EE389 Electronic Design Lab-II, EE Dept, lIT-Bombay, November 2006

DIGITAL ORGANIZER WITH AUDIO MEMO

Group D12

Ali Imran Razvi (03d0702l) <imran@ee.iitb.ac.in>
Manasa Mandava (03d07036) <manasa@ee.iitb.ac.in>
Nandini Dikshit (03d07024) <nandini@ee.iitb.ac.in>

Supervisors : P.C.Pandey, V.K.Tandon, L.R.Subramanyan

ABSTRACT

This project involved developing an electronic organizer equipped with multiple functionalities such as an alarm clock, a calendar, a phonebook and a list of reminders. This organizer provides the option of recording and playing back audio reminders. This is a much more convenient option for a user than typing out and reading text messages.

1. INTRODUCTION

This project involved making a battery-operated organizer equipped with an alarm clock, calendar, phonebook and text and audio reminders. The audio memo facility distinguishes this organizer from existing ones.

The organizer incorporates a recurring and a daily alarm whose priorities can be set. It also provides two types of reminders. It allows the user to enter a piece of text that will be displayed at the set time. It also provides the option of storing an audio reminder which is saved on the Secure Digital (SD) card and then played back at the appropriate time.

The Nokia 6100 colour LCD monitor provides an interactive GUI. An SD card is used to store audio and text data in FAT format.

The phonebook contains information regarding a person's name, residence and mobile phone numbers. The calendar displays all the days of that particular month and highlights the date. It also displays the year and week number. When not in use, the organizer displays the time in the 'clock-dial' format.

2. BLOCK DIAGRAM

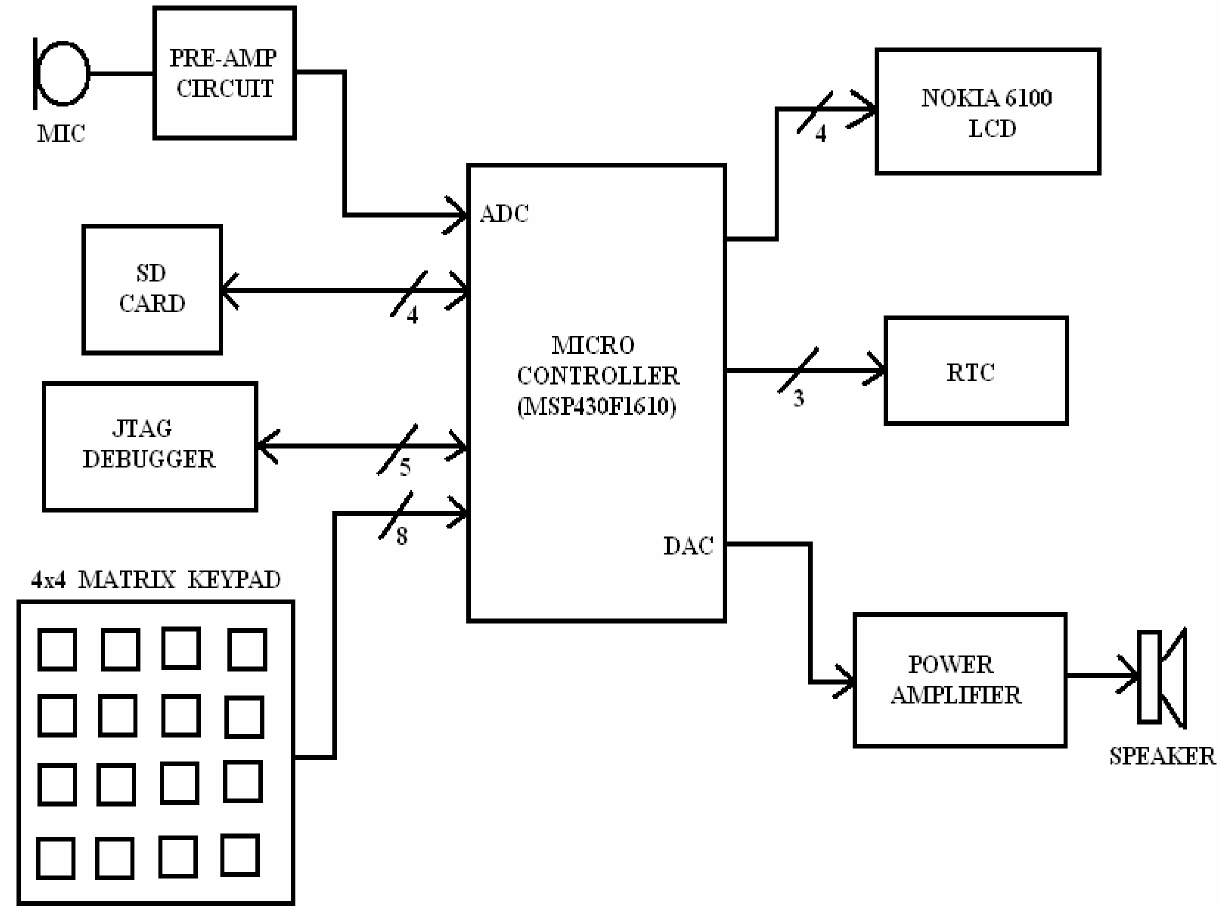

Fig.1: Block diagram of the organizer

3. HARDWARE DESIGN

3.1. Microcontroller

We have used a Mixed Signal Processor MSP430F1610 of the MSP430 family based on 16-bit RISC architecture by Texas Instruments. This processor has the following features:

- 6 I/O ports with interrupt capability on two of them.
- In-built 8-channel 2-bit ADC (Analog to Digital Converter).
- In-built 2-channel 12-bit DAC (Digital to Analog Converter).
- Serial Communication Interface (SPI).
- 5 kB RAM and 32 kB ROM.
- 3-channel internal DMA (Direct Memory Access).

This project involved interfacing a large number of peripherals such as a keypad, LCD, SD card. Hence, a large number of I/O ports were required. The interrupt capability alleviates the need for constant polling.

The ADC is used to sample the incoming audio sample (reminder) and store the samples on the SD card. When the audio memo is to be played back, these samples are given to the ADC which then outputs the reconstructed audio message.

The SD card is serially interfaced with the microcontroller using the SPI feature.

The MSP is programmed in embedded C++. The IAR workbench is used for compilation and emulation.

3.2. Keypad

The organizer uses a 4x4 keypad with each key representation multiple alphanumeric options. The arrangement of numbers and characters on the keypad is similar to that of a standard Nokia keypad. Port 2 of the MSP has interrupt capability and hence the keypad is interfaced to port 2 and port 3.

When a key is pressed, an interrupt is generated and hence, at other times, the microcontroller can go into a power saving, idle mode. This eliminates the need for constantly polling the keys of the keypad for a key press. This saves processing power and time. The keypad interfacing is as shown below:

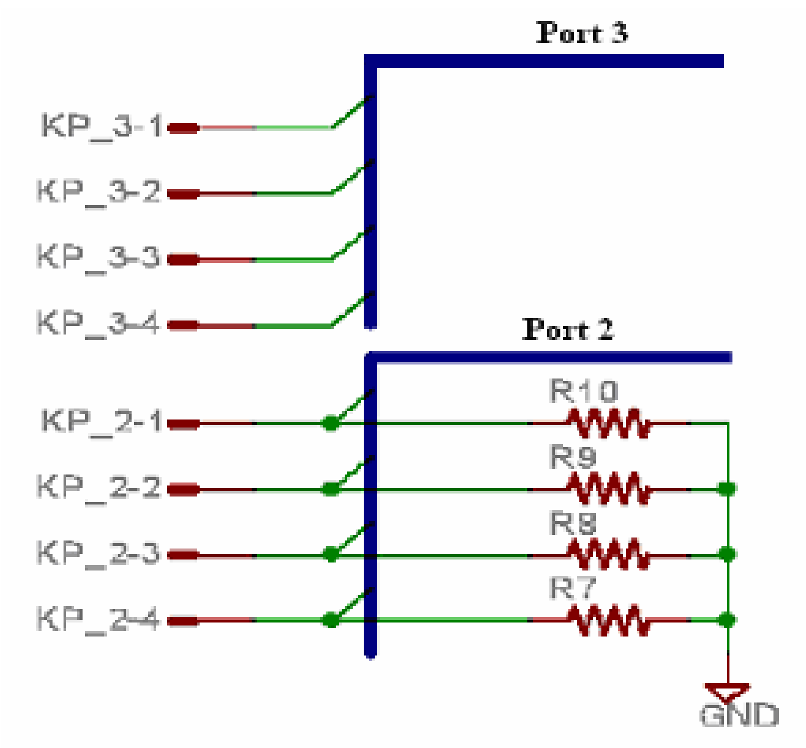

Fig.2: Keypad interfacing circuit

3.3. Audio Sampling

According to Nyquists's sampling theorem, an analog signal must be sampled at atleast twice the maximum frequency in the signal, for perfect reconstruction to be possible. For telephone quality speech, the frequency range 300 Hz to 3 kHz is found to be sufficient. We use a sampling frequency of 16 kHz for better clarity of the voice played back.

