RESEARCH ARTICLE OPEN ACCESS

Interfacing Of PIC 18F252 Microcontroller with Real Time Clock via I2C Protocol

Arnab G. Samanta
(Electronics Design technology (EDT) department, R.C.O.E.M, RTM Nagpur University, India)

ABSTRACT
This paper describes a microcontroller based digital clock which can be used in real time systems. The system is constructed using PIC18F252 (microcontroller), DS1307 (real time clock IC) and its software program is written with C programming language. A 3v battery backup is provided to real time clock IC. Communication between PIC microcontroller and DS1307 takes place through I²C Bus protocol
.***Keywords* -** I2C, LCD, PIC, RTC, DS1307

I. INTRODUCTION

Now a days electronic clocks have predominately replaced the mechanical clock. They are much reliable, accurate, maintenance free and portable. In general, there are two kinds of electronic clocks. They are analog clock and digital clock. But digital clocks are more common and independent of external source. It would need the controlled devices and implementation of software for microcontroller controlled system because the hardware devices cannot do any desired task to execute in a real time system.

Digital clocks that run on mains electricity and have no battery must be reset every time the power is cut off or if they are moved. Even if power is cut off for a second, most clocks will still have to be reset. This is a particular problem with alarm clock that have no battery backup, because even a very brief power outage during the night usually results in the clock failing to trigger the alarm in the morning .To reduce the problem, here we have designed a digital clock that operates on household electricity and incorporate a battery backup to maintain the time during power outages and during times of disconnection from the power supply. More recently, some devices incorporate a method for automatically setting the time, such as using a broadcast radio signal from an atomic clock getting the time from an existing satellite television or computer connection, or by being set at the factory and then maintaining the time from then on with a quartz movement powered by an internal rechargeable battery.

Also RTC (Real time clock) are available now. These are Digital clocks that run on mains electricity and have no battery and therefore must be reset every time the power is cut off or if they are moved. Even if power is cut off for a second, most clocks will still have to be reset .But there are RTC in the form of integrated circuits also along with battery backup and show accurate time even after power is cut off. Examples are DS1307 which runs on I2C protocol and DS1306 uses SPI protocol.

The aim of our system is to provide accurate time during or after any condition of mains (or VDD) supply.

II. HARDWARE DESCRIPTION

In this section the hardware considerations are described

1. Circuit diagram

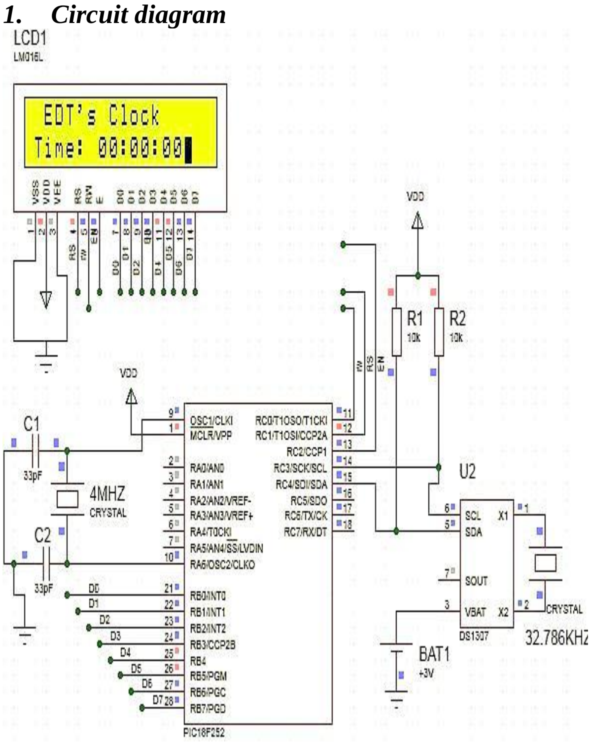

Fig. 1. Proteus design and simulation of the system

2. *Major Components*

2.1. *PIC18F252 Microcontroller*

PIC stands for Peripheral Interface Controller. PIC microcontrollers are manufactured by Microchip Technology Corporation. PIC18F252 microcontroller has a RISC architecture having 28 pins that comes with standard features.

