EE389 Electronic Design Lab II Project Report, EE Dept, IIT Bombay, 2 December 05

LAN Based Power Monitoring System

Group No: D13
Ayush Mittal (02D07029) <ayush@ee.iitb.ac.in>
M Anirudh (02D07030) <anirudh@ee.iitb.ac.in>
Satish Meena (02D07037) <sattu@ee.iitb.ac.in>
Supervisor: Prof. N. K. Khosla
Course Instructors: Prof. P. C. Pandey, Prof. L. R. Subramanian.

1. Introduction This project is aimed at devising an electrical monitoring system that can measure the three phase power at an electrical outlet and store this information in a database. The monitoring system is intended to be remote and autonomous; the data acquisition board will consist of a microcontroller along with a few components to complete the data gathering. Also since the medium of transmission is the LAN there has to be some mechanism that allows us to interface a microcontroller with the LAN. To do this we are using a chip called Beck IPC which has two serial ports and one Ethernet port. We will touch upon the working and features of this component in detail later. The next part of the project is to create and maintain a database. This is done on a Linux server which receives data from the Beck IPC. The Linux server also runs a java based application which when sent a query will display different information and graphs, e.g. the power consumed at a particular instance, a graph of the voltage over time etc.

There are a few clarifications to be made regarding the approach we have used in our project. Firstly, instead of measuring the instantaneous voltage, current and power factor and then multiplying to get the power and storing the data, we are actually measuring the voltages and currents one after another and transmitting the data. The power is calculated at the final stage i.e. the Linux server which multiplies the current, voltage and the power factor of each phase and calculates the power. This approach was taken in the light that we did not want this project to be solely power measurement. We wanted the data acquisition board to be a more general purpose board that can transmit voltages representing not just the current and voltages but also voltages sensed from other sensors like temperature, pressure etc. This way by changing the sensor board we can use the project to acquire a wide variety of data rather than just voltages and currents. Secondly, we are using an independent ADC (Analog/Digital converter) and an independent microcontroller and not a microcontroller with an attached ADC. This is partly because the ADC in use, the ICL7135, is easy to interface with a microcontroller.

2. Hardware Block

Fig 1 shows the block diagram of the entire project

1) Power Sensor Board: This board takes the three phase inputs from a terminal and feeds them to the 3 voltage and current transformers present on the board. The voltage transformers will step down the voltage from 220V to less than 5V. The current

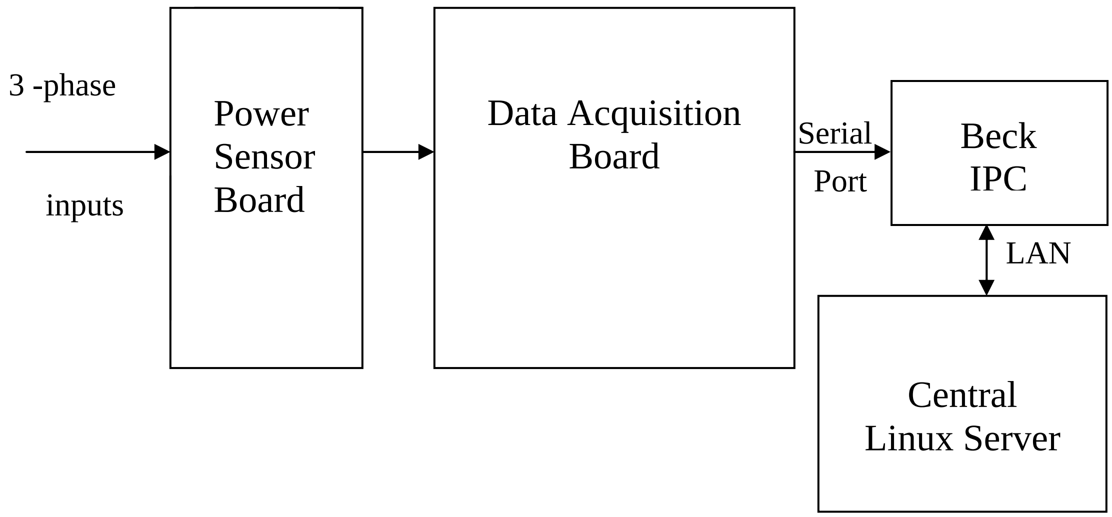

Fig 1 – Complete Block Diagram

transformers will provide the currents in each phase in voltage form. The outputs from this stage are 6 signals – 3 representing the voltages in each phase and 3 representing the currents in each phase.

