Keywords: DS32X35, DS32B35, DS32C35

Mar 23, 2007

APPLICATION NOTE 3974

Using a DS32x35 with an 8051-Type Microcontroller

Abstract: This application note describes how to interface the DS32x35 real-time clock with an 8051-type microcontroller. It also provides a schematic of the application circuit and example software code for implementing basic operating routines.

Overview of the DS32x35

The DS32x35 real-time clock (RTC) is a temperature-compensated clock/calendar that includes an integrated 32.768kHz crystal, an I²C interface, and a bank of nonvolatile memory in a single package. The DS32x35's integration of a crystal resonator enhances the long-term accuracy of the device and reduces the piece-part count in a manufacturing line.

The RTC provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Additionally, the clock provides two programmable time-of-day alarms and a programmable square-wave output.

Interfacing the DS32x35 to a µC

This application note demonstrates how to interface the DS32x35 to an 8051-type microcontroller, such as the DS2250 soft-microcontroller module. Additionally, **Figure 1** provides example code, written in C, for implementing basic interface routines, including routines for reading the time and date from the real-time clock (RTC); writing time and date data to the RTC from user entries; writing and reading data to and from the FRAM; reading the temperature from the temperature sensor; and selecting the proper FRAM density. The software supports both FRAM densities: 16kb (DS32B35) and 64kb (DS32C35).

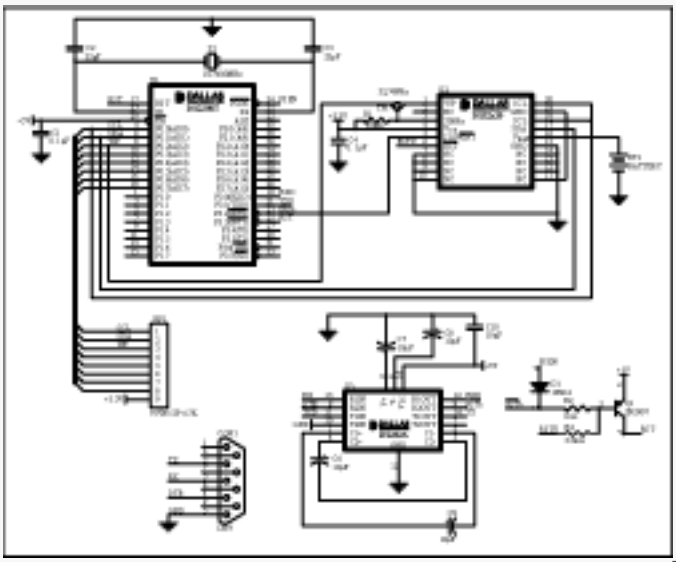
Figure 2 is a schematic that illustrates the connections required for interfacing the DS32x35 RTC with an 8051-type microcontroller. The program uses two general-purpose (GP) port pins on the microcontroller to communicate with the DS32x35 using an I²C serial interface; a third GP port pin is used to enable or disable the write-protect (WP) input pin for the FRAM. The microcontroller's internal UART drives two port pins, which are connected to a DS232 line driver/receiver. User inputs and data outputs from the program are passed through the RS-232 interface from a PC's terminal-emulator program to the microcontroller. The RS-232 DTR input line controls the microcontroller's reset input.

```
/*****************************************************************/
/* DS32X35sm.c                                                   */
/*****************************************************************/
#include <stdio.h>            /* Prototypes for I/O functions     */
#include <DS5000.h>           /* Register declarations for DS5000 */
/************************** Defines ****************************/
/************************ bit definitions ***********************/
#define     ACK   0
#define     NACK  1
#define     ADDRTC      0xd0  /* base address of TCI0 */
#define     ADDRAM 0xa0 /* base address of IRAM */
sbit  scl = P0^0;             /* IIC pin definitions */
sbit  sda = P0^1;
sbit  wp  = P0^2;       /* WP input */
sbit  sqwint = P3^3;    /* SQW/INTb output to INT1 on 8051 */
/********************** function prototypes *********************/
void  start();
void  stop();
uchar writebyteiic(uchar d);
uchar readbyteiic(int);
void  readbytesrtc();
void  writebytesrtc();
void  initialize_rtc();
void  rd_temp();
void  disp_regs(uchar);
void  fill_ram(uchar);
void  wr_ram(uchar);
void  ver_ram(uchar);
void  rd_ram();
/************************ Global Variables **********************/
uchar ram = 7, sec, min, hr, dy, dt, mn, yr;
/************************** Functions ***************************/
void start()            /* ------------------------------------------- */
{
  sda = 1;  scl = 1;    scl = 1; scl = 1; /* Initiate start condition */
  sda = 0;
}
void stop()       /* ------------------------------------------- */
{
  sda = 0;  sda = 0;  sda = 0;  sda = 0;  /* Initiate stop condition */
  scl = 1;  scl = 1;  sda = 1;
}
uchar writebyteiic(uchar d)   /* ---------------------------- */
{
uchar i;

      scl = 0;
      for (i = 0; i < 8; i++)
      {
            sda = (d & 0x80);
            scl = 1;
            d = d << 1;
            scl = 0;
      }
      sda = 1;    /* Release the sda line */
      scl = 1;
      i = sda;
      scl = 0;
      return i;
}
uchar readbyteiic(int b)      /* ---------------------------------- */
{
```

[Download Program Files](#) (ZIP, 4kB)

Figure 1. Program Listing



[More detailed image](#) (PDF, 260kB)

Figure 2. A schematic illustrating the connections for interfacing the DS32x35 with an 8051-type microcontroller.

Application note 3974: www.maxim-ic.com/an3974

More Information
For technical support: www.maxim-ic.com/support
For samples: www.maxim-ic.com/samples
Other questions and comments: www.maxim-ic.com/contact

Automatic Updates
Would you like to be automatically notified when new application notes are published in your areas of interest?
Sign up for EE-Mail™.

Related Parts

DS32B35: QuickView -- Full (PDF) Data Sheet -- Free Samples

DS32C35: QuickView -- Full (PDF) Data Sheet -- Free Samples

AN3974, AN 3974, APP3974, Appnote3974, Appnote 3974