

Experimental Evaluation of the MSP430 Microcontroller Power Requirements

Karel Dudacek *, Vlastimil Vavricka *

* University of West Bohemia, Dept. Computer Science and Engineering
Univerzitni 8, Pilsen, Czech Republic

Abstract—In battery operated electronic devices, the energy required to execute given tasks set is one of very important parameters. Modern microcontrollers design provides a number of methods to reduce microcontroller power requirements depending on the microcontroller load. In this paper we present experimental measurement of energy consumption, measured on the Texas Instruments MSP430 microcontroller. Conclusions important for the energy consumption minimization are presented as well.

Keywords—microcontroller, power requirements.

I. INTRODUCTION

Many modern microcontroller (MCU) controlled electronic devices are required to be operated from battery power, without possibility of using outlet AC adaptor or any other energy source. The time the electronic device can operate between battery changes is often one of the most important parameters. For some devices like thermostats, flowmeters, remote controllers etc. the required battery change period can be one or more years. To achieve this goal, microcontroller manufacturers offer devices with a number of power saving modes. The system designer task is to use this possibilities to minimize the system power requirements, ie. to maximize the battery life.

Generally, the CMOS processor power requirement P can be evaluated by the formula

$$P = C_e \cdot V_{cc}^2 \cdot f + I_L \cdot V_{cc}, \quad (1)$$

where C_e is the effective switching capacitance (can be considered to be a constant), V_{CC} denotes supply voltage, I_L is leakage current and f is working (clock) frequency. As the number of clock pulses needed for some specified task execution is a constant, the way to reduce microcontroller power is by reducing supply voltage V_{CC} . The supply voltage can be reduced to the level that guarantee processor operation on frequency required for all tasks to meet their deadlines. This method is known as Dynamic Voltage Scaling.

The Dynamic Voltage Scaling (DVS) method has been studied in many works ([1], [2]). However, a lot of those works consider simplified processor power requirements model. So that processor power requirements in true real time application can be more complex ([3]), we decided to experimentally investigate this problem in more details. Measurement on the true processor will consider influence of all devices' functional blocks (clock generator, internal memory etc.). The goal of our experiments was to find optimal method for microcontroller frequency and supply

voltage scheduling in order to meet specified performance requirements with minimal energy consumption.

II. THE MSP430 MICROCONTROLLER

For our experiments we choose the Texas Instruments MSP430 microcontroller. This device is designed for low power applications (see [4] and [5]). It can be use with three internal and/or external clock sources with dedicated prescalers (dividers) for processor core and for on-chip peripheral subsystems:

- XT1 – Low frequency oscillator can be used in low frequency or high frequency mode. In the low frequency mode, the 32.768 kHz crystal can be used to provide slow clock frequency. In the high frequency mode the maximal frequency can be 8 MHz.
- XT2 – High frequency oscillator. Can be used in 450 kHz to 8 MHz frequency range.
- DCO – Digitally controlled RC type oscillator. Frequency of this on-chip oscillator can be digitally controlled by writing appropriate values to control registers. Frequency range of this oscillator is in the range from approximately 400 kHz to 4.5 MHz. With external resistor the frequency can be set to max. 8 MHz.

All three oscillators can be switched on or off by writing appropriate values to the control registers.

Simplified block diagram of MSP430's clock circuits is on the Fig. 1.

III. MCU POWER MEASUREMENTS

For energy-optimal MCU voltage and clock frequency planning, the power characteristics of specific device should be known. First, we investigated the MSP430 power requirements in run and idle states. The HF oscillator was turned on to generate the clock signal for

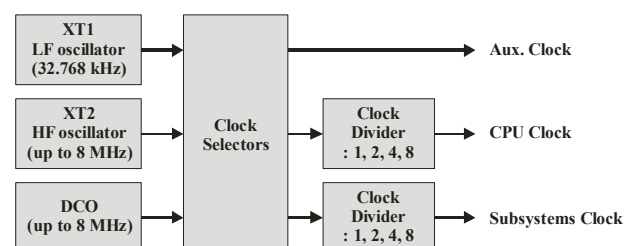


Fig. 1. MSP430 clock system.