The pre-amplifier uses Texas Instrument's TLV2252, a low-voltage and low-power dual amplifier. The R6-C3 combination provides a low-pass first order RC filter with a cut-off of 2.7 kHz. Being a first-order filter, its roll-off is very gentle. The output is super-imposed on 1.2 V DC to make it compatible with the ADC input range.

For better quality of the reconstructed signal, we use the entire 12-bit resolution provided by the ADC.

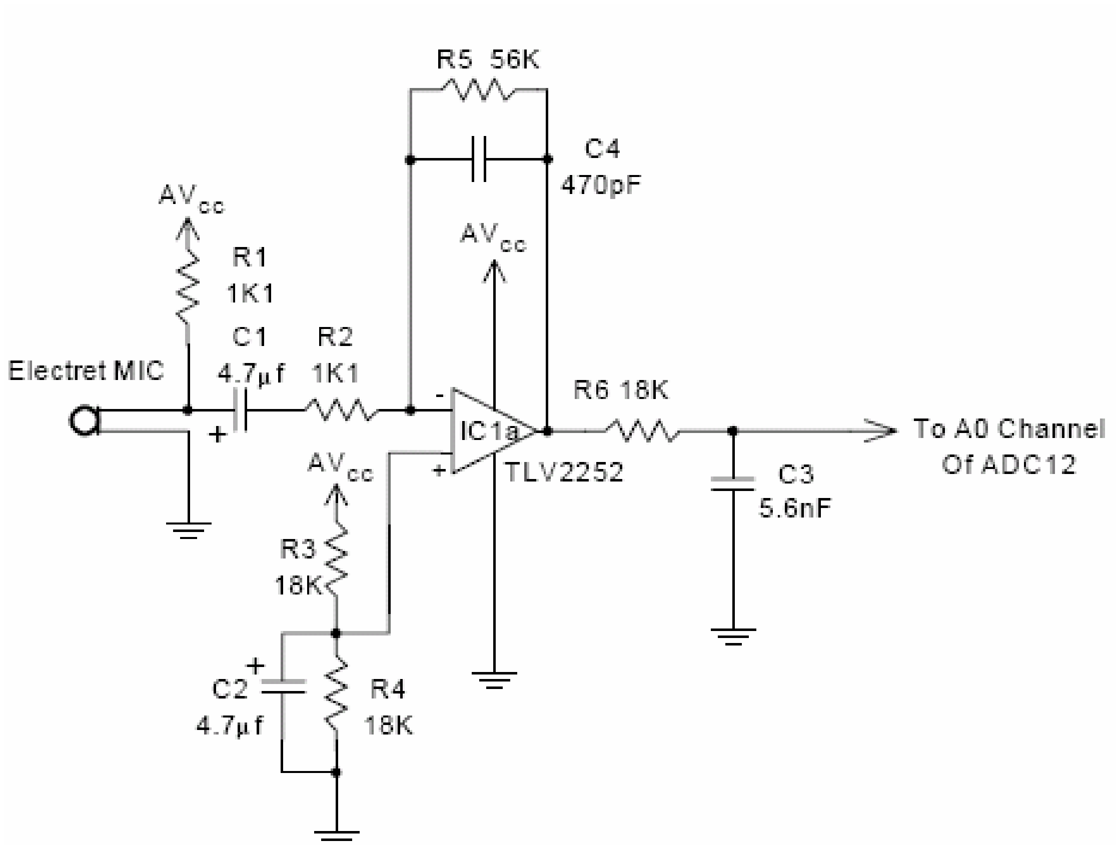

Fig.3: Microphone pre-amplifier and filter

3.4. Nokia 6100 LCD

A Nokia 6100 colour LCD, driven by an Epson S1D15G10 controller has been used. The distinguishing features of this display are as given below:

- High resolution of 128 x 128 pixels (as compared to the 84 x 48 pixel resolution of the Nokia 3310 LCD used in the first stage).
- 9 line display (as compared to the 6-line 3310 LCD). A 9-line display was required to display the calendar.
- Operates at the same voltage as the MSP i.e.: 3.3 V. Only the backlight requires a 6.5 V supply for which a separate regulator is required.
- Powerful backlight.

These were the chief features contributing to the choice of display for the organizer.

The use of a colour LCD improves the aesthetic appeal, which is important for a product like an organizer. The use of a high resolution, colour LCD will enable future improvement in the graphical interface.

The LCD has an external connector of 0.5mm pitch that needs to be soldered separately onto a PCB of appropriate pitch. Since the SPI port of the MSP is used by the SD card, an SPI port is simulated on port 5 of the MSP to drive the LCD.

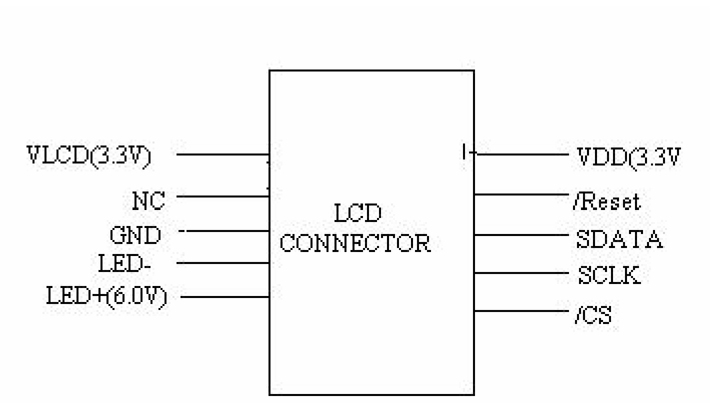

Fig.4: Nokia 6100 LCD pin diagram

3.5. RTC

DS1302 is used as a real time clock (RTC) chip. It needs a 32.768 kHz crystal for its time-keeping operation. It has a battery backup functionality and requires three pins to interface with the microcontroller for serial data transfer. The timing diagram and internal registers of the RTC are given in Appendix B.

3.6. Power Amplifier

The speech output is taken from the DAC0 terminal of the MSP. Since the MSP cannot drive large loads like the speaker, we use a TDA8551 power amplifier by Texas Instruments. It is a Bridge Tied Load amplifier of 1 W rating. It also provides a volume control which is a digital controlled attenuator between the audio input pin and the power amplifier.

The TTL tristate buffer 74LS125 simulates the action of a switch in software. It also converts the 3 V logic of the microcontroller to the 5 V logic required by the audio amplifier.

