EE 389 Electronic Design Lab II Project Report, EE Dept, IIT Bombay, Submitted Nov 2006

Load Scheduler Programmable Over Ethernet

Group No: D05
Akshay Mehra (03D07009) akshay@ee.iitb.ac.in
Sambit Palit (03D07020) spalit@ee.iitb.ac.in
Sham Chabukswar (03D07022) sham@ee.iitb.ac.in
Supervisors: L R Subramanayan, V.K.Tandon, P C Pandey

Abstract

This project is aimed towards developing a device which can switch on and switch off electrical appliances at a predetermined time, with high configurability. The device can be connected to a 10/100 Mbps LAN and can be programmed with a remote computer connected to the same network. An RS232 interface has also been provided for its use where LAN is unavailable. Main features include a low-power MSP430 microcontroller, real time clock with a back up battery, Ethernet controller, and internal relays for switching appliances on 230V mains line.

1. **Introduction**

The objective is to develop a device which is capable of switching electrical appliances, alarms and indicators on and off at a predetermined time. This system has many applications, e.g. one may want that before he wakes up, food and hot water should be ready. All that has to be done is to connect the geyser, the toaster and the alarm to this device, and switch them the geyser and toaster on before the alarm. Set the necessary durations and make it happen with a weekly period.

The device we have made is capable of handling eight appliances at the same time independently of one another. Options include five different levels of periodicity, variable durations and a pulsed output. However appliances taking more than 1400 watts of electricity cannot be connected directly.

2. **Block Diagram**

The block diagram of this project is given in Fig: 1. The description of the different blocks are written below. Detailed circuit diagrams and descriptions are given in the next section.

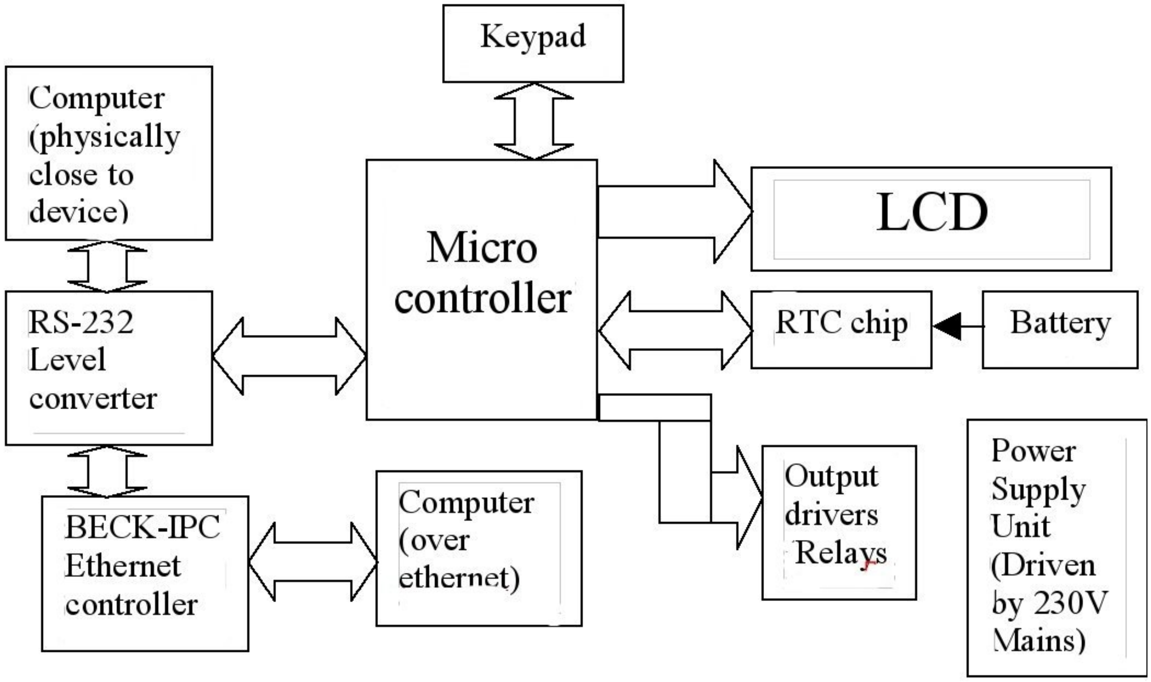

Fig. 1: Block diagram

2.1 Microcontroller

The microcontroller used in this device is the MSP430 based mixed-signal microcontroller, MSP430F1610, developed by TI [1]. The Texas Instruments MSP430 family of ultra low power microcontrollers consists of several devices featuring different sets of peripherals targeted for various applications. The architecture, combined with five low power modes is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that attribute to maximum code efficiency. The digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 6μs.

Other features include two built-in 16-bit timers, two universal serial synchronous and asynchronous communication interfaces (USART) and 48 I/O pins. In addition, this microcontroller also offers extended RAM addressing for memory-intensive applications and large C-stack requirements. Also, a JTAG interface is extremely convenient for coding the microcontroller through the computer, and for hardware emulation and debugging.

The devices connected to the microcontroller are a keypad (8 data lines), Nokia 3315 LCD display (5 data lines), UART for Ethernet controller (2 data lines), real time clock chip (3 data lines) and relays for switching electrical devices (8 data lines). In addition, 4 pins are required to be interrupt enabled, and also UART control is required. An easy hardware emulation interface (JTAG) [9], and a large internal RAM size (5 Kb) makes the MSP430F1610 ideal for this purpose, where using a 8051 based controller would be unsuitable. The microcontroller is the main processing center for all operations on the device, except the LAN based data transfer, where the work is shared by the MSP430 and the 8085 based processor on the Ethernet controller.

.

