Measuring Power Consumed By Devices

(Tanmay Verma 2008054)

Synopsis

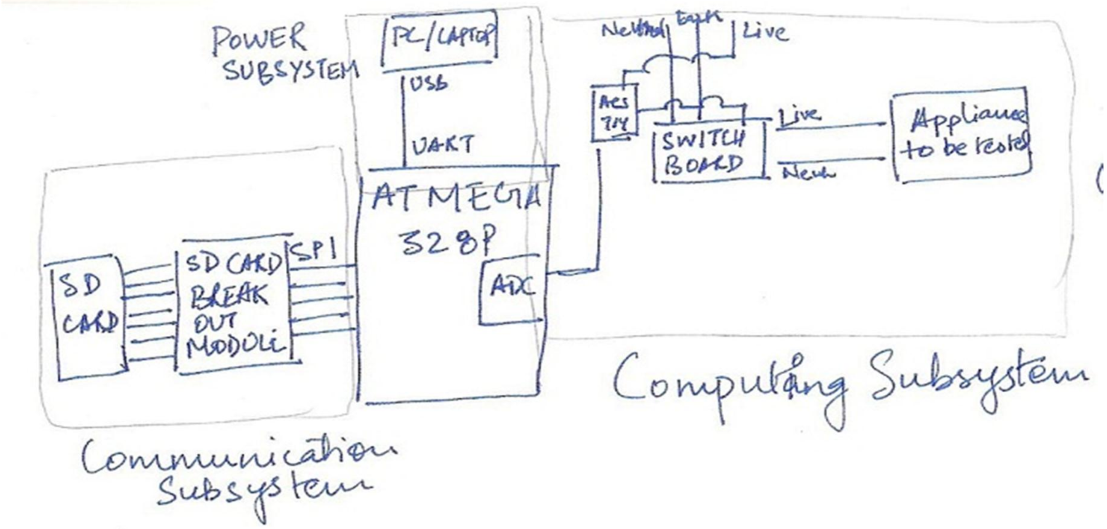
Power is one of the important resources in current times and if we can measure the amount of wastage we do, we can definitely reduce the wastage. This was the main motivation of this project. With this system one can measure the amount of power consumed by their devices, log the data collected by the system in a SD card or Micro-SD card whatever is available with the user and visualise the data in the form of a graph with the help of an easy to use executable file. This system can form a basis of a lot of applications like power saving devices, smart switch boards etc. Currently the system measures power of systems which draw current between -5A to 5A but with a simple change in sensor and formulation it can be extended to measure much larger power consuming devices.

The microcontroller was basically interfaced with a current sensor which will basically converted the drop/rise in current to drop/rise in voltage which was then be detected by the ADC present on our microcontrollers and outputted as a value between 0 and 1024 as the ADC is 10 bit.

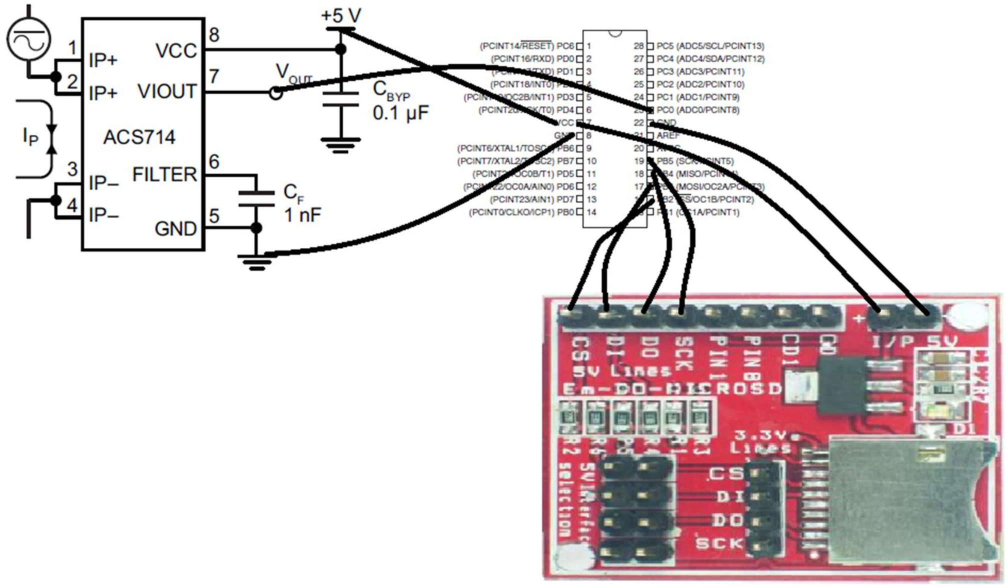
For calculation of power RMS value of current was calculated over a 192 values (the number 192 comes from the frequency of power supply and that of ADC). In 1second around 50 values get calculated so an average of 50 RMS values is taken and then the power in WATTS. The power calculated is logged in a text file log.txt in the memory card's root directory.