CPU core. The DCO and LF oscillators were turned off.

In the run condition, the MCU core clock frequency was changed by changing HF oscillator frequency, while CPU clock divider was set to 1:1 ratio. All other on-chip peripheral modules were switched off. Supply voltage was changed in the whole MCU working range (from 1.8 V to 3.6 V). The CPU executed task with mix of instructions that used both register and internal memory operands.

The power curves measured in run condition are on Fig. 2 and Fig. 3. Both curves show good correspondence with theoretical assumption given by formula (1). While MCU power dependency on the clock frequency is linear, the power dependency on supply voltage has the obvious quadratic shape (see Fig. 2).

For the idle state measurement the CPU clock switched off in the clock selectors block (see Fig. 1), while HF oscillator was still running. As we wanted to investigate the MCU power requirements from the practical point of view, we decided to make conditions more realistic. Therefore the LF oscillator was running at 32.768 kHz and sourcing internal Timer_A. (Permanently running hardware timer is required in RT operating systems to

provide timer ticks for the real time base.)

Idle mode power consumption diagrams are shown on Fig 4 and Fig. 5. While the supply voltage – power dependency corresponds to equation (1), the frequency – power dependency show slow power increase on high frequencies. This can be important in situations, where the CPU core is in idle state for the most time.

To save more energy the HF oscillator can be switched off if the CPU core is in the idle state. The basic disadvantage of this method is long oscillator starting time, causing big latency when the CPU is woken up to service internal or external interrupt. To avoid this problem, the MSP430 has the DCO (Digitally Controlled Oscillator) with starting delay as short as 6 μ s. DCO can be used to provide clock pulses for CPU core as well as for other MCU subsystems.

Power characteristics of the CPU in run state with DCO clock source are on Fig. 6 and Fig. 7.

If we compare power diagrams for MCU running with high frequency XT2 oscillator (Fig. 3) with diagrams for DCO oscillator as the clock source (Fig. 6), we note

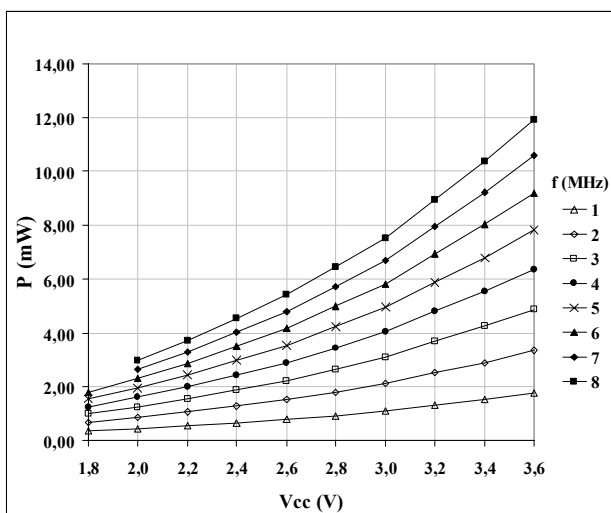


Fig. 2. Power – Voltage curves, run state.

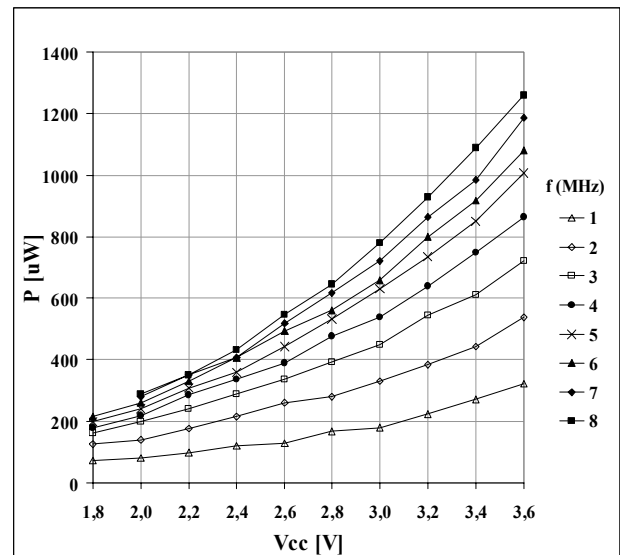


Fig. 4. Power – Voltage curves, idle state.