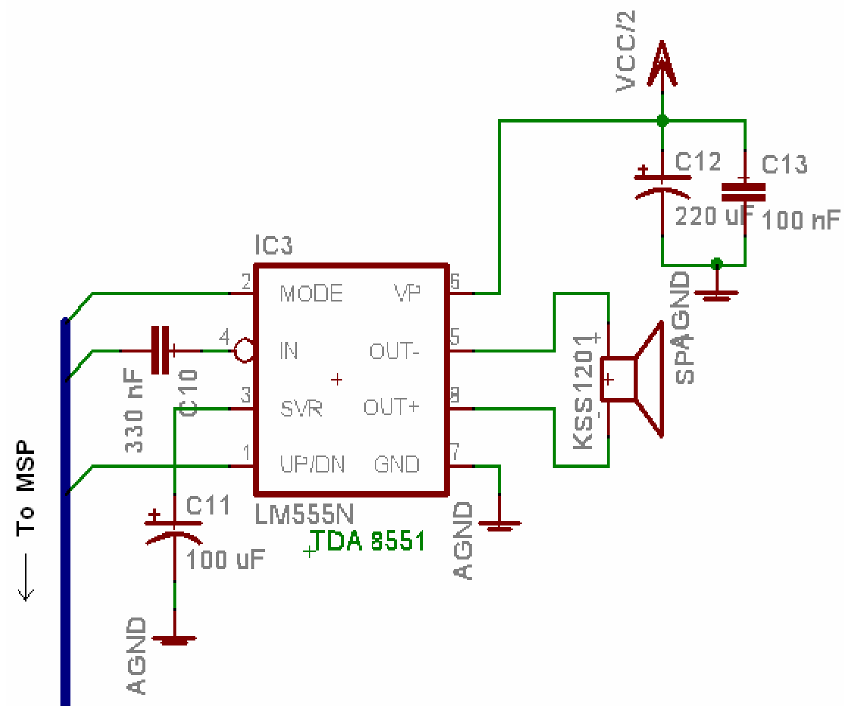

Fig.5: Power amplifier circuit

3.7. Power Supply

The entire circuit operates at two different voltage levels – 3.3 V and 5 V. The analog circuits namely the pre-amplifier and power amplifier operate at 5 V while the MSP, SD card and LCD operate at 3.3 V. The backlight of the LCD requires a separate 6.5 V supply.

Thus the circuit contains three regulators:

- LM7805 – for 5 V supply
- LM317 – for 3.3 V supply
- LM317 – for 6.5 V supply

Thus the entire board can be driven by a 9 V battery with a current requirement of 120mA.

3.8. Secure Digital (SD) Flash Card

SD-card is flash based memory card specifically designed for portable applications like cellular devices. It works on 3.3 V and has SPI interface facility. MSP430 has an inherent support for SPI interface. Hence, the communication between the SD-card and the MSP430 was set up using an SPI interface and the information is transferred byte wise. The data transfers between the SD card and the MSP430 are managed by DMA channels. For a byte transferred from the MSP430 through MISO, a byte is written back from the SD-card through MOSI. Hence two DMA channels are utilized for this purpose.

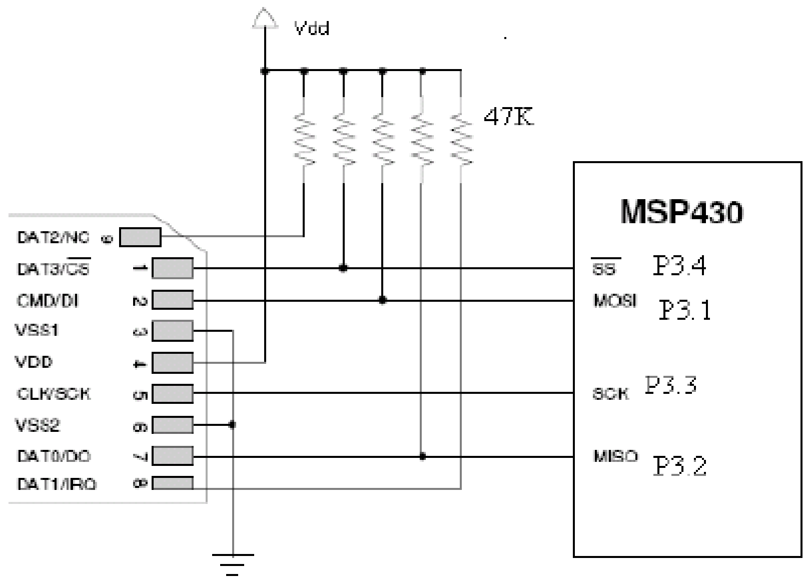

Fig.6: SD card interfacing

4. SOFTWARE DESIGN

4.1. Initialization

All the ports are initialized as input or output ports as required. Peripherals like the LCD and SD card are initialized as well. The interrupt pins are configured. The MSP then goes into a low power mode until an interrupt appears.

4.2. Keypad

An interrupt is generated whenever a key is pressed. The MSP then exits the low power mode it was in. With an alphanumeric keypad, as in a standard Nokia display, the time interval between two key presses determines if the same character should appear twice or the second character on the same key should be displayed. The difference between two valid key presses is calculated using timer A0 of the MSP. The threshold value for the same key to be pressed is set at 0.5 s.

The state diagram governing the determination of which character to display is as given below:

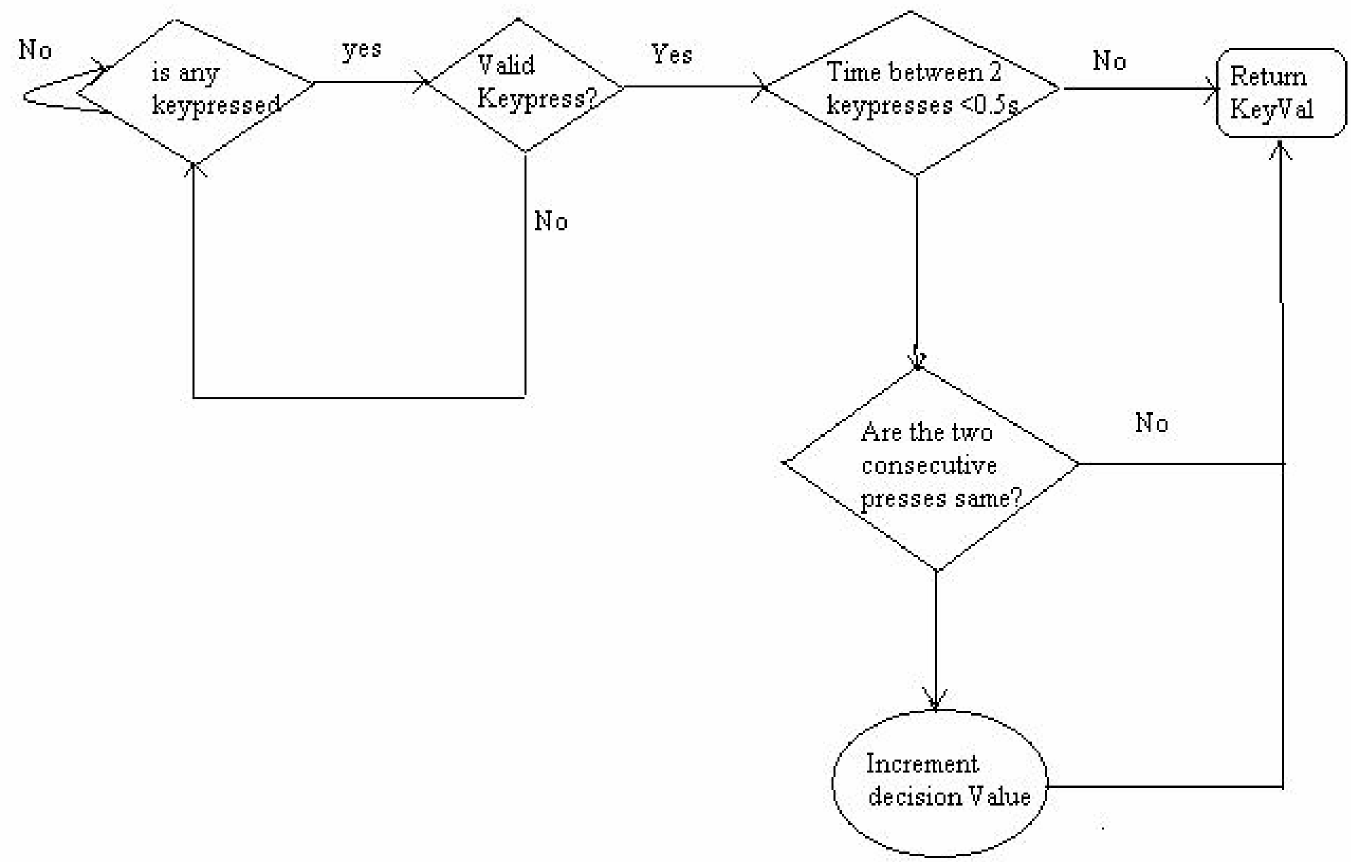

Fig.7: State machine for keypress

The value of the key is determined using the lookup table and the value of the decision value. For example, the value of the decision value for the letter ‘B’ would be 2. Care is taken to return the appropriate value even if number of times a key is consecutively pressed is greater than number of characters assigned to the key.