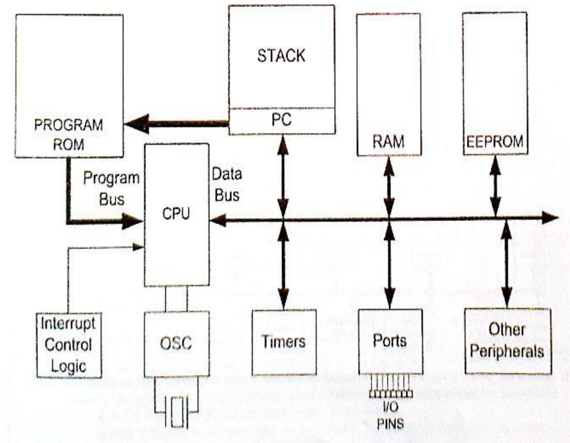

Fig. 2. Simplified view of PIC microcontroller

Some of the peripheral features are:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period registers (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as: - Capture input: capture is 16-bit, max. resolution 6.25 ns (TCY/16) - Compare is 16-bit, max. resolution 100 ns (TCY) - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz, 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module Two modes of operation: - 3-wire SPI (supports all 4 SPI modes) - I2C Master and Slave mode

2.2. *DS1307 (Real Time Clock IC)*

DS1307 is a low power serial real time clock with full binary coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM (Non Volatile Static Random Access Memory). Data and Address are transferred serially through a bidirectional I2C bus. The RTC provide year, month, date, hour, minute and second's information. The end date of months is automatically adjusted for months fewer than 31 days including leap year compensation up to year 2100. It can operate either in 24-hour format or 12-hour format with AM/PM indicator. DS1307 comes with built-in power sensing circuit which senses power failures and automatically switches to back up supply. The DS1307 RTC uses an external 32.768kHz Crystal Oscillator and it does not require any external resistors or capacitors to operate.

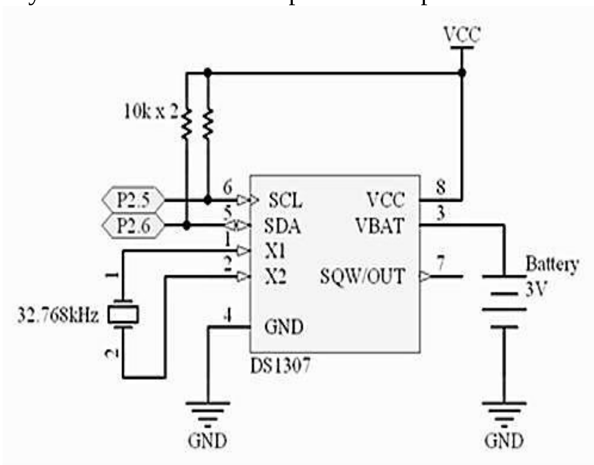

Fig. 3. DS1307 IC with primary connection

There is a list of the registers that one has to configure and set for the RTC to work. There are six registers for time keeping purpose and one register at 07h which is for configuring a square wave output derived from the clock.

The column function tells us what function each register is doing. Most of the details such a seconds, minutes, hours, months and years are given in 4-bit BCD format. Similarly the tens place of the months is never going to go beyond 1 so it just a single bit saying 0 or 1. The registers bits that are marked 0 will always be read 0.

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12 / 24	10 Hour / PM/AM	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

Fig. 4. Table keeper registers

Next comes the representation of the time, either in 12-hour format or 10-hour format. This is done by the bit 6 of the hours register. If set the clock is in 12-hours mode and if cleared it is in 24-hour mode. In 24-hour mode the bit 5 will be used along with the bit

4 to read the tens digit hours (as it can go up to 2). In the 12-hours mode the bit 5 is used as an AM or PM indicator. If the bit is set then the time is in PM and if it is cleared it is in AM mode. And as shown in fig. 4. (Table keeper registers) only bit 4 is used to display the tens digit of the hours.