.2.2 Ethernet Controller

The Ethernet controller is the development kit DK51 provided by BECK, which belongs to the IPC@CHIP embedded web controller family [4]. This includes a x86 based processor running a DOS-based operating system. It also includes a full TCP/IP stack with a UDP/TCP socket interface, DHCP, FTP, PPP, Telnet, Web server etc. The serial interface runs at 19200 baud. Executables need to be compiled with Borland-C++ IDE [13], and uploaded on to the controller over FTP.

2.3 Nokia 3315 LCD

We have used the Nokia 3315 based LCD display for this device. It is a 84x48 pixel display, as compared to the 16x2 character LCD display, hence allowing for more configurability and hence a good and intuitive GUI. This LCD is driven by a single chip named PCD8544 manufactured by Philips Semiconductors. All necessary functions for the display are provided in a single chip, including on-chip generation of LCD supply and bias voltages, resulting in a minimum of external components and low power consumption. The PCD8544 interfaces to microcontrollers through a serial bus interface [5].

2.4 Real Time Clock (RTC)

For timekeeping when the device is switched off, as in the case of power cuts for example, we need a backup battery. The DS1302 timekeeping chip is perfect for the purpose [2]. This RTC counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100. The I/O is 3-pin serial, resulting in minimum pin count. It also runs on a wide power supply range of 2.0V to 5.5V, and hence is directly compatible with the MSP430 microcontroller, which is powered on 3.3V. Other features include low power consumption, and an expected backup battery lifetime of 10 years with the CR2430 battery. An external 32.768 KHz oscillator is needed for its operation.

2.5 Keypad

A 4x4 matrix keypad has been used for taking user input, especially for network configuration. This includes parameters like IP (Internet Protocol) address, subnet mask and gateway. These are the most important parameters because without these parameters being set on the device, it cannot be connected on LAN. The LCD is used in conjunction with the keypad. Both the LCD and the keypad have been put exclusively for this purpose.

.

2.6 RS232 Level Converter

In order to interface the UART of the microcontroller to an RS232 [8] of a PC, or the Ethernet controller, we need a level converter to convert data at RS232 levels to 3.3V digital logic levels. The MAX232 from Maxim Semiconductors is ideal for this purpose [3]. However, the output levels of MAX232 are of 5V digital logic, and hence another level converter is used after the output of MAX232, namely the SN74HC244 is used. This is manufactured by several companies, including Texas Instruments and Philips Semiconductors. This chip is able to convert 5V logic to 3.3V logic, and vice versa.

2.7 Output Control

Our device should be able to switch on and off electrical appliances, where the desired signal comes from the microcontroller. We have used 12V relays for this purpose. The appliance is connected between the common and N/O terminals of the relays, and is switched on whenever the coil is active. An npn transistor, SL100, is used as a switch to activate the coil.

2.8 Power Supply

Our device can be connected directly on 230V mains, and hence we need to use a transformer - rectifier - filter - regulator arrangement. To supply the worst-case current required when all eight relays are switched on, we had to use a 230-15V transformer with a rated current of 2A. The AC supply from the transformer is regulated and filtered by a bridge rectifier and filter circuit.

The device requires four different power supply levels. 3.3V is required for the microcontroller, RTC, LCD and 5V-3.3V logic converter (SN74HC244); 5V is required for the RS232-5V level converter (MAX232); 9V is required by the Ethernet controller; and an unregulated rectified supply is given to the coils of the relays. In order to generate the three regulated DC power voltages, the regulators LM317, LM7805, and LM7809 are used.

3. **Hardware**

The entire schematic of the circuit, made in Eagle [12] is shown in Fig. 2. In the following subsections, each of the modules is explained in detail. The PCB layout is shown in Fig. 3.