4.3. Digital Clock

TimerA of the MSP430 is has a 32.768 kHz crystal as its source. This timer generates one second interrupts which updates the clock. The RTC is for backing up the time. When the MSP430 is initialized the time is loaded from the RTC. After that the clock is locally maintained. Both the MSP430 and the RTC individually keep time. This is done to prevent losing the current time when power goes off. The RTC has a separate battery as its source. Therefore it continues to keep track of time even when the MSP loses power.

When the power comes back the time can again be loaded from the RTC. While changing the time the registers of the RTC are also set with the new values. This is done to synchronize the time in the MSP and the RTC.

4.4. Calendar Display

When the calendar is displayed, the top line of the display includes the month, year and the current week of the year. This is followed by the entire month's calendar with the present date highlighted. By pressing the UP and DOWN keys we can change the month whose calendar we want to view.One can traverse the calendar using UP, DOWN, LEFT, RIGHT.

4.5. Nokia LCD Display

The routines written for the LCD include the following:

1 *LcdInit()* to initialize the display.
2 The *LcdLittleChr(), LcdLittleBoldChr(), LcdBigChr(), LcdBigBoldChr()* to generate one of two font sizes in normal or bold font.
3 *DrawLine()* To generate straight lines between any two co-ordinates.
4 *DrawCircle()* To generate a circle with the desired radius and centre.

When idle, the organizer displays the time in the 'clock-dial' format. The *update_graphical_display()* function is called every second and the display refreshes to display the second, minute and hour hands in their new positions.

4.6. Phonebook

The phonebook contains information about the name, home number, office number, fax number of each individual. The main menu provides the following options:

- Add a new contact
- Edit existing contact information
- Delete an existing contact
- Search for a particular contact

A doubly-linked list is implemented to store each entry of the phonebook. A *head_ptr* marks the beginning of the list. The list is sorted on the basis of the contact's name. Each new entry that is to be added, is inserted in the appropriate position using a temporary pointer, *temp_ptr* to compare the new name with that of all the existing contacts.

Deletion of a contact or modification of its details, involves 'searching' for that particular contact before any action can be taken. The *temp_ptr* is again used to search for the particular contact.

Thus the routines written involve *search(), add()* routines. A *str_cmp()* routine compares the name of the new contact or the contact to be located and the name of each contact in the linked list.

4.7. Alarm Clock

The organizer provides two types of alarms – a daily alarm and a one time alarm. When the one time alarm is set, it goes off at the specified time and disables itself. However in the case of the daily alarm, once it is set it will continue to go off at the specified time every day until it is manually disabled.

Once the time of the alarm has been set, the user can also choose the audio file which has to be played.

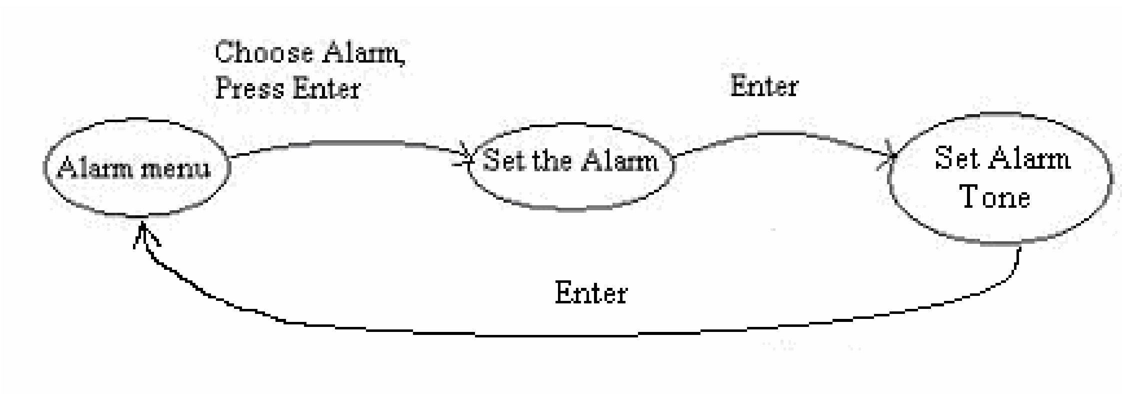

Fig.8: State diagram for editing the alarms

.

4.8. Reminder

The Reminder is a one time alarm in which we have to set both the time and date of the alarm. As the name suggest it is an alarm to remind us of a particular event. Therefore we can save a text message which will be displayed at the specified date and time. We can also optionally add an audio memo which can be played at the specified date and time.

4.9. Audio Recording

Audio recording is done by sampling the audio signal by ADC of the MSP. The samples are then stored in a local buffer. When the buffer is filled we empty the buffer into the SD card and start writing the samples at the beginning of the buffer. We stop the recording when the key corresponding to end of recording is pressed.

4.10. SD-card

The SD protocol used is the SPI (Serial Peripheral Interface) mode of the SD card protocol. This SD protocol is a simple command-response protocol. All commands are initiated by the MSP430 (Master).The SD card responds to the command with a response frame and then, depending on the command, may be followed by a data token to indicate either the beginning of the data transfer or an error condition. The highest order bit in the SD card response is always zero.

The SD card controller is implemented using functional calls initialization, block read, block write and set of status functions.

4.11. FAT file system

FAT file system implementation was necessary on the SD card. This need arises due to the requirement of dynamic storage and removal of various different types of files. For the implementation of the FAT file system we have written many functions which will allow us to write, delete and close files. These functions are each called by with the filename as input. The functions written for handling files are:

- *FAT_init()* for initializing the FAT parameters.
- *FAT_ListDir()* lists all the files in the present working directory. It returns the number of files in the directory.
- *FAT_DeleteFile()* is used to delete files.
- *FAT_ReadFile()* is used to read various types of files
- *FAT_loadFileWithName()* is used to check whether the current file exist or not.
- *CreateFile()* is used to create a file. Before writing a file we have to create a file and have to assign one cluster to it. It initializes the file size to zero
- *CloseFile()* closes the open file and sets the FAT parameter like the file size.
- *FAT_CD_Root()* is used to change the working directory to the root directory
- *Chdir()* fist check whether the directory exists or not and then changes the present working directory to the new directory.

More information about the FAT file system is in Appendix C.

5. CIRCUIT DESIGN SPECIFICATIONS

The important components used in the circuit include:

- MSP430F1610 Mixed Signal Processor
- 128 MB Secure Digital Card
- 4x4 Matrix Keypad
- Nokia 6100 LCD display
- DS1302 Serially interfaced RTC
- TLV2252 low-voltage, low- power dual OPAMP
- TDA8551 power amplifier
- LM7805: 5 V voltage regulator
- LM317: 3.3 V voltage regulator
- LM317: 6.5 V voltage regulator
- 8 MHz crystal
- 32.768 kHz crystal for RTC

6. RESULTS AND FUTURE SCOPE

During the course of the project, we have successfully interfaced the Nokia 6100 colour LCD and a SD card and implemented the FAT16 file system. We have also successfully tested the recording and playback of audio data.

The organizer can be further improved by incorporating a calculator as well as a radio to enhance its feature set. We can also access and set various reminders to be activated on a particular day as chosen through the calendar, instead of having to explicitly set these reminders.

The clock could be interfaced with a personal computer to download daily schedules of an individual, so that the clock acts accordingly. The phonebook could also be updated through the computer.