2.3. *16x2 LCD*

16×2 Character LCD is a very basic LCD module which is commonly used in electronics projects and products. It contains 2 rows that can display 16 characters. Fig.5 is the pin diagram of a 16×2 Character LCD display. As in all devices it also has two inputs to give power Vcc and GND. Voltage at VEE determines the Contrast of the display. A 10K potentiometer whose fixed ends are connected to Vcc, GND and variable end is connected to VEE can be used to adjust contrast. A microcontroller needs to send two informations to operate this LCD module, Data and Commands. Data represents the ASCII value (8 bits) of the character to be displayed and Command determines the other operations of LCD such as position to be displayed. Data and Commands are send through the same data lines, which are multiplexed using the RS (Register Select) input of LCD. When it is HIGH, LCD takes it as data to be displayed and when it is LOW, LCD takes it as a command. Data Strobe is given using E (Enable) input of the LCD. When the E (Enable) is HIGH, LCD takes it as valid data or command. The input signal R/W (Read or Write) determines whether data is written to or read from the LCD. In normal cases we need only writing hence it is tied to GROUND in circuits shown in fig.5.

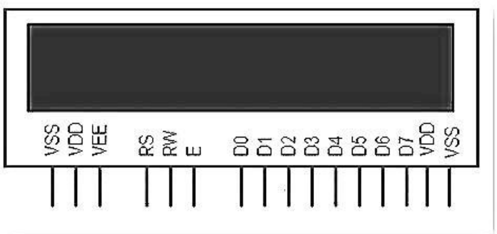

Fig. 5. 16x2 LCD

The interface between this LCD and Microcontroller can be 8 bit or 4 bit and the difference between them is in how the data or commands are send to LCD. In the 8 bit mode, 8 bit data and commands are send through the data lines DB0 – DB7 and data strobe is given through E input of the LCD. But 4 bit mode uses only 4 data lines. In this 8 bit data and commands are spited into 2 parts (4 bits each) and are sent sequentially through data lines DB4 – DB7 with its own data strobe through E input. The idea of 4 bit communication is introduced to save pins of a microcontroller. You may think that 4 bit mode will be slower than 8 bit. But the speed difference is only minimal. As LCDs are slow speed devices, the tiny speed difference between these modes is not significant. Just remember that microcontroller is operating at high speed in the range of MHz and we are viewing LCD with our eyes. Due to Persistence of Vision of our eyes we will not even feel the speed difference.

III. PRINCIPLE OF OPERATION

In order to understand the working principle of RTC system, first there is a need to understand I 2C protocol, as communication between microcontroller and DS1307 IC is done through this protocol, and then the systems working.

1. I2C protocol

I2C stands for Inter-Integrated circuit is a multimaster serial single-ended computer bus invented by the Philips semiconductor division (today NPX semiconductors) and used for attaching low speed peripherals to a motherboard, embedded system, cell phone, or other digital electronics devices

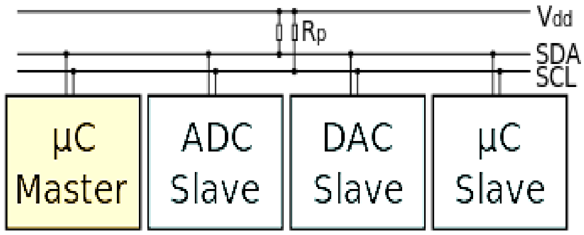

Fig. 6. One master (uc) and three slave's configuration

I2C uses only two bidirectional open drain lines, SDA-Serial Data Line (half duplexed) and SCL-Serial Clock Line, pulled up with registors. Typical voltages used are +5V or +3.3V, although systems with other voltages are also permitted. The I2C reference design has a 7 bit or a 10 bit address space. Common I2C bus speeds are 100 Kbit/s standard mode and 10Kbit/s low speed mode, but arbitrarily low clock frequencies are also allowed. There are also other features such as 16 bit addressing mode.