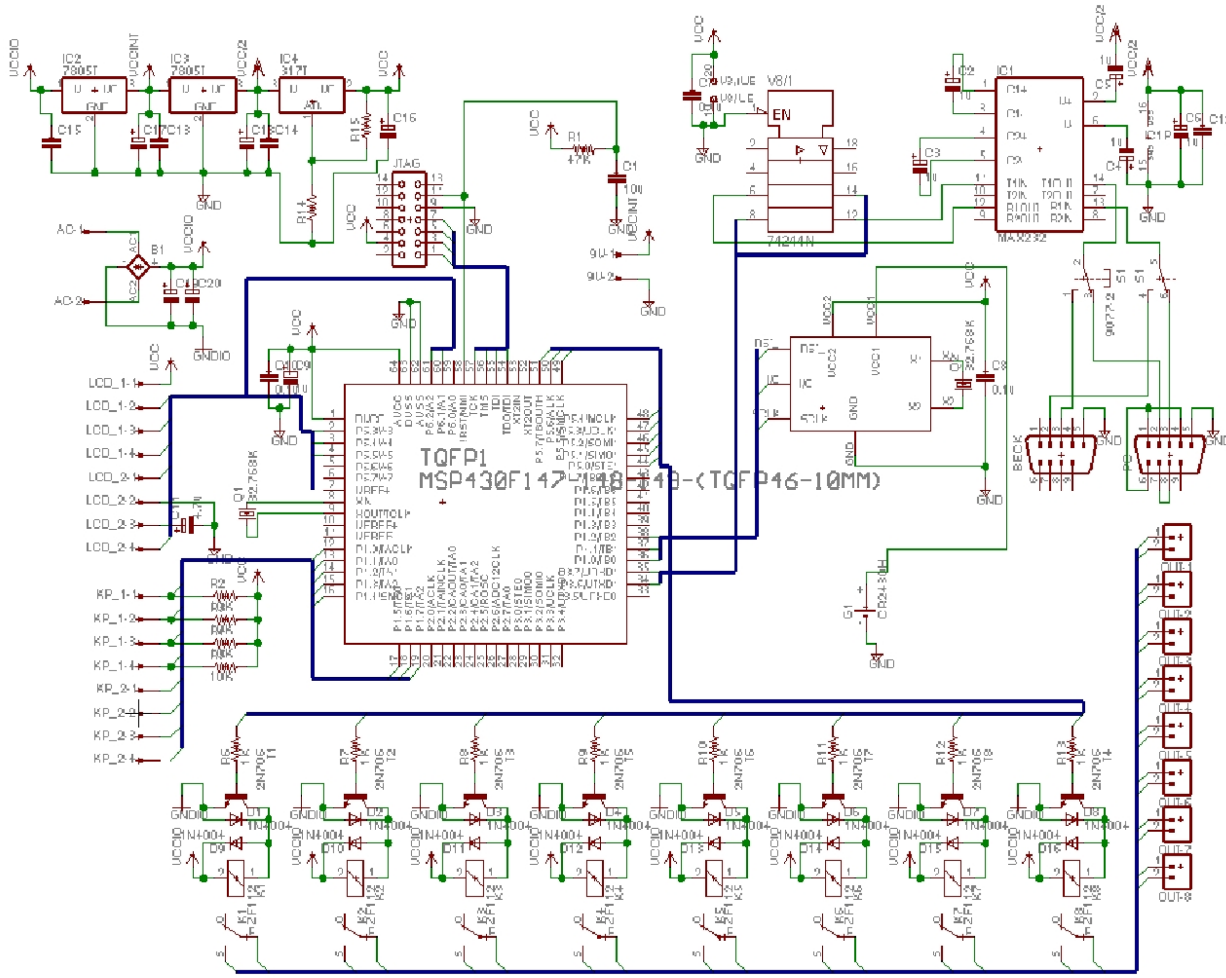

Fig. 2: Schematic of the entire circuit.

3.1 Microcontroller

The MSP430F1610 microcontroller [1] requires a 3.3V power supply. We can use three different clock sources at the same time. The first one is the internal DCO (Digitally Controlled Oscillator) based clock with RC type characteristics. The second is an optional high-frequency oscillator that can be used with standard crystals or external clock sources in the 450 KHz to 8 MHz clock range, connected to the XT2 terminals. The final one is either a low frequency or high frequency oscillator, which can be used either with 32.768 KHz watch crystals, or resonators in the 450 KHz to 8 MHz high frequency range, connected to the XT1 terminals. These three clock signals are named DCOCLK, XT2CLK and LFXT1CLK respectively. For our purposes, we are using the DCOCLK for computation purposes and the LFXT1CLK for interrupt based time keeping within the microcontroller.