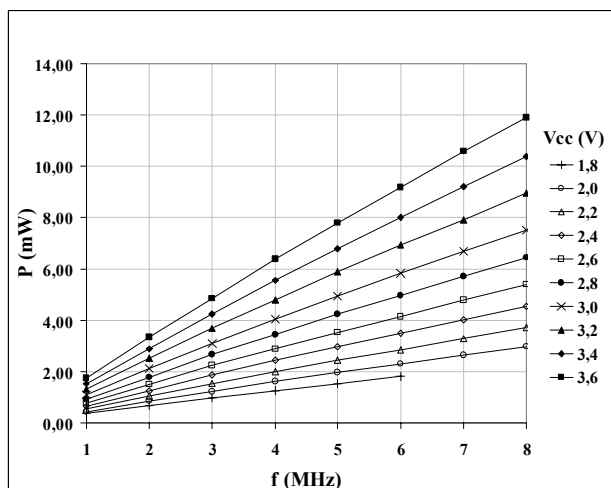


Fig. 3. Power – Frequency curves, run state.

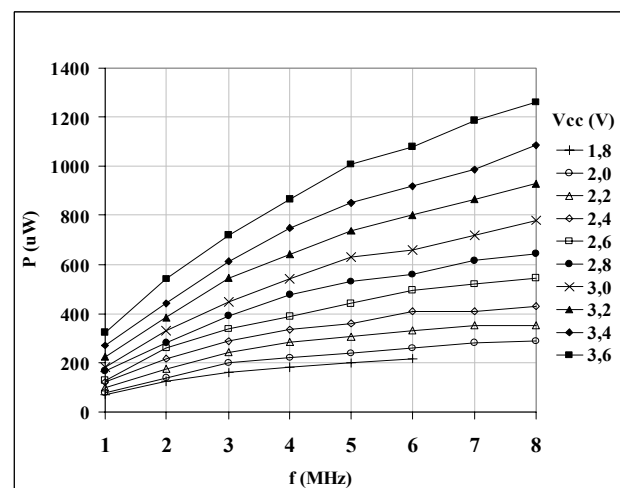


Fig. 5. Power – Frequency curves, idle state.

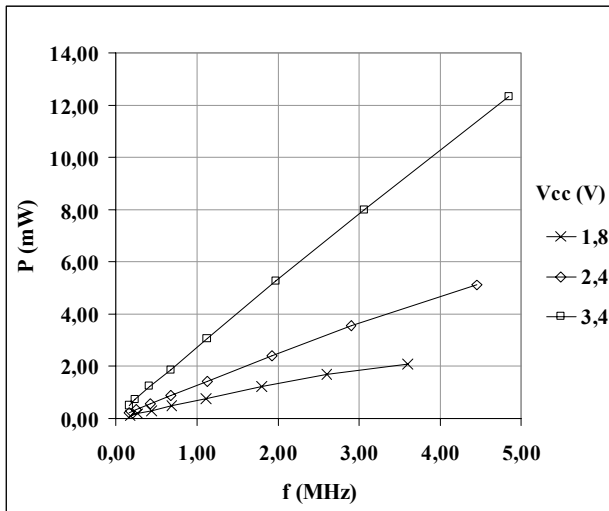


Fig. 6. Power – Frequency curves, run state, DCO oscillator.

markedly higher power for DCO. At the frequency set to 4 MHz and the supply voltage 2.4 V the XT2 clocked MCU consumed 2.9 mW while at the same frequency and supply voltage the DCO clocked MCU consumed 4.7 mW.