2) Data Acquisition Board: This is the primary block of the project. This receives the 6 voltages from the power sensor board. The output data of this block is a RS-232 serial data transmission to the Beck IPC. The data contained in the serial data transmission is the average values of the 6 voltages, the phase differences between the signals representing the voltages and currents of the three phases, the temperature and a flag byte.

3) Beck IPC: The Beck IPC is an embedded controller designed to WEB- or LAN-enable products and includes a 16-bit 186-Processor, RAM, Flash-Disk, serial interfaces, an Ethernet-Controller and many I/O pins - all of this in one single DIL32-housing. It is a ready to use combination of hardware and software. The hardware consists of 16 bit 186 CPU, RAM, Flash, Ethernet, Watchdog and power-fail detection. The preinstalled software consists of a Real Time Operating System (RTOS) with file system, TCP/IP stack, web server, FTP server, Telnet server and hardware interface layer. The operating system installed on the chip is similar to DOS and it is possible to program the chip to do certain applications. Also this chip has very easy to use interfacing with microcontrollers. The salient feature of this chip is its RTOS or Real Time Operating System. In this the operating system, application program and data can be exchanged independently of each other. The application program and data can also be controlled by remote access. The memory on the chip consists of 512 kb working storage (RAM), 256 KB operating system (Flash) and 256 KB Flash-disk. Furthermore, the flash-disk memory of the IPC can theoretically be expanded to 2 gigabytes via an external IDE-interface by memory-on-chip or compact-flash-memory. Beyond that, the IPC operating system contains a

complete API (application programming interface), which makes access to all functions of the IPC and its operating system possible, as e.g. the control of the E/A- interface and access to the TCP/IP services.

4) Central Linux Server: This is Linux based server which has a database server running on it. We have used MySql database server. This server regularly pulls data from the Beck IPC and stores the data in the database. The data from the Beck IPC is just the average voltages, currents and phase differences and the temperature. All these are stored in the database. This server also runs a java based application that provides a graph or a particular value as requested by the query.

3. Circuit Design: In this section we shall discuss the circuits of the first two blocks.

1. Power Sensor Board: There are 3 voltage and 3 current transformers on this board. The voltage transformers are used to step down the voltage from 220V to less than 5V. The current transformers have a rating of 5A to 4mA.

2. Data Acquisition Board: Fig 2 shows the block diagram of the entire block. The individual components and circuits will also be addressed in detail.

a) AC/DC Converter: Fig 3 shows the circuit used for a AC/DC converter. An AC/DC converter takes in a sine wave as an input and gives a square wave as an output. This circuit takes in a sine wave voltage input with amplitude between 0 to 5V and the output is a dc voltage that ranges between 0 to 2V

b) Comparator Circuit: This is the circuit for the comparator. The input is a sine wave and the output is a square wave with voltages 5V and 0V.

c) Multiplexer: The multiplexer used is a CD4051 8:1 analog multiplexer. The select lines come from uC2. The six input voltages go to six pins. Of the remaining pins one is used to for temperature measurement while the other will be used as reference voltage. Fig.4 shows diagram of the multiplexer. X0 to X7 are the pins for input voltages. Pins A, B and C are for the select lines. INH or inhibit pin is to prevent the multiplexer from transmitting any data.

d) ADC- ICL 7135: This is a dual slope integrating ADC. This chip integrates the signal for 20000 clock cycles and de-integrates for the next 20002 cycles. The clock cycles are determined by the clock in frequency. The RUN/HOLD pin determines the start of the signal conversion. The RUN/HOLD pin if open or high continues conversion. Effectively this averages the signal over the entire time range and eliminates the noise. The input range of the signals can vary from –V to +V supplied voltages. However we are rectifying the signal because a symmetric signal will give an average value of zero. The under-range and over-range pins go high if the measurement result falls below the range (in our case 1800/10000= .18 V) or goes above the range. The BUSY signal is high until the first zero crossing in the de-integrating phase else until the end of conversion. The RUN/HOLD pin if open or high continues conversion. However if it is given a low input