7. ACKNOWLGEMENTS

We would like to thank Dr. P.C.Pandey, Dr. L.R.Subramanyam and Mr.V.K.Tandon for their able guidance and support. We would also like to thank our seniors especially Aneesh Nainani and Sandhya Hegde for sparing their valuable time to help us with MSP and FAT file system. We would also like to thank the lab staff for their co-operation during and outside lab hours.

APPENDIX – A

JTAG programming tool

The schematic of the JTAG programmer is given in Fig.9.

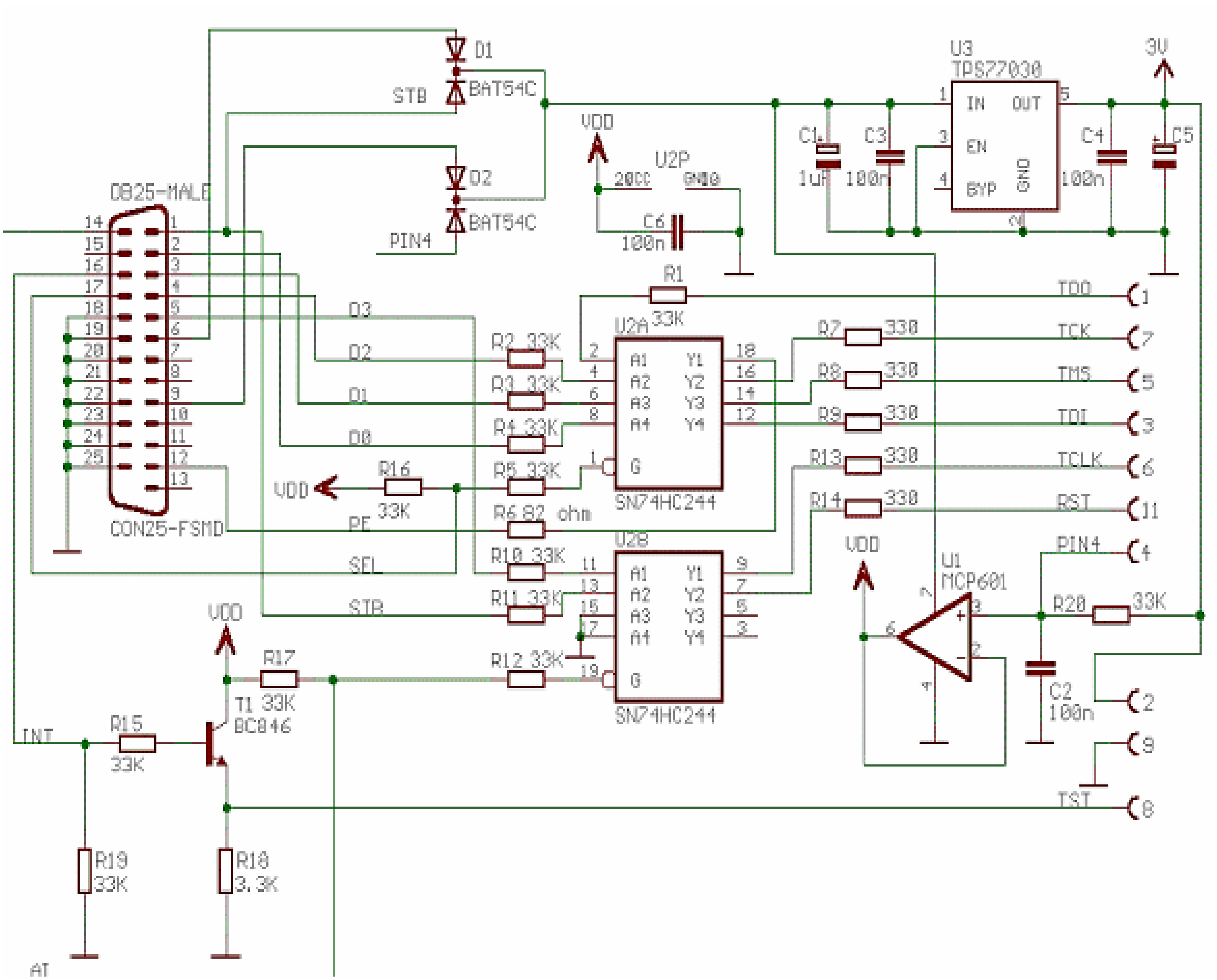

Fig. 9: JTAG Programmer

Circuit Schematic

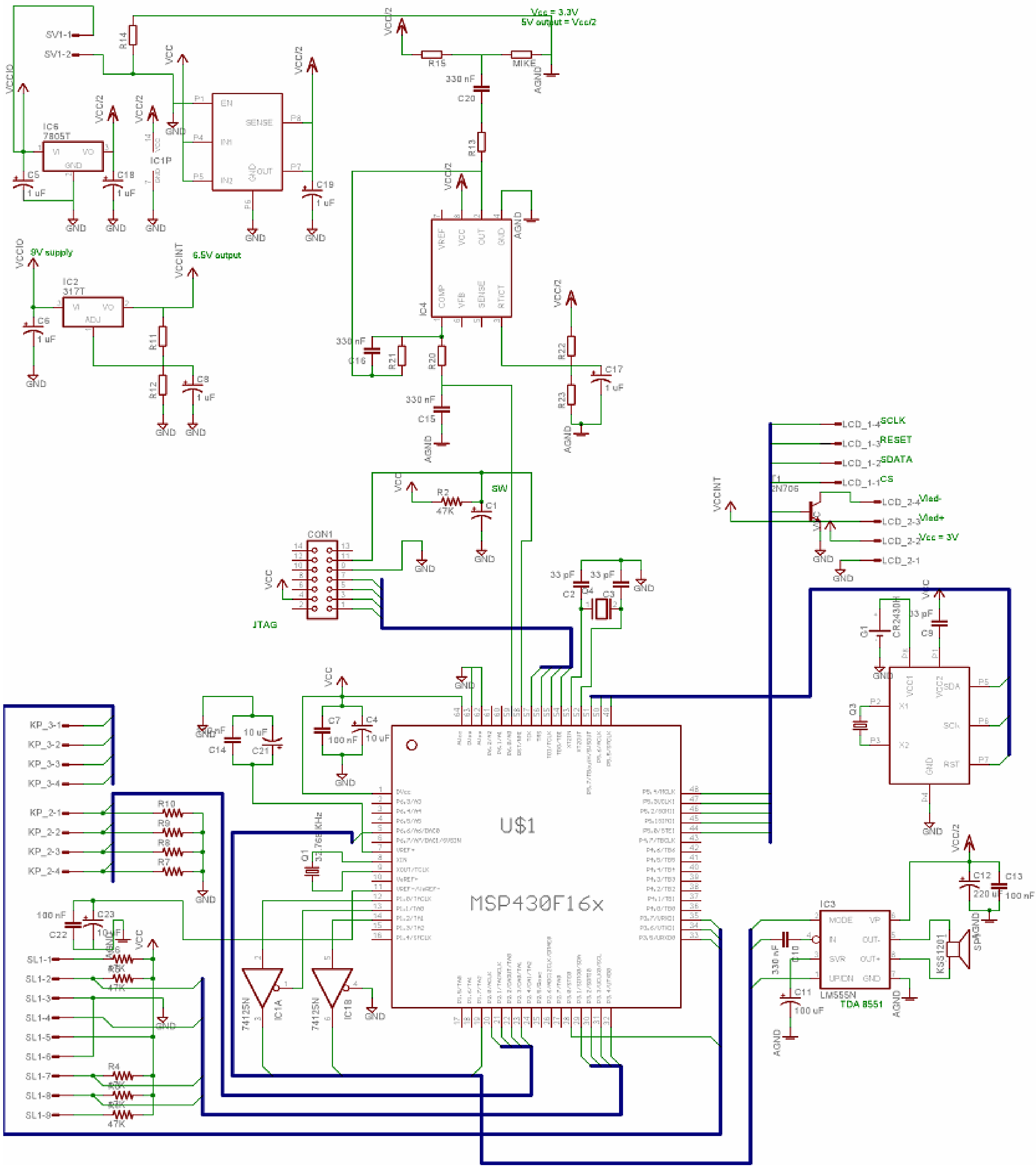

Fig.10: Circuit Schematic

PCB Layouts

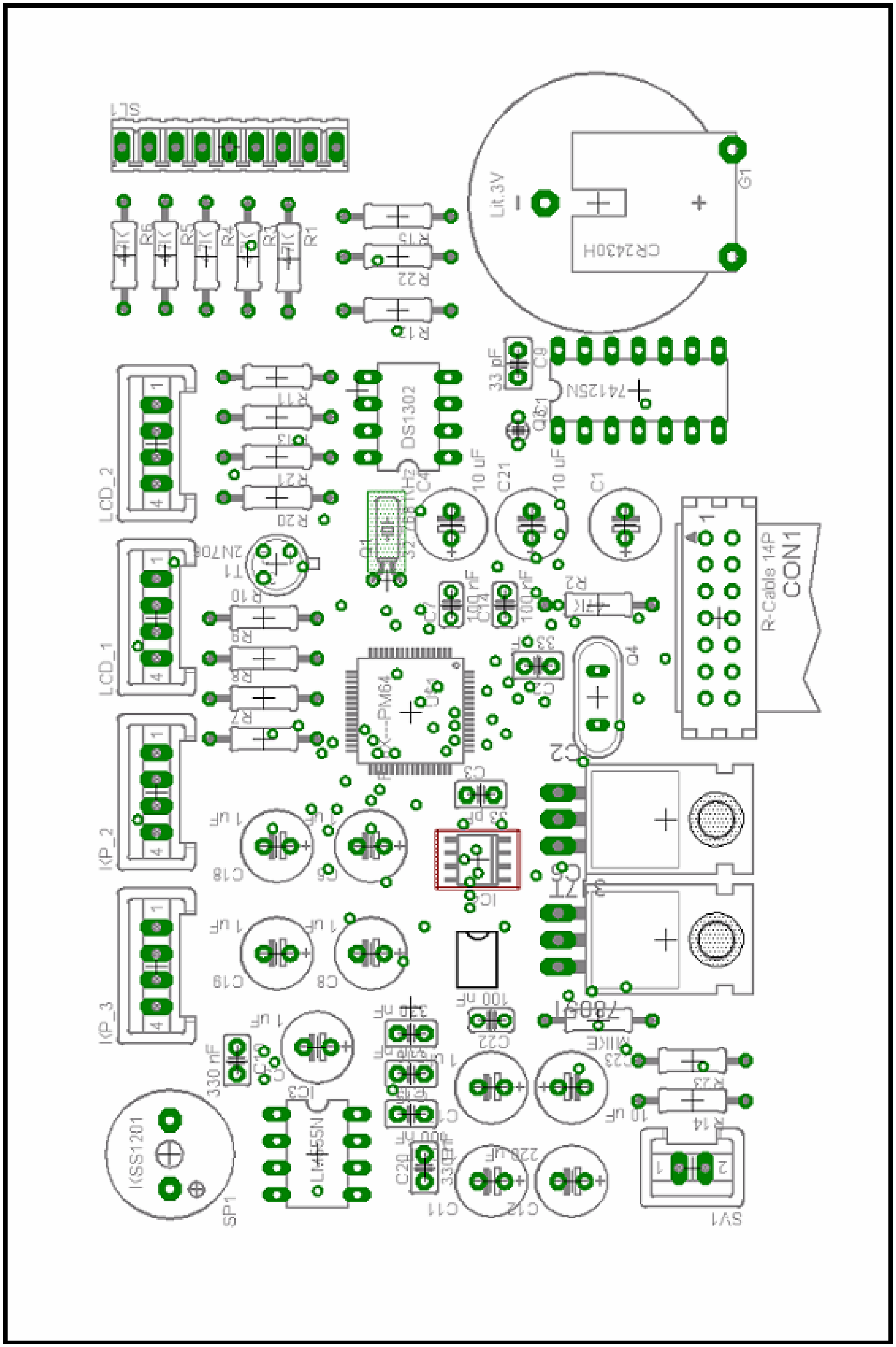

Fig.11: Component Layout

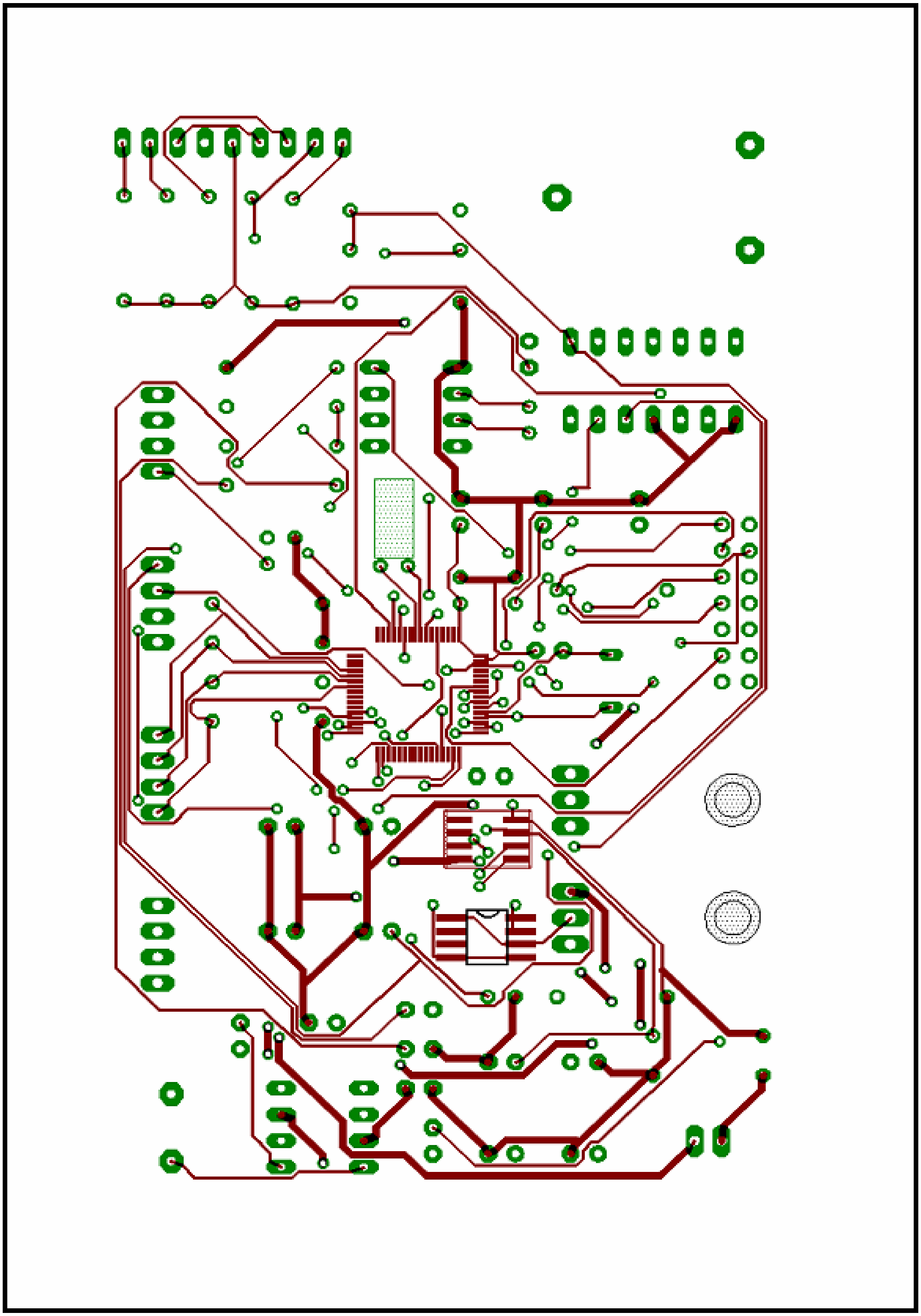

Fig.12: Top layer

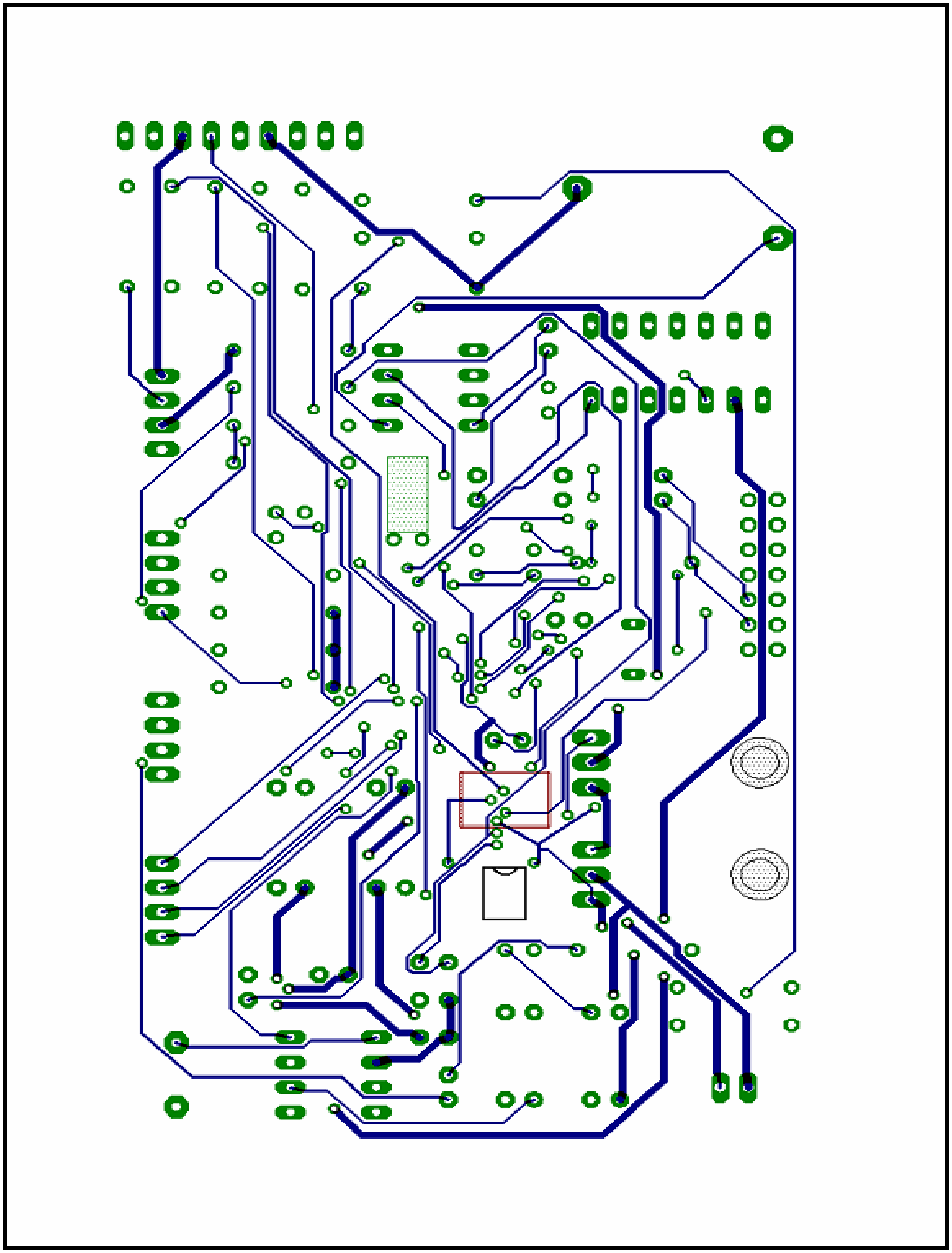

Fig.13: Bottom Layer

Appendix – B

DS1302 Real Time Clock