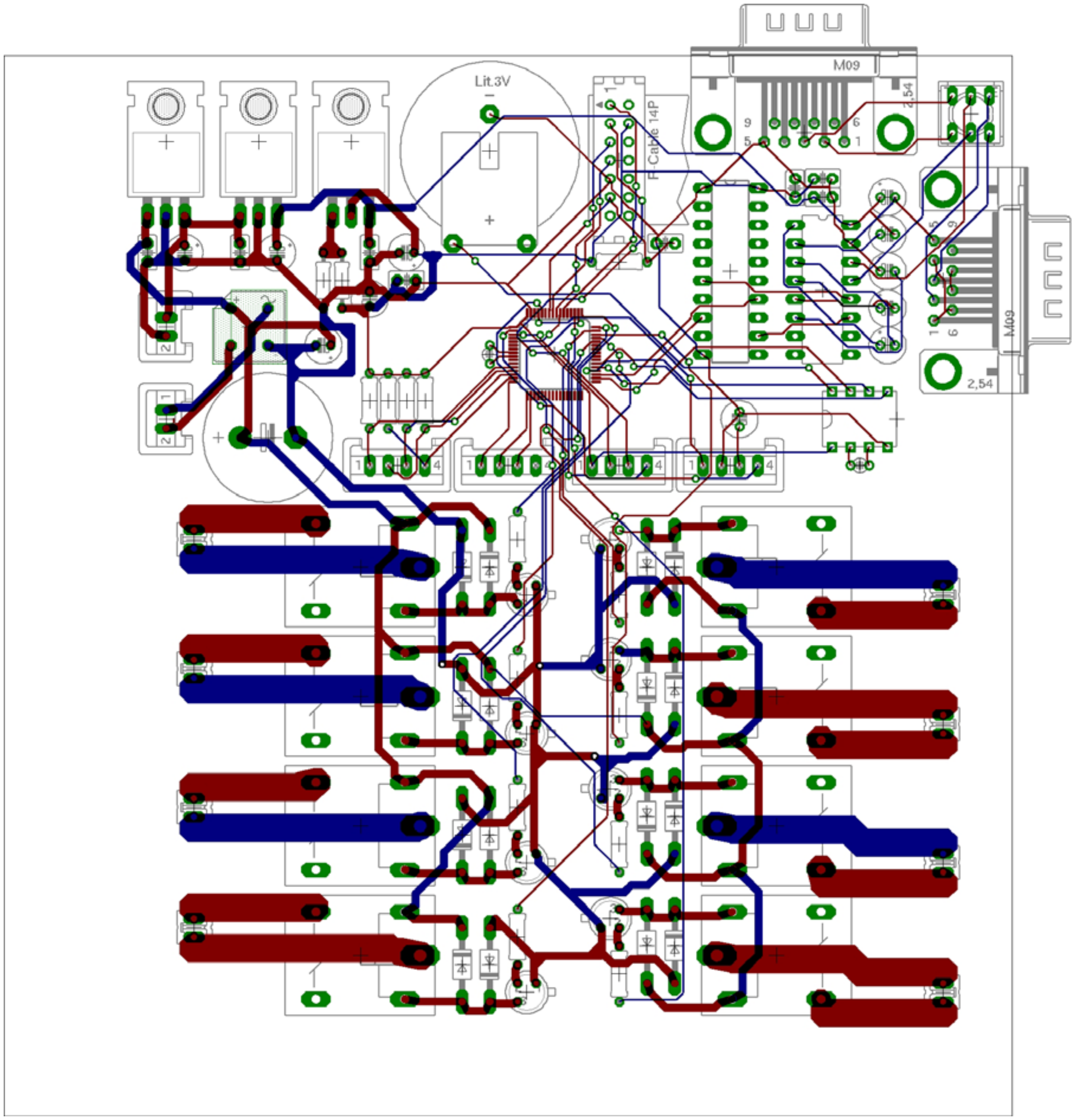

Fig. 3: PCB layout of the circuit.

Low power modes (LPM) are another of many features of MSP430 series microcontrollers. There are five different levels of LPMs in MSP430, namely LPM0, LPM1, LPM2, LPM3 and LPM4, arranged in decreasing typical current consumption. While typically active mode demands 350 μA current, LPM4 requires only 0.1 μA. However in LPM4, the CPU and all clocks are switched off. But we need to maintain the time, hence the basic clock interrupt needs to be kept alive. Hence, we use the LPM3 mode whenever the device is inactive.

This microcontroller can be programmed and debugged using a JTAG interface. A JTAG adapter for MSP430 can connect to a PC through a parallel port, and interacts with the MSP with five of its terminals, namely TDO, TDI, TMS, TCK and RST. The schematic of the JTAG adapter for MSP430 is given in Fig: 4. The connector at the right of the schematic is a FRC connector, where the pins 1,3,5,7 and 11 are connected to TD), TDI, TMS, TCK and RST pins of the microcontroller respectively. A big advantage of using a JTAG interface [9] is ability to do hardware debugging. IAR embedded workbench [11] is compatible with the JTAG adaptor made in this way.

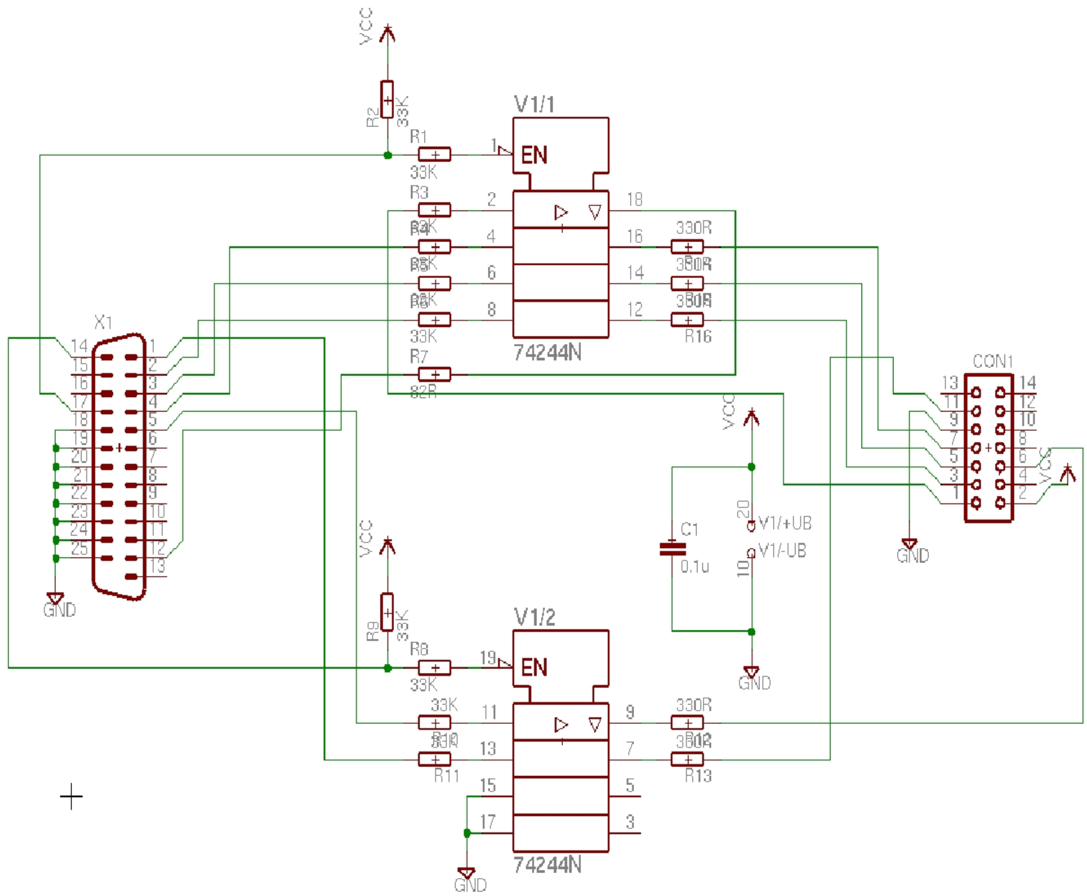

Fig. 4: Schematic of the JTAG emulator interface.

Advantages of using the MSP430 controller over 8051 family microcontrollers are presence of maximum 16 interrupt enabled I/O ports, large RAM size of 5 Kb, UART communication port, low power modes, presence of internally generated clock, JTAG interface for hardware debugging and less area occupancy on PCB. The only disadvantage that we have faced is handling of the MSP430 microcontroller is difficult as it is very sensitive to static charge, and a non standard non-DIP package.