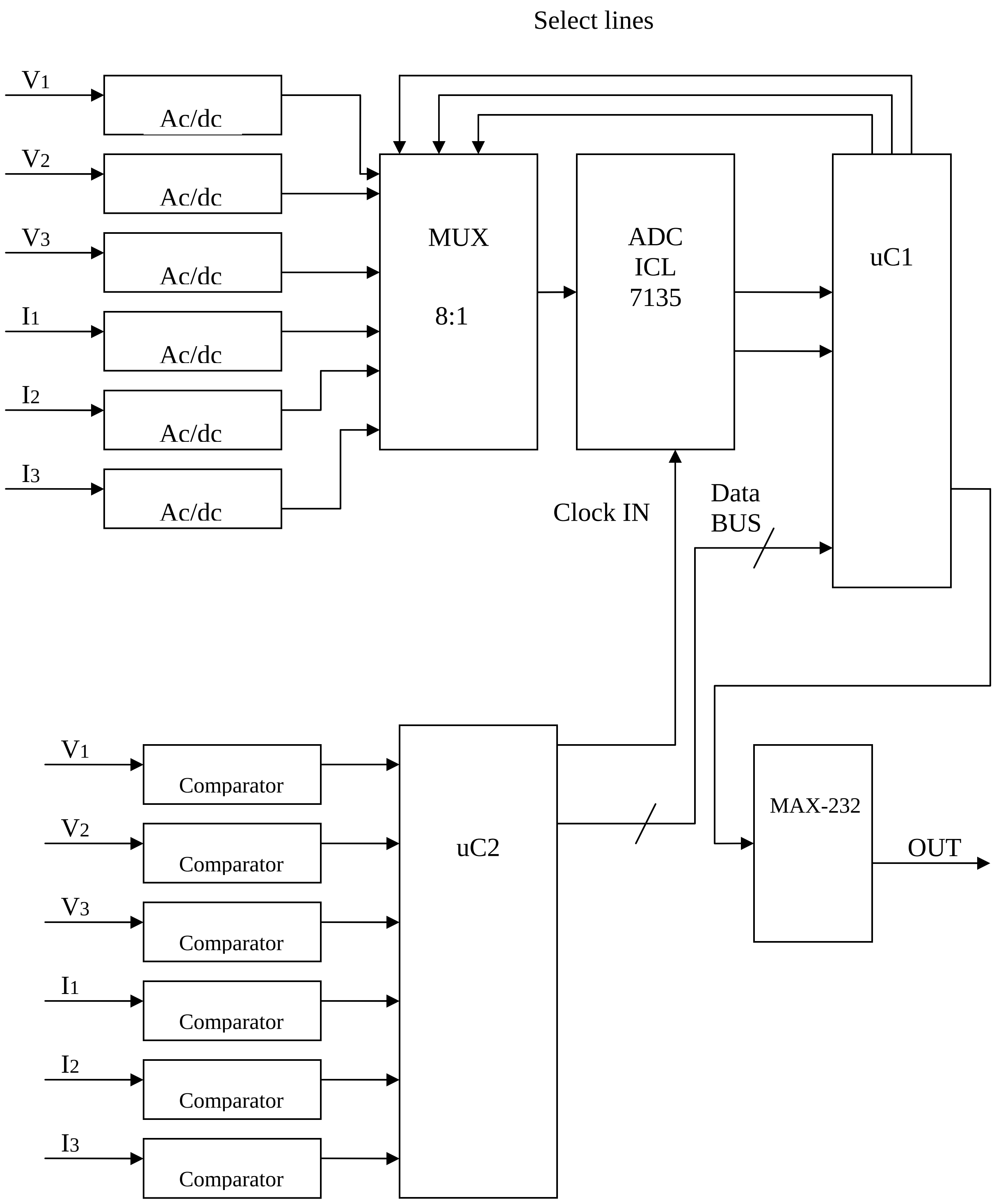

Fig 2 – Block diagram of the Data Gathering Unit

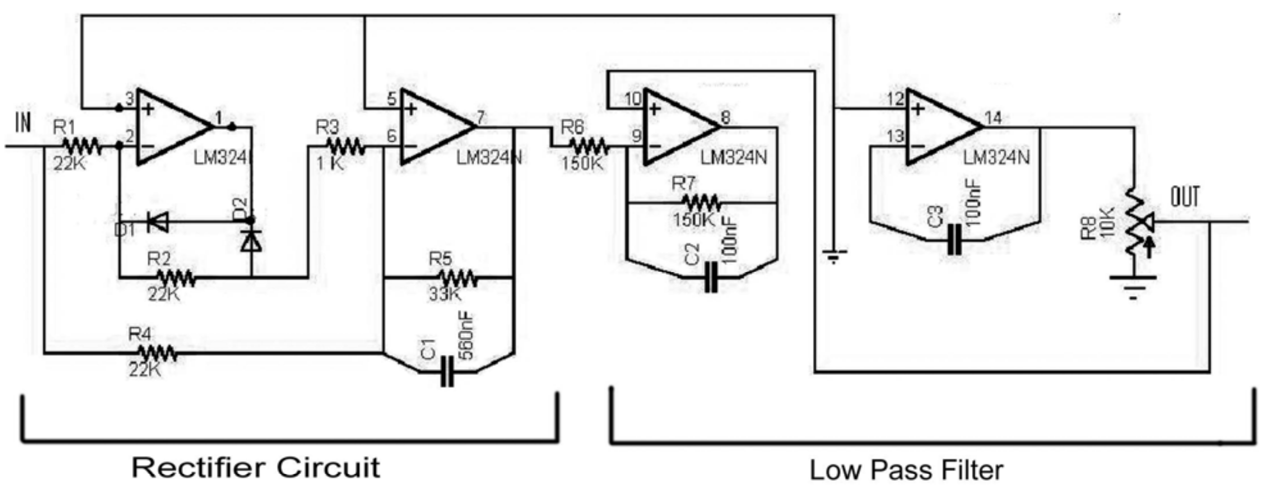

Fig 3 – AC to DC converter

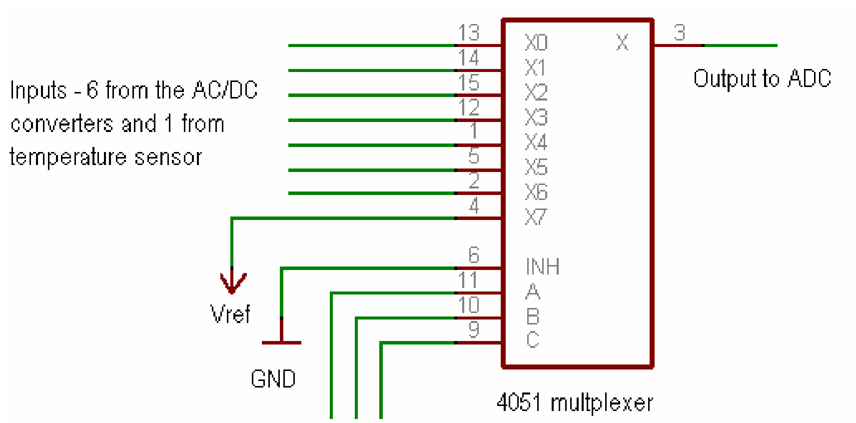

Fig 4 – Connection diagram of the multiplexer

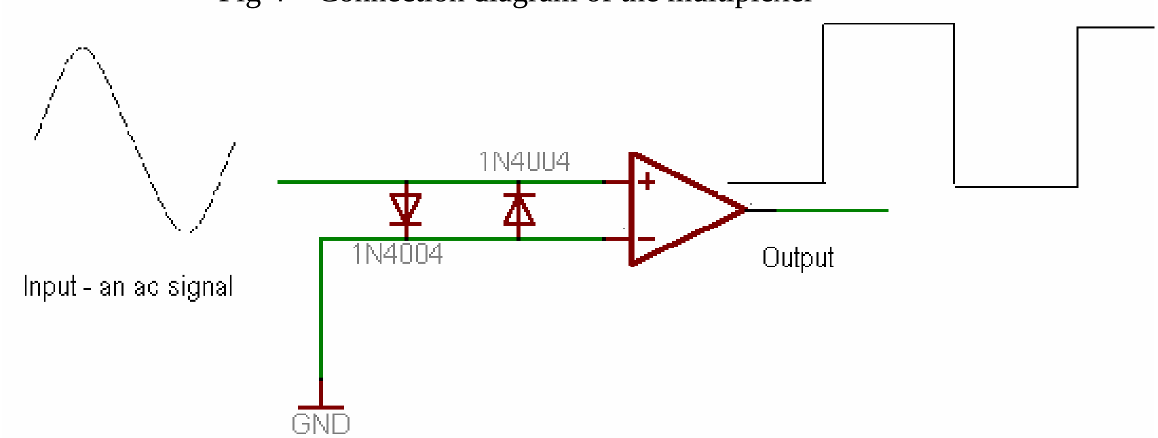

Fig 5 - Comparator

the ADC completes the conversion in progress and then remain in the auto zero phase while holding the last value. The STROBE pin is a negative going pulse that goes low at the end of conversion. The output

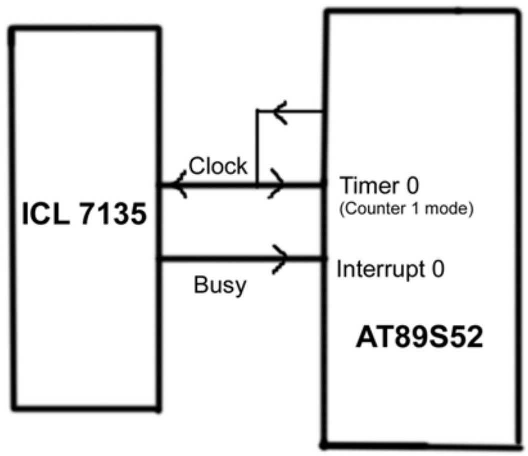

Fig 5- Interfacing ICL7135 with microcontroller

count value is 10000 * (V_{in}/V_{ref}) . The internal data latches are latched during the first clock cycle after busy goes low. Since Busy is high for 10000 + reading + 1 clock pulse, by sending the "AND" of busy and the clock to a counter and subtracting 10001 from the count will give us the reading. This is the method we have used for our circuit.

