=0x2d;
VICVectAddr0=(unsigned long )read_rtc;
   }

/*  ----------------- RTC Interrupt Service Routine -----------------  */
void read_rtc()  __irq
{
int hour=0;
int min=0;
int sec;
```

```
/* Clearing Interrupt */
ILR=0x1;

/* Reading Consolidated Time registers */
hour=(CTIME0 & MASKHR)>>16;
min= (CTIME0 & MASKMIN)>>8;
sec=CTIME0 & MASKSEC;

/* Printing the time on Serial port */
printf("\nTime is %02d:%02x:%02d",hour,min,sec);

/* Updating VIC */
VICVectAddr=0xff;
}
```

2.2 Sample code output

Screenshot from Terminal program Tera Term Pro 2.3 is shown in Figure 1. Any terminal program such as Hyperterminal can be used to view the output.

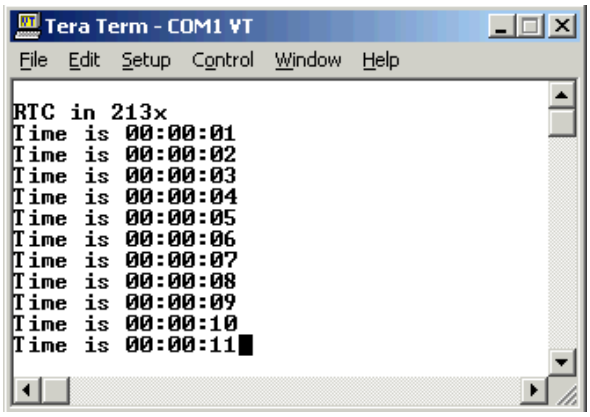

Fig 1. Sample code output

3. Current consumption from V_{bat}

Table 1 provides typical I_{bat} current consumption values under active mode conditions. For Table 1, consider the core clock to be the same as the peripheral clock (VPB clock).

Table 1: I_{bat} current consumption

VPB clock	I_{bat} (typical)
60 MHz	280 μA
25 MHz	130 μA
0 MHz	17 μA

4. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

5. Trademarks

Notice — All referenced brands, product names, service names and trademarks are the property of their respective owners.

6. Contents

Date of release: 6 June 2005

Published in The Netherlands