3.2 LCD display

The Nokia 3315 display have 8 pins for connections, out of which 5 are required to be connected to the microcontroller [7]. The connection diagram of the LCD is given in Fig. 5. The five data lines for LCD control are SCLK, SDIN, DORC, SCE and RST.

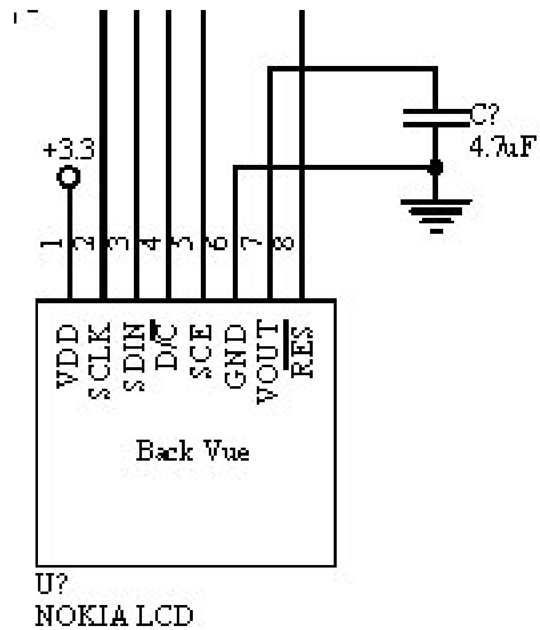

Fig. 5: Connection to the Nokia LCD from the microcontroller

3.3 Real Time Clock

DS1302 [2] is used as a real time clock (RTC) chip in our project. It is capable of having a battery backup, and has only 3 pins for connection to the microcontroller due to serial transfer of data. A typical connection to the microcontroller is shown in Fig. 6. A more detailed operation is explained by timing diagrams. The RTC requires a 32.768 KHz watch crystal for its time-keeping operation.

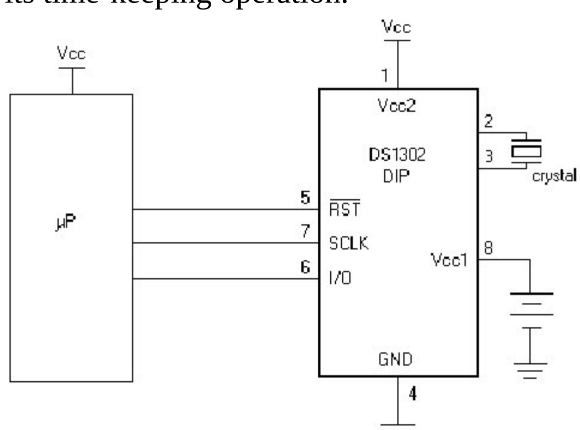

Fig. 6: Interfacing the RTC to the microcontroller

3.4 Keypad

4x4 Matrix keypad is used in our project. It is interfaced with the microcontroller (MSP430) with the help of 8 data lines [10]. The circuit diagram is given in Fig. 7. The scan lines generate interrupt on the microcontroller so that a key press is detected. The use of the keypad in the device is for network configuration.

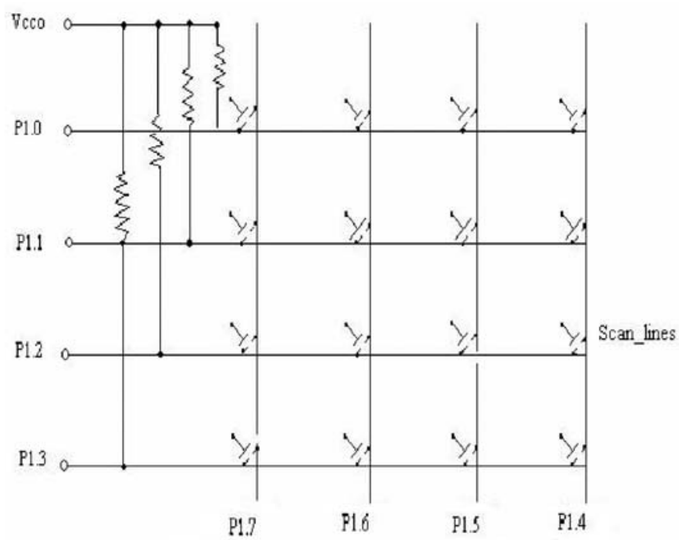

Fig. 7: Interfacing the keypad to the microcntroller

3.5 RS232 Interfacing

The Ethernet controller has a RS232 interface through which data can be transferred to and from the microcontroller [8]. However, even with a UART port, the microcontroller cannot directly communicate with the Ethernet controller since the voltage levels are different. This requires the use of a RS232-3V logic level converter arrangement. A MAX232 converts the RS232 levels to 5V digital logic levels. The 5V logic level is then converted to 3V logic level by 74HC244, which is a tristate buffer. This arrangement is shown in schematic in Fig. 8.

3.6 Ethernet Controller

The Ethernet controller [4] interfaces to the microcontroller through the RS232 interface. One can telnet to the controller at the IP configuration it is set to. Whatever data is transferred over telnet is also reflected over the RS232 channel, and vice versa. This property is used for data transfer. For data transfer from LAN to microcontroller, one connects through telnet, and starts data transfer. This data reaches the microcontroller through RS232 interface. Similarly for someone connected over telnet, data can be received if the microcontroller sends data to the RS232 interface. A picture of the Ethernet controller is given in Fig. 9.