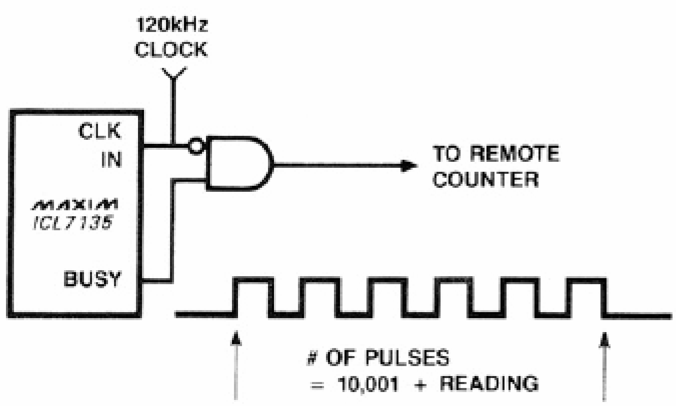

Fig 6- Interfacing ICL7135 with microcontroller

e) MAX 232: This is to convert the TTL/CMOS signals into RS232 form.

Working: In addition to the above components we have two microcontrollers. The uC1 is used to measure the phase measurement. This uC1 gets the square wave output from the

comparators. It checks for the falling edges at each of the pins and notes the time difference between the falling edges of the different inputs. It transmits data to the main microcontroller uC2 using parallel port transmission. This uC1 also generates the clock for the ICL 7135.
The inputs going to the multiplexer are multiplexed one by one and sent to the ADC. This ADC is a slow ADC and takes around 0.4 seconds to make a conversion. The speed is limited by the clock frequency (it integrates and de-integrates for 20000 clock cycles) and the frequency being used by us is 100 kHz. We have seven data lines thus after 7 cycles of data conversion the output is ready for transmission. All the data is converted to ASCII before transmission.

4. Software

i) Microcontroller codes- We have used two microcontrollers on our Data Acquisition Board. One of them is AT89S52 and other is AT89C52. The central controller is AT89S52 which generates signals for the multiplexer, takes the data from the ADC ICL7135 as shown in Fig. 5 and stores this data into a stack. Parallel to this it also receives phase difference data from the second microcontroller and stores it in another stack. The data transmission between the microcontrollers is interrupt driven parallel communication. When one complete set of data has been collected by this microcontroller it converts them into ASCII format and moves them into a transmit buffer from where they are transmitted serially to Beck IPC. The transmit buffer is used because serial transmission is a slow process.
The second microcontroller detects the zero-crossings of the pulses given to it from the comparator and stores the time at which the 6 individual lines go to zero. For this it uses timer in mode 1.