MCU power requirements for DCO and HF oscillator clock sources are compared on the Fig 8. When the CPU uses DCO as clock source, the power consumption is approximately 1.6 times higher in comparison to HF oscillator clock source. To avoid this disadvantage, more sophisticated wake up algorithm can be considered: After waking up, both HF and DCO oscillators are restarted. After 6 μ s CPU can run the program, using DCO clock source. After HF oscillator stabilizes, CPU clock source is switched to it and DCO is stopped.

When using DCO for MCU clock sourcing, some properties of this oscillator type should be considered. As the DCO uses RC type oscillator, the frequency has poor stability if the temperature changes. Moreover, the frequency varies if the MCU supply voltage changes.

IV. CLOCK DIVIDERS

For the system where fixed set of tasks is executed, there is no problem to find worst case processor load and to specify clock frequency required to execute all tasks in

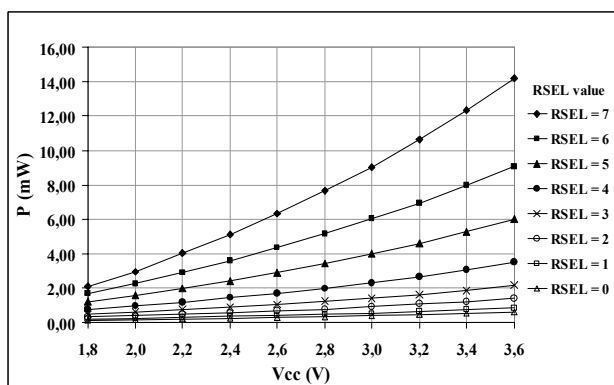


Fig. 7. Power – Voltage curves, run state, DCO oscillator.

their deadlines. If the clock frequency is known, supply voltage can be set to minimal possible value in order to achieve maximal energy conservation. However, if the processor load varies and a dynamic task planning is used, the problem is more complex. The CPU clock frequency has to be changed dynamically to tune the CPU performance to varied load.

For this purpose the MSP430 clock generator contains programmable frequency divider. In this circuit the oscillator frequency can be divided by software selectable ratio. The divider ratio can be set to 1, 2, 4 or 8, slowing the CPU clocks appropriately.

If the CPU clocks are slowed by the clock divider, the oscillator is still running on high frequency, increasing total power of the microcontroller. In our work we did some measurement to investigate this effect.

The microcontroller executed sample code on various frequencies. Each clock frequency was set by all possible combinations of crystal oscillator frequency and clock divider ratio. For example, the clock frequency 1 MHz was set using 1 MHz oscillator and the clock divider 1, then using 2 MHz oscillator and the clock divider 2 etc. Note that both minimum and maximum oscillator frequencies are limited, so some clock divider ratio could not be used. Microcontroller power was measured for all applicable frequency settings.

Results of this experiment are shown on the Fig. 9.

V. CLOCK FREQUENCY OPTIMIZATION

For execution of some task, the specific number of CPU clock pulses is required. As the energy consumed in one clock period is a constant (supposing fixed supply voltage), the total energy for the task execution does not depend on the CPU clock frequency. However, the higher clock frequency is used, the higher supply voltage is required and vice versa (in boundaries specified in the MCU datasheet).

The MSP430 can be run with supply in the range 1.8 to 3.6 V. For each clock frequency the minimal required supply voltage is specified in the datasheet. MSP430 power consumption for a number of frequencies – minimal voltage sets is presented in Table I and on Fig 10.

In RT operating systems DVS method can be used for energy efficient planning. The goal is to find the slowest

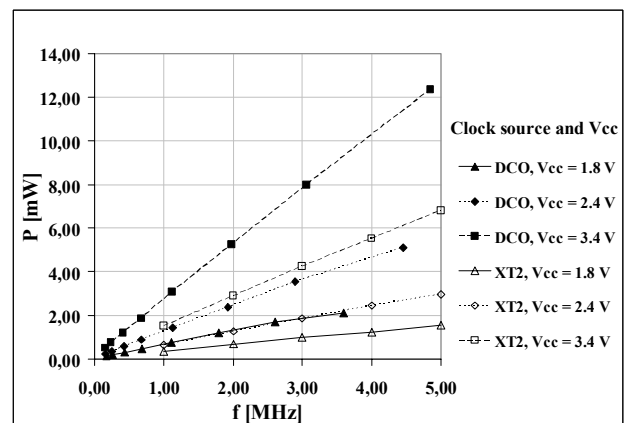


Fig. 8. DCO and XT2 power curves, run state.