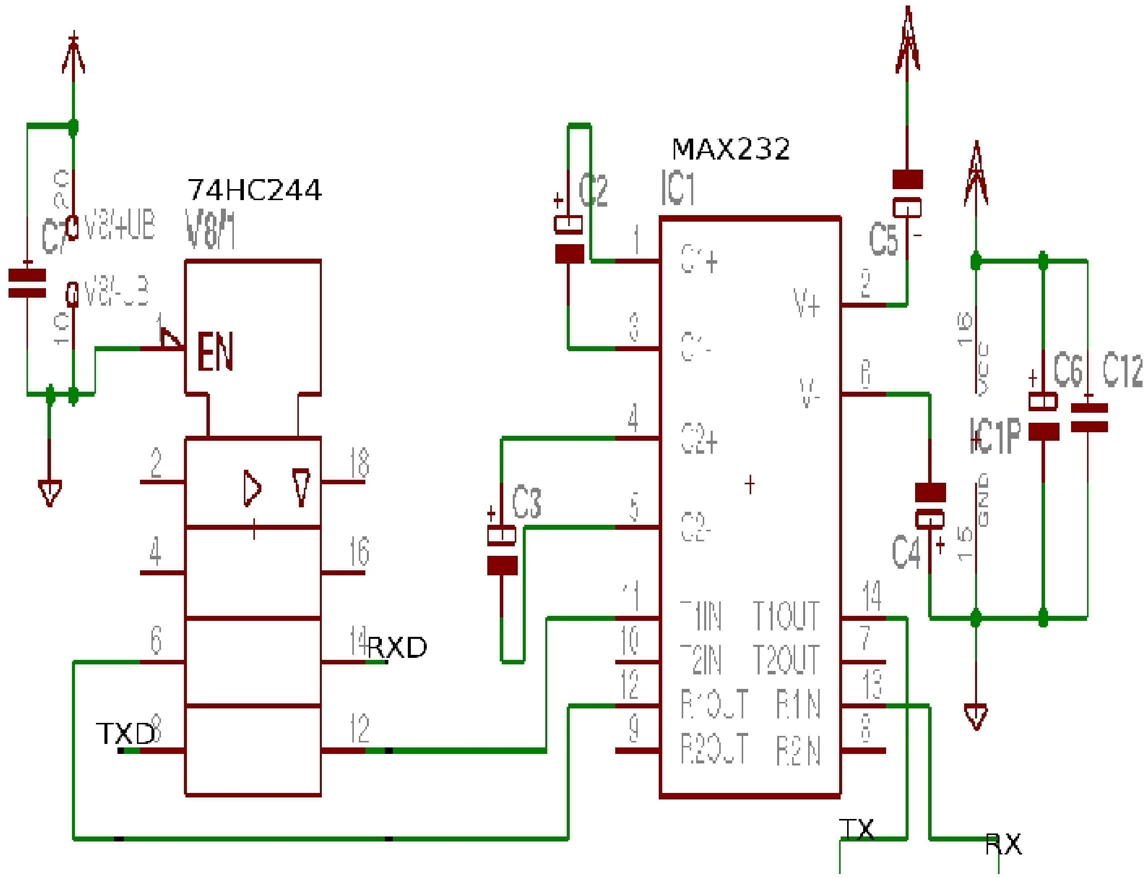

Fig. 8: RS232 interfacing of the microcontroller

Fig. 9: Picture of the BECK-IPC Ethernet controller

3.7 Power Supply

The unit is driven from 230V mains power supply. The transformer used is of 230V/15V 2A rating. A bridge rectifier and a filter circuit produce an almost constant DC voltage of 14V. Since four different voltage levels are used for the working of our device, three regulators are used, and the fourth voltage level is the 14V unregulated DC supply, given to the relays. The entire power supply schematic is shown in Fig. 10.