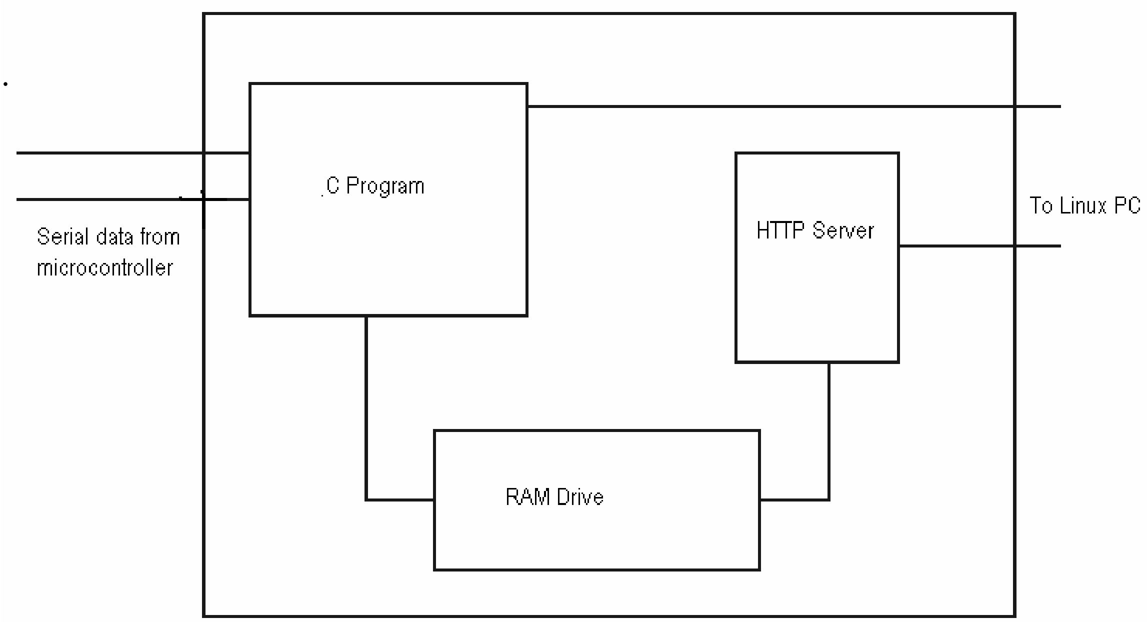

Fig 7 -Beck IPC block diagram

ii) Beck IPC: Serial to Ethernet converter: Beck IPC is an 80186 based device with a reduced form of DOS installed and running over it. It has got built in HTTP server and user can write system specific applications in C programming language. For our system we require an application to interact with our AT89S52 microcontroller based data gathering board using serial communication. It should be able to collect the serial data being transmitted by microcontroller, process it to convert it to meaningful values and store it over the available RAM disk. Now this RAM disk being limited in size, the file needs to be deleted after a while and created again. This also has to be taken care of by the program. The program also interacts with the LINUX based PC to ensure error free transfer of data from here to the database

Working of the system: When the system boots up, it executes AUTOEXEC.BAT file created by us. This file starts the C application by calling the executable file of this application stored on device's flash memory. This C program uses the "stdaux" input stream defined in DOS to read the serial port data and puts it into a file in RAM drive. This program uses multithreading to create a parallel process which allows the Java application on Linux server to connect to itself using TCP/IP socket. It sleeps while the data is being received and as soon as the file on RAM disk is ready for transmission it transmits a signal to the java application indicating it to pick up the data. The built in HTTP server (configured by us to read the data from RAM) allows this Java application to access the data stored in the file. The C program sleeps meanwhile when the data is not coming, thus saving processor's computational power and allowing other applications to run.

.

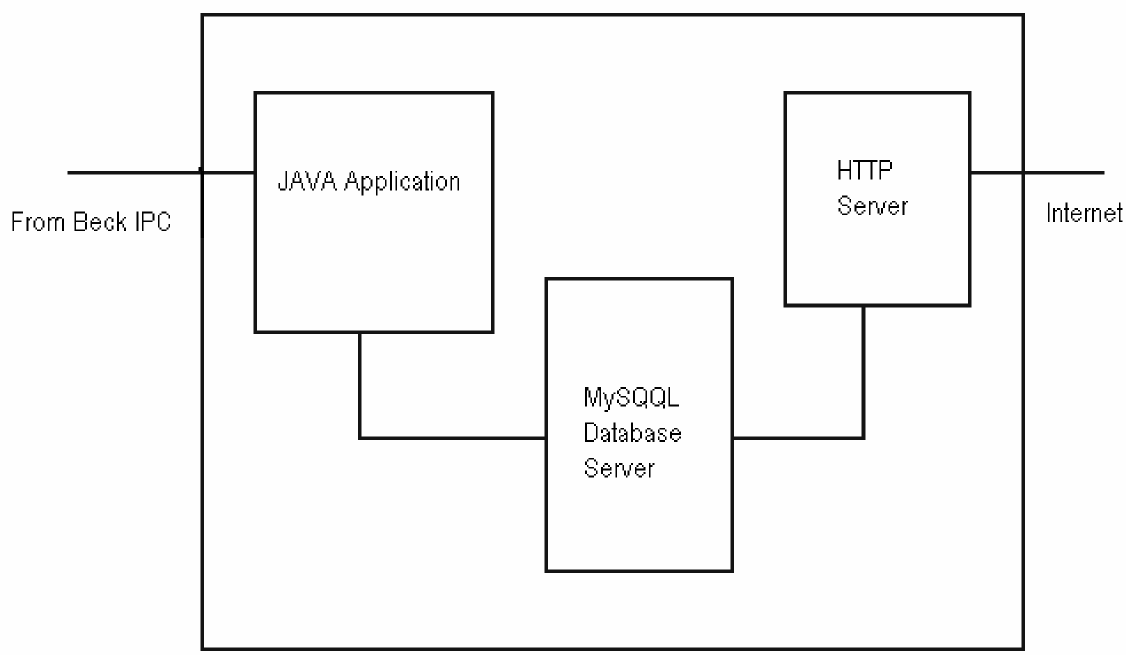

Fig 8 – Linux Server Block Diagram

The programming of this device requires sufficient device specific knowledge and it also covers various aspects of programming like multithreading, network programming, floating point arithmetic which have to be implemented differently than the common approach because of the device architecture

iii) LINUX PC based database and HTTP server: This part of our project has two separate software components.

a) A "Java" application with GUI which connects through Local Area Network (LAN) to Beck IPC picks up data from there and puts it into the database server..