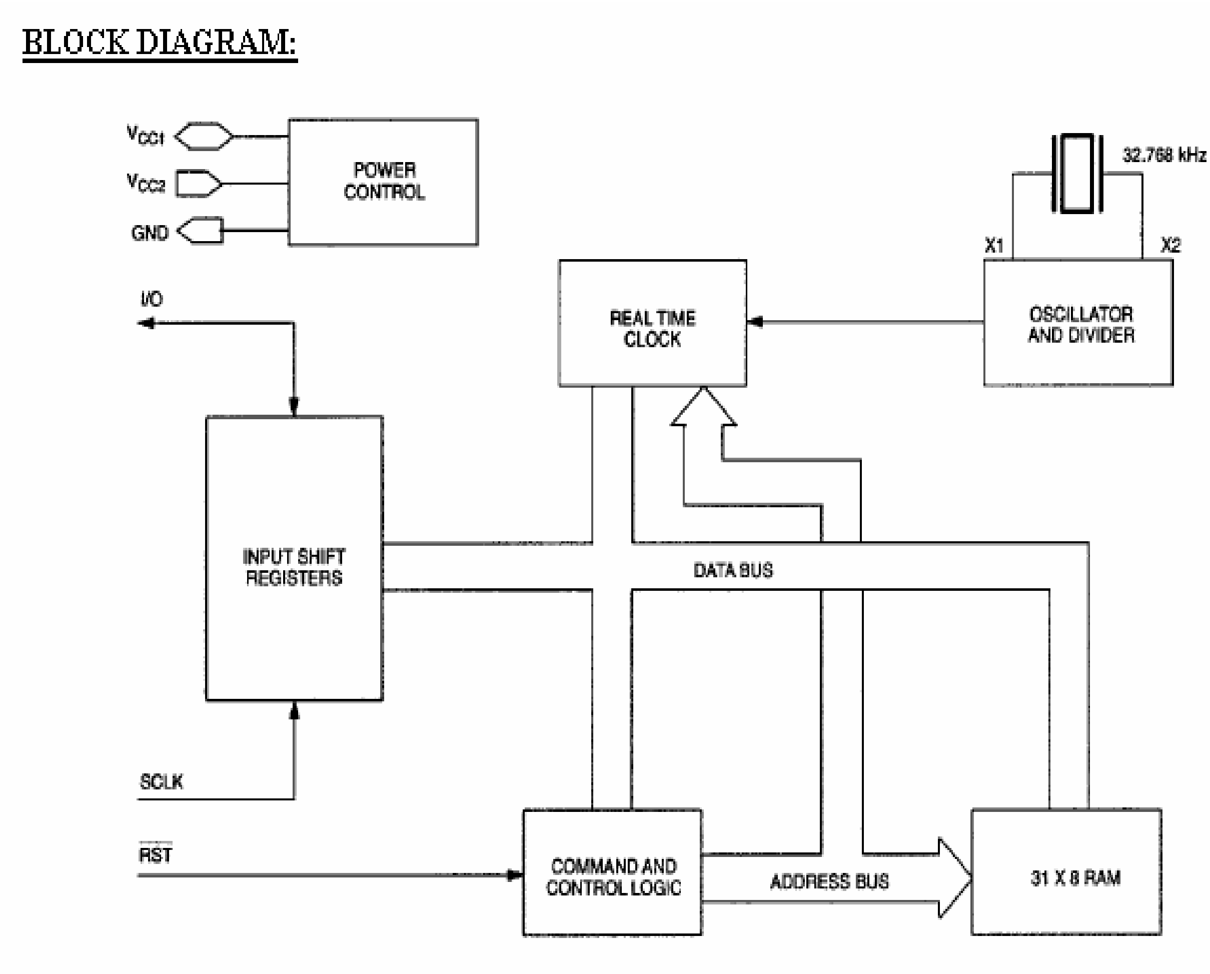

Fig.14: Internal Block Diagram of RTC

SIGNAL DESCRIPTIONS

VCC1 – VCC1 provides low-power operation in single supply and battery-operated systems as well as low power battery backup. In systems using the trickle charger, the rechargeable energy source is connected to this pin. UL recognized to ensure against reverse charging current when used in conjunction with a lithium battery.