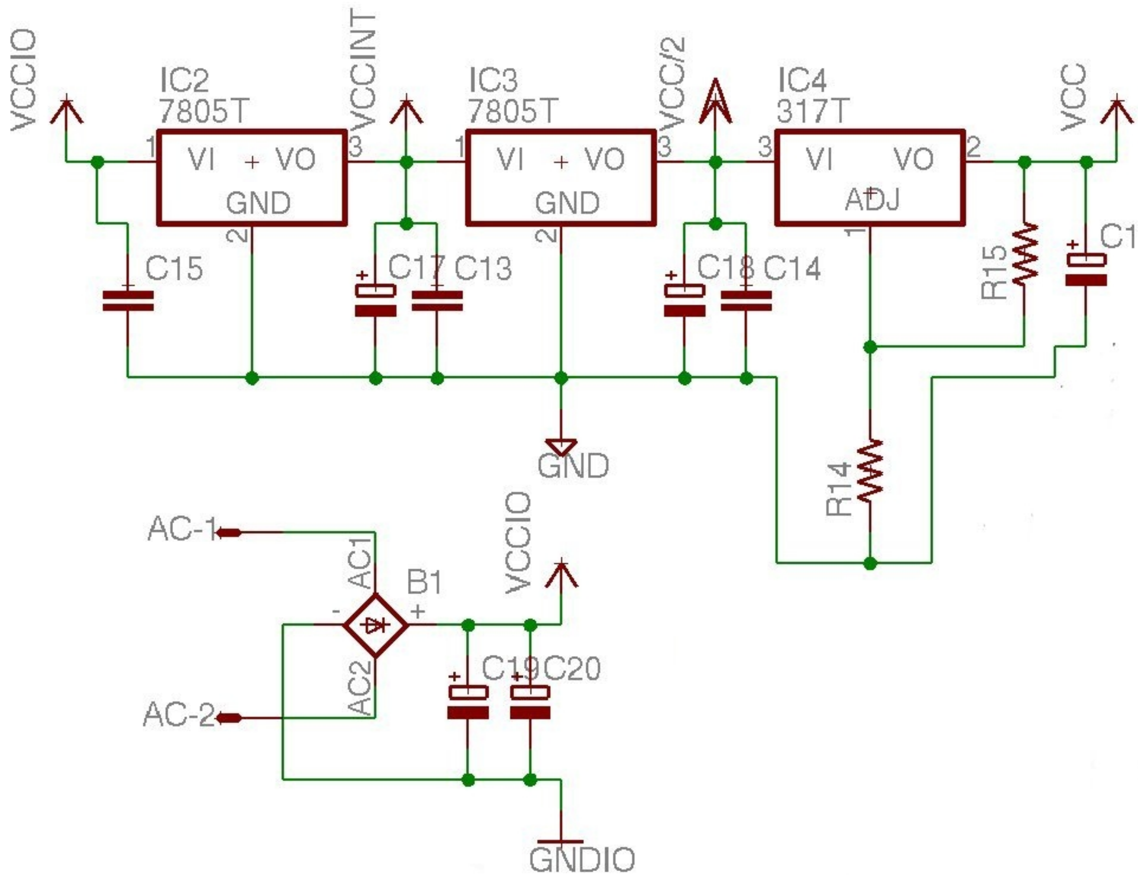

Fig. 10: The rectifier-filter-regulator unit of the power supply module

3.8 Output section

The main purpose of our device is to switch electrical appliances. We use relays, where the appliances are connected between the common and N/O terminals of the relay. When the coil becomes active, the common and N/O terminals get shorted, and the appliance switches on. Since the resistance of the coil is 30Ω and the coil is connected to a 14V unregulated power supply, an n-p-n transistor is used as a switch, where the microcontroller drives the base. The connection for one relay is given in Fig. 11. The diodes are used to bypass switching transients.

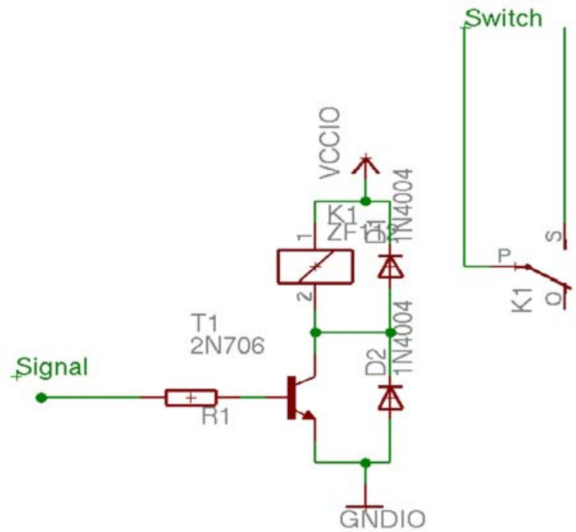

Fig. 11: Driving the relay units using a transistor as a switch

4. **Software**

4.1 Alarm Features

The most important feature of our device is that it can provide highly configurable alarms, given to eight different ports. When the user sets the alarms, the following options are provided to the user.

1. Periodicity: Alarms can be periodic hourly, daily, weekly and monthly, or the user can even set single non periodic alarms.

2. Duration of alarm: The user can set the number of minutes for which the alarm should remain activated.

3. On/Off time: If the user wants, he can provide for a pulsating alarm, with configurable on-time and off-time.

4. Output port: There are eight different output ports, through which eight different appliances can be controlled. Each port behaves independently of one another.

4.2 Microcontroller Programming

This part can be best understood with the help of a flowchart as shown in Fig. 12. The flowchart is described below in brief, block by block.

When the user will switch on the device then the microcontroller will read the RTC to restore the time as the RTC has a backup battery. Then it will initialize the LCD display and will go to idle state. There are three different interrupts which will take MSP out of its idle state. The interrupts are Clock interrupt, UART interrupt and Keypad interrupt.