I2C defines basic types of messages, each of which begins with a START and ends with a STOP. They are:

- Single message where a master writes data to a slave
- Single message where a master reads data from a slave
- Combined messages, where a master issues at least two reads and/or write to one or more slaves

In RTC, micro-controller is configured as master and DS1307 as slave. The following steps are considered while writing to slave device:

- Send a start sequence
- Send the I2C address of the slave with the R/W bit low (even address)
- Send internal register number you want to write to
- Send the data byte
- [optionally, send any further data bytes]
- Send the stop sequence

Here we have DS1307 at the factory default address of 0xD0 (in hex). To start the writing operation we would write 0x01 to the seconds register at 0x00 as follows:

- Send the start sequence
- Send 0xD0 (I2C address of the DS1307 with the R/W bit low (even address))
- Send 0x00(internal address of seconds register)
- Send 0x01 (01 data as second)
- Send the stop sequence

To read seconds from DS1307 the following steps are followed:

- Send a start sequence
- Send 0xD0 (I2C address of the DS1307 with the R/W bit low (even address))
- Send 0x00(internal address of seconds register)
- Send the start sequence again (repeated start)
- Send 0xD1 (I2C address of the DS1307 with the R/W bit high (odd address))
- Read data byte from DS1307
- Send the stop sequence

2. Working principle

The circuit diagram of the digital clock contains different components in which Port B of the controller (PIC18F252) is used as the data lines for the LCD (starting from pin 21- pin 28). Here 16 x 2 lines LCD display is used. In the first line, it'll display "EDT's Clock", in the second line, LCD will be displaying "TIME : : : " with am/pm. Port C is used for the clock setting and it also provides the necessary control signals for the LCD. RC0, RC1, RC2 are connected to R/W, RS, EN of LCD respectively. Pins RC3 and RC4 are connected to SCL and SDA pins of DS1307IC respectively. Pin no. 1 and 10 are connected to the crystal of micro-controller (of 4Mz). The function of crystal is to give pulses to the microcontroller which allows the microcontroller to do various calculations. Pin no.1 of the microcontroller is connected to +5v. This causes the microcontroller to reset at Power ON and to begin execution of program from the first line. In order to limits and stabilize the voltage to +5v to the circuit , voltage regulator IC 7805 can be used. Even if the input voltage goes high up to 14 volts, this IC gives only +5v to the circuit thus preventing any damage.

IV. PROGRAMMING DETALIS

Program of RTC is written in C language. For this MPTLAB software and C-18 compiler is used. For simulation purpose PROTEUS software is used.