When this log.txt is given as input to the executable a graph showing your power consumption is displayed.

Architecture



Hardware Involved



1. Microcontroller Board: ATMega328p, Cost: Rs 1400/-

2. Current Sensor: ACS714 Current Sensor, Cost: Rs590/-), Vendor: Rhydolabz India

3. SD Card Reader: Code works for two breakout boards

a) Micro SD Breakout module, borrowed from Lab

b) SD Card Module, Cost: Rs 30/-, Vendor: www.onlinetps.com

Software

The following are the Software Components of the code:

a. Current Sensor Calculation: ADC code of Assignment 3 was modified to run in Auto Trigger on Conversion Complete Mode.
b. Open Source libraries of SD-Card reader were used and the implementation just required file writing so every time the program is run the old Log file is deleted and a new one is made.
c. For Debugging rather than using LCD to display values UART interface was used to display the values in the terminal. This required defining the appropriate baud rate which is the number of signal changes in a second. Appropriate open source libraries were used for UART functions.

Learning

The most important gain out of the project was the calculation and calibration of the current sensor. The sensor required to be soldered and put in series with the positive line of the power supply. A thorough reading of datasheet was required to get a proper sense of the working of sensor. The formulation of power from the Voltage value from the sensors took a lot of time and involved concepts like sensitivity of the sensor and the Zero Current Output Voltage etc.

One of the most productive output of this project was proper debugging technique in AVR. With the help of UART debugging code was much more convenient rather than using LCD. Even the connections were much lesser due to removal of LCD which reduced the debugging task of checking too many connections.

Another indirect learning due to this project was better understanding of FAT file systems. Since the values can't be stored in the SD Card directly, a FAT files system needs to be created and store the values in a file so that data can be interpreted.

Challenges

The toughest challenge faced in the project was calculation and calibration of Current sensed by the sensor. Basically at 0A 0.5Vcc is given as output by the sensor. Now the RMS of the current is calculated and multiplied by a coefficient (approx 5) to give Power. So even with the error of 1 whether in negative or positive side, due to square and averaging the error does not cancel out and since it multiplies by a factor of around 5 error s up to 5-10 Watts are normally seen. This problem could not be solved as the Sensor itself has a error percentage of 1.5%.

Initially when LCD was being used for debugging purposes, for calculation of RMS accurately the signal had to be sampled according to the frequency of the power supply and ADC and since LCD requires some time delays to display the value, incorrect values were coming out. Because of this Uart interface was used for debugging.

Another problem faced was which approach should be followed for storing data in the SD card whether to store in FAT files system format so that it is easier to read by other devices or to just use the native interface and put the data and leave the interpretation of data to other devices. In the end FAT files system was used.

References

1. SD Card libraries:

http://www.roland-riegel.de/sd-reader/

2. UART open source libraries:
http://read.pudn.com/downloads157/sourcecode/embed/697054/AVR_SDcard_FAT32/uart.h__.htm

3. Sensor calibration Idea was taken from this thread:
http://8515.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&p=619829

4. A lot of doubts in concept and implementation were addressed by Pandarasamy Sir.