We have used "MySQL" database server for this purpose. The Java application makes use of JDBC and "MySQL" drivers to connect to the database server and execute queries over it. This application extends the class "Java.awt.*" to generate graphical user interface and "Java.net.*" for connecting to Beck IPC through LAN. This software follows a small protocol with Beck IPC to maintain the data transfer error free.

b) Jakarta-Tomcat HTTP server with a Java Server Page (JSP) – The JSP will provide users over internet connectivity to the database server in which we have stored the data which was gathered from the electronic board via Beck IPC. This data will also be used to generate weekly or monthly statistics. Graphs based on this data can also be generated and provided on the internet to the users. The JSP uses JDBC to connect to MySQL server

5. Conclusion and Future Scope: The applications of this monitoring system are numerous. One can obtain data from remote places and use it to monitor efficiently. Also it is possible to control the circuit via remote access because of the flexibility offered by the Beck IPC. By changing the design to suit our needs we can get a variety of data-temperature, pressure, humidity, water levels etc. Also since the data obtained is stored in a PC it is possible to archive the information which can prove to be useful later. Also one may be able to use the data to see the time variance of different parameters. This is particularly useful in power monitoring. By noting the times when the power consumption is going to be high one can conveniently shift the power distribution in such a way that the system does not have a shortage of power. Another use is in temperature monitoring. Another feature these systems can have is having two way communications that can allow some remote controlling and increase the scope of functionality. For examples, in power monitoring we can provide breakers in the circuit that can turn on or off and manage the power or in water management provide controls that can increase/decrease the flow of water.

6. References:

[1] Kenneth J Ayala, "8051 Microcontroller Programming" (Second Edition). Published by Penram International Publishing (India) Pvt. Ltd.

[2] Beck IPC Product details,

i) URL: www.beck-ipc.com/ipc/products/category/index.asp?cat=1&sp=en
Author: Beck IPC GmbH, last modified 7 November 2005.
ii) URL: www.beck-ipc.com/ipc/download/software/example.asp?sp=en
Author: Beck IPC GmbH, last modified 7 November 2005.

Appendix: The Printed Circuit Board (PCB) design for the project

Board 1: This board consists of the 6 AC/DC converters and the 6 comparators for phase measurement. There is a 12 pin connector which takes the input from the transformers and two 6 pin connectors which act as the output from the AC/DC converters and comparators respectively.