1. Flowchart

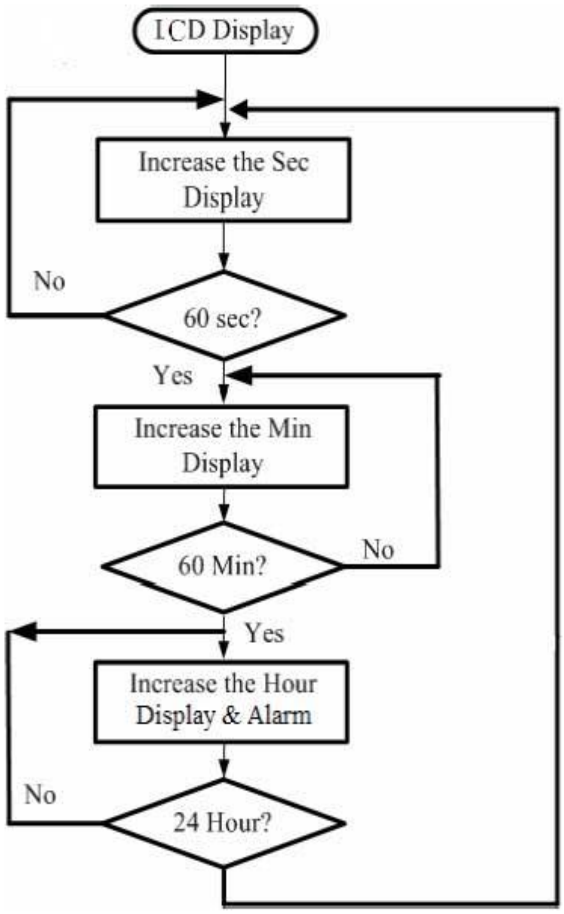

Fig. 7

2. Programming in C

```
#include <p18f252.h>
#include “msdelay.h”
#include "lcd.h"
#include “rtc.h”
#include “clock.h”
#include “lcd.h”

#pragma config OSC=HS, FCMEN=ON,
WDT=OFF, IESO=OFF, XINST=OFF, LVP=OFF
void disp_frame();
void main()
    {
      SSPADD=9;              //set i2c clock
      SSPCON1=0b00101000;  //set  i2c master
      SSPSTATbits.SMP=1;
      ADCON1 = 0x0F;
      LCD_init();
```

```
    msdelay(1000);           //delay
    disp_frame();
     reset_time();              // reset the time-keeping register    to avoid  collusion
    set_time(0x00,0x45);      // seconds
    set_time(0x01,0x57);      // Minutes
     set_time(0x02,0x66);       // Hours along with 12hour and        AM/PM selection bit
    while(1)
     {
        display();               // Repeatedly call the display function
        msdelay(1000);
      }
    }
void disp_frame()          // To display the basic frame that does not change.
  {
      LCD_cmd(LCD_LINE1);
      LCD_string(" EDT's Clock   ");
      msdelay(1000);
      LCD_cmd(LCD_LINE2);
      LCD_string("Time:  :  :     ");
      msdelay(1000);
  }
```

The above program is just the main program. All the subprogram (such as rtc.h, lcd.h, clock.h, i2c.h (in order to construct rtc.h), msdelay.h) are needed to be defined and constructed as per requirement.

The following bits are needed to be configured while constructing your own i2c subprogram:

- PIR1bits.SSPIF
- SSPCON2bits.SEN
- SSPCON2bits.ACKSTAT
- SSPCON2bits.ACKDT

V. APPLICATIONS/ FUTURE SCOPE

In a portable system such as a data collection terminal, smart card reader, which required the real time clock to keep track of the day and time of certain task being taking place. After the tasks had completed, usually most of the portable system will return to a standby mode to conserve power. The dual alarm can be set to wake up the system at certain time interval to perform other tasks. For example, in an access control application, when a person try to access the building through certain doors, the day and time of the entry will get recorded and this information can be used for accounting, security purposes, and etc. A typical computer system includes a processor subsystem of one or more microprocessor, a memory subsystem, one or more chipsets provided to support different types of host processors for different platforms such as desktops, personal computers (PC), servers, workstations and mobile platforms, and to provide an interface with a plurality of input/output (I/O) devices. Chipsets may integrate a large amount of I/O bus interface circuitry and other circuitry onto only a few chips. These chipsets may implement the I/O bus interface circuitry, timer, real-time clock (RTC), direct memory access (DMA) controller, and other additional functionality such as integrated power and thermal management with quick resume capabilities and random seed number generation for security applications such as cryptography, digital signatures, and protected communication protocols. Clock circuits generate a regular series of pulses based on a piezoelectric crystal, which governs the frequency of the pulses. The clock signal that is generated is used to synchronize the operation of the other components and circuits in the system. The boot up operation of a computer system is reliant upon the establishment of a reliable system clock. In these cases RTC, designed as explained above, can be implemented.

REFERENCES

[1] Muhammad Ali *Mazidi, Rolin D. Mckinlay, Danny Causey, "PIC Microcontroller and Embedded System*," Pearson Education, INC., Pearson Prentice Hall.

[2] Vicent Himpe, "*Mastering the I2C Bus: LabWorX 1".*

[3] Martin P. Bates, "*Programming 8-bit PIC Microcontrollers in C*", second edition

[4] Basics of I2C from "www.embedded.com"

[5] "*Real Time Clock on the MSP430*"-Taxas Instrument, published on Jan 2001.

[6] "*DS1307 64x8, Serial, I2C Real-Time Clock*"-Maxim Integrated, Rio Robles, San Jose, CA 95134 USA, published in year 2008.

[7] "*Real Time Clock Handbook*"- National Semiconductor Corporation, 1993