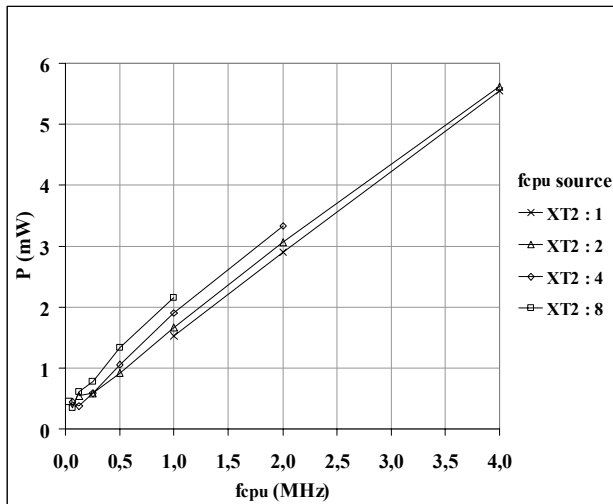


Fig.9. Power for various clock divider ratios.

possible processor clock frequency which guarantees that all tasks loaded in the system can be executed before its deadlines. As we can use MCU supply voltage as low as possible for this frequency, we can achieve remarkable power conservation (more in [1] and [2]).

To verify this paradigm on the MSP430, we used three tasks (Task_1, Task_2 and Task_3) that executed mix of instructions. Tasks were started periodically in 100 ms period. Task starting interrupt was generated by on-chip

TABLE I.

CPU frequency [MHz]	Minimal Vcc [V]	Icc [mA]	P [mW]
1	1,80	0,20	0,36
2	1,80	0,38	0,68
3	1,80	0,55	0,99
4	1,80	0,70	1,26
5	2,20	1,11	2,44
6	2,66	1,68	4,47
7	3,13	2,38	7,45
8	3,60	3,31	11,92

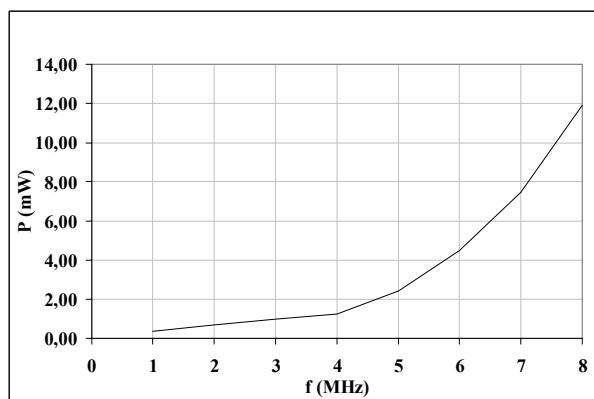


Fig. 10. Power – Frequency curves, optimal supply voltage.

timer Timer_A, clocked by low frequency oscillator XT1. In the active part of the task period (ie. task execution) CPU was clocked by high frequency oscillator XT2. In the idle part of the task period the processor was stopped by switching CPU clock off, while the crystal oscillator XT2 and the low frequency oscillator XT1 were still running (see Fig. 11).

For each task we did a set of measurements, changing CPU frequency from the lowest possible value (giving 100% CPU load) to the maximal frequency 8 MHz. For each frequency the supply voltage was set to the minimal possible value according to the MSP430 datasheet (see also TABLE I). The total energy consumption in one task period was measured. Energy consumption curves are shown on the Fig. 12. The theoretical curves on this diagram were computed using separate MCU power measurements in idle and active modes as described in section above. For all three tasks, the minimum power consumption was achieved for frequencies less or equal to 4 MHz. The consumption curves have flat