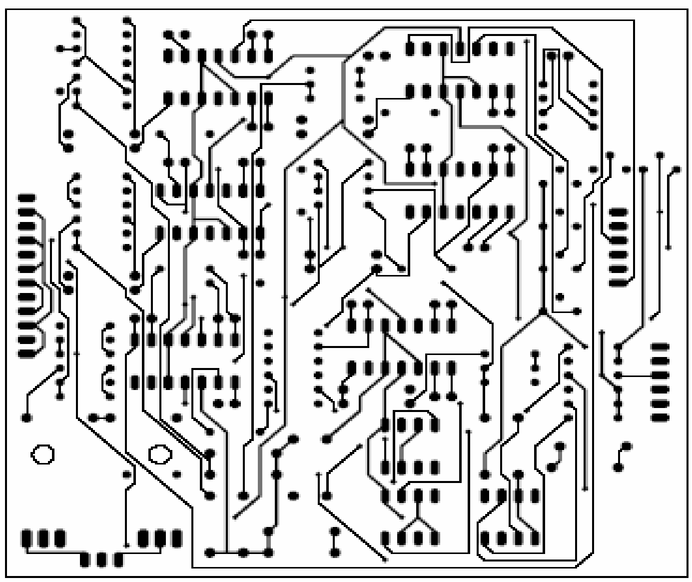

Fig a. Top surface of Board1

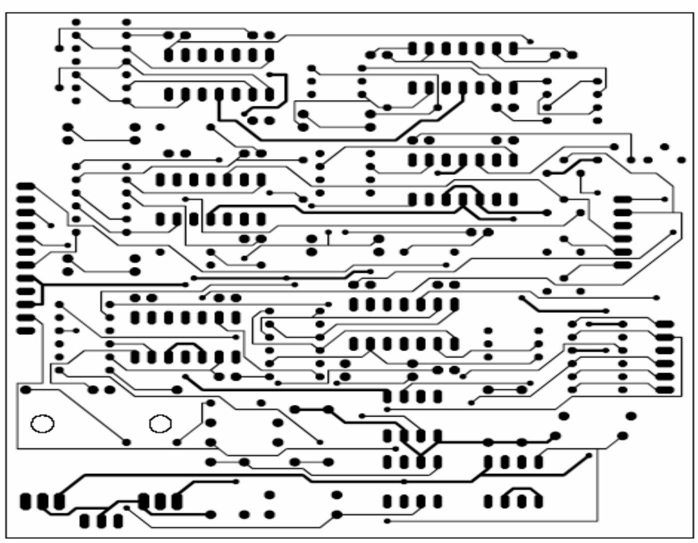

Fig b. Bottom surface of Board 1

Board 2: This board contains the microcontrollers, the ADC, the multiplexer and MAX232.

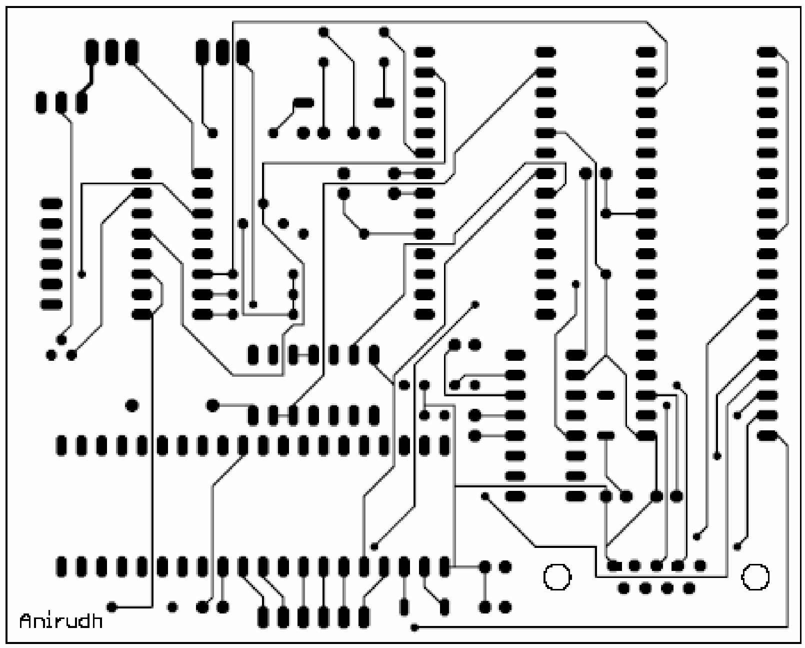

Fig c. Top surface of board 2

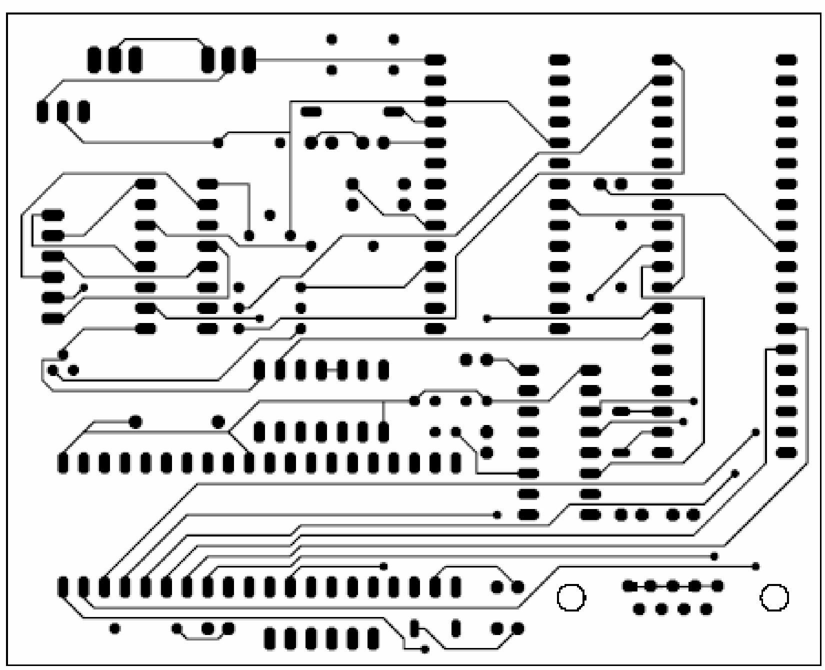

Fig d. Bottom surface of Board 2