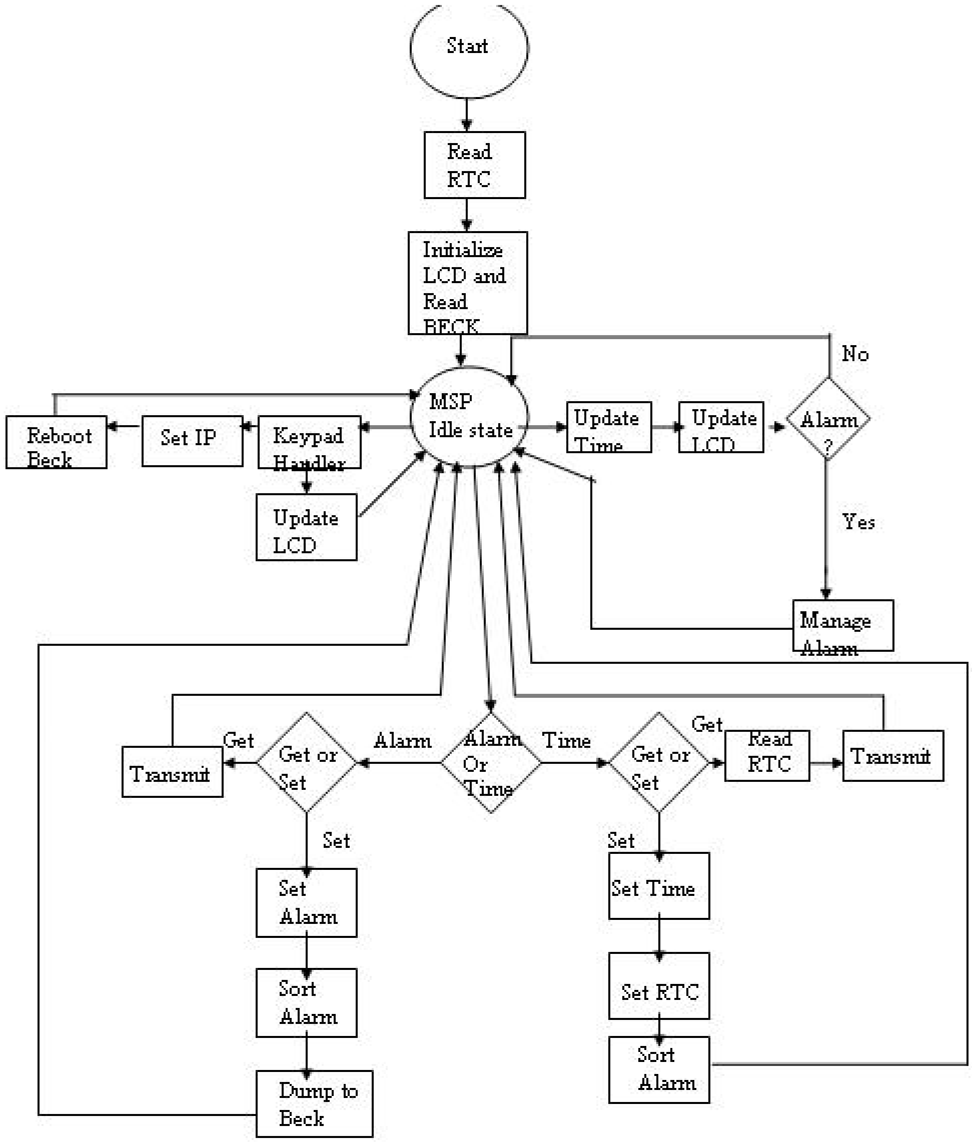

Fig. 12: Software flow chart

1. Clock interrupt comes regularly with a period of 1 sec. This is generated from the LFXT1CLK. When the microcontroller receives this interrupt, it will update the time and LCD display; manage the alarms and goes back to the idle state.

2. The UART interrupt will be generated for any receive event on the terminal. The UART is primarily used for setting a new alarm, or time, and retrieving the alarms and current time. Every time the microcontroller receives data (alarm data or time data) from

the UART terminal it sorts all the alarm and gets the immediate alarm. It then dumps back all alarms to the Ethernet controller for non-volatile storage.

3. Keypad will generate an interrupt when it detects that a key is pressed. The keypad is used for setting the network configurations, so when the microcontroller receives this interrupt it stores the key which has been pressed and updates the LCD display. Then when the user is done with network configurations it sets the parameters of the Ethernet controller given by the user.

4.3 Computer GUI Programming

The device can be properly used with a GUI on the computer, which acts as an interface for setting time and setting alarms. All these operations are possible over LAN or through serial communication, through an interface developed in java. All functions of the alarms are implemented in the GUI, and the user can add a new alarm, edit or delete an existing alarm. More details on the usage of this GUI have been provided in the user manual. The GUI has been developed in java using the netbeans IDE [14].

5. **Suggestions for Further Improvements**

We have achieved what we have aimed but still we can improve on it by following ways:

1. Wi-Fi enable: We can use a wireless Ethernet controller to make the device Wi-Fi enabled.

2. Ports and Alarms: We can increase the number of ports it is driving and also the maximum number of alarms which can be set by using an advanced microcontroller.

3. GUI: We can also improve the GUI by giving more functions on the ports it is driving. For example, if one of the port is connected to a water heater then by giving an option to set the temperature of water.

We can improve on this device and can use it for smart home purposes. Every electrical appliance in the house will be connected through this device and the user can connect to the device from any remote location through internet. Then the user can switch on and off the appliances and will also be able to perform various functions. For example, the user can switch on the air condition of the house at a given time and also he will be able to set the temperature the air condition will maintain. This is the most important use of this device, through this way we can operate every appliance through internet and this will increase the comfort.

6. **Acknowledgements**

We would like to acknowledge the immense help provided by Prof. P.C. Pandey, Mr. V.K. Tandon, and Dr. L.R. Subramanayan, which was hugely responsible for the successful completion and working of the project. We would also like to thank the lab assistants, seniors and the TA's.

References

[1] Texas Instruments, MSP430F1610 datasheet, *http://www.ti.com*

[2] Dallas Semiconductors, DS1302 datasheet, *http://www.maxim-ic.com*

[3] MAXIM Semiconductors, MAX232 datasheet, *http://www.maxim-ic.com*

[4] BECK-IPC, IPC@BECK, DK51 embedded Ethernet datasheet, *http://www.beck-ipc.com*

[5] Phillips Semiconductors, PCD8544 84x48 pixel resolution Nokia 3310 LCD controller, *http://semiconductors.phillips.com*

[6] All Other Datasheets, *http://www.alldatasheet.com*

[7] sbissonnette@microsyl.com, Nokia LCD interfacing, *http://microsyl.com*

[8] Craig Peacock, RS232 documentation, *http://beyondlogic.org*

[9] Olimex Ltd, MSP430 JTAG, *http://www.olimex.com*

[10] EE Department, IIT Bombay (microprocessor laboratory webpage), keypad interfacing, *http://www.ee.iitb.ac.in/~ee315*

[11] IAR systems, IAR Embedded Workbench for MSP430, *http://www.iar.com*

[12] CadSoft Online, Eagle Layout Editor, *http://www.cadsoftusa.com*

[13] Borland Software Corporation, Borland C++ IDE, *http://www.borland.com*

[14] Netbeans, Netbeans java IDE, *http://www.netbeans.org*