VCC2 – VCC2 is the primary power supply pin in a dual-supply configuration. VCC1 is connected to a backup source to maintain the time and date in the absence of primary power. The DS1302 will operate from the larger of VCC1 or VCC2. When VCC2 is greater than VCC1 + 0.2 V, VCC2 will power the DS1302.When VCC2 is less than VCC1, VCC1 will power the DS1302.

SCLK (Serial Clock Input) – SCLK is used to synchronize data movement on the serial interface. This pin has a 40 kΩ internal pull-down resistor.

I/O (Data Input/Output) – The I/O pin is the bi-directional data pin for the 3-wire interface. This pin has a 40 kΩ internal pull-down resistor.

RST (Reset) – The reset signal must be asserted high during a read or a write. This pin has a 40 kΩ internal pull-down resistor.

X1, X2 – Connections for a standard 32.768 kHz quartz crystal. The internal oscillator is designed for operation with a crystal having a specified load capacitance of 6 pF. The DS1302 can also be driven by an external 32.768 kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

TIMING DIAGRAMS

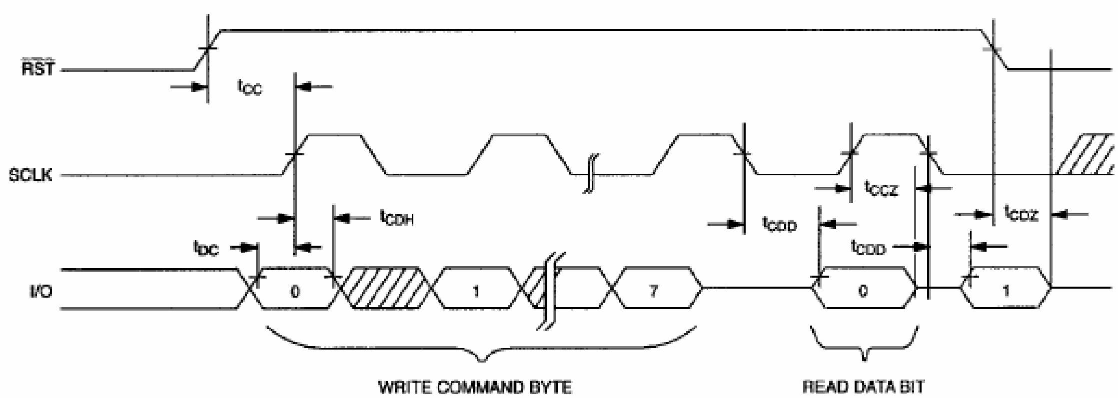

Fig.15: Timing Diagram: Read Data Transfer

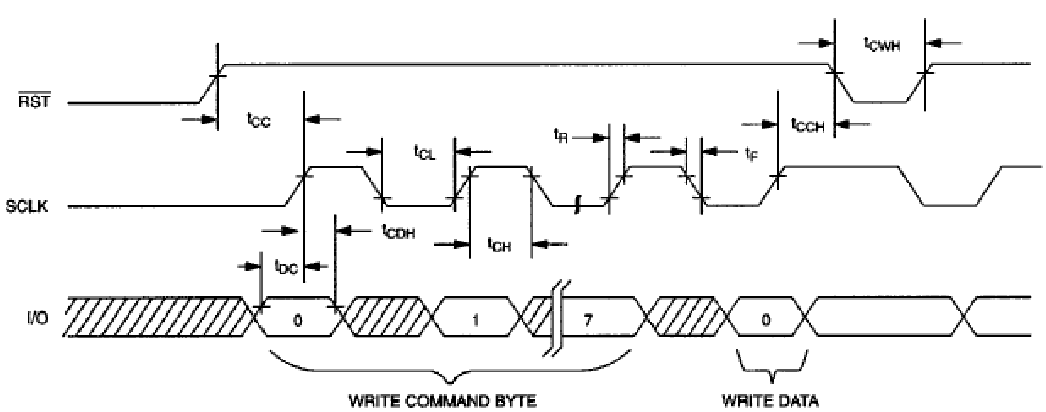

Fig.16: Timing Diagram: Write Data Transfer

Appendix – C

FILE ALLOCATION TABLE: FAT-16 File System

Boot sector	More reserved sectors (optional)	File Allocation Table #1	File Allocation Table #2	Root Directory (FAT12/16 only)	Data Region (for files and directories) … (To end of partition or disk)

Fig.17: Structure of FAT partition

Boot Sector

The first disk sector is always reserved for the boot record [5]. It contains information about the disk or its partitions. The boot record information enables the file system to handle the disk. Within boot sector there is an structure known as the BIOS Parameter Block, or "BPB." Taken as a whole, the BPB provides enough information for the executable portion of the boot sector to be able to locate/read files and directories.
Some of the Entries of the BPB which are of particular importance to our project are:

Field	Offset (from start of Boot Sector)	Length(in words)
Bytes Per Sector	11	2
Sectors Per Cluster	13	1
Reserved Sectors	14	2
FATs	16	1
No Entries In root	17	2

Table 1: Boot Sectors

Bytes Per Sector: This is the size of a hardware sector and for most disks in use in the United States, the value of this field will be 512.

Sectors Per Cluster: Because FAT is limited in the number of clusters (or "allocation units") that it can track, large volumes are supported by increasing the number of sectors per cluster. The cluster factor for a FAT volume is entirely dependent on the size of the volume.

Reserved Sectors: This represents the number of sectors preceding the start of the first FAT, including the boot sector itself. It should always have a value of at least 1.

FATs: This is the number of copies of the FAT table stored on the disk. Typically, the value of this field is 2.

Root Entries: This is the total number of file name entries that can be stored in the root directory of the volume. On a typical hard drive, the value of this field is 512. Note, however, that one entry is always used as a Volume Label, and that files with long file names will use up multiple entries per file. This means the largest number of files in the root directory is typically 511, but that you will run out of entries before that if long file names are used.

FAT areas

After the boot record, we get to the FAT areas. There are usually two identical FATs. FAT number 2 is simply a spare copy of number 1, since FAT is essential for the function of the disk.

FAT consists of a table of whole numbers, which has 65,536 16-bit entries. Each of these *entries* contains information about a cluster.

The content of each FAT entry consists of a whole *number*. In the table below, they are written as four digit hexadecimal numbers, which show one of four options.

Root directory and other directories

The last administrative area on the disk is the root directory. Since there are always 512 file or directory entrances in the root directory, the root directory is unique in its fixed size and its location in the root. Other than that, it is a directory like any other.

Actually, a directory is a list of files and other directories. Thus, you can read the names of files and sub directories in the directory. The directory structure consists of a number of directory entries.

Let us look at these directory entries, each of which occupies 32 bytes. The directory entries are identical, whether they are in the root directory or a sub directory.

Data Area

The rest of the disk contains the most important part, the data area, where all files and sub directories are stored. The data area is by far the largest part of the disk.

The sectors in the data area are allocated in clusters. As mentioned before, the maximum number of clusters for data is $2^{16} = 65{,}536$. For example consider a disk is of size 160 MB. That results in 40,400 clusters, 8 sectors each.

All sub directory entries in the data area are organized in 32 byte files, which contain the same fields as the root directory entries.

REFERENCES

- MSP430F1610 user guide, Source - **http://www.ti.com**
- MSP430F1610 datasheet, Source - **http://www.ti.com**
- MSP430F1610 sample C demos, Source - **http://www.ti.com**
- JTAG hardware emulation for MSP430, Source - **http://www.olimex.com/dev**
- Nokia LCD library, Source - **http://electronique.marcel.free.fr/LCD%20Nokia%20couleurs.html**
- SD card interfacing with MSP430
- TDA8551 datasheet- **http://www.semiconductors.philips.com/pip/TDA8551T_N1.html**
- DS1302 Real Time Clock datasheet –
- **http://datasheets.maxim-ic.com/en/ds/DS1302.pdf**
- TLV2252 OPAMP - **http://www-s.ti.com/sc/ds/tlv2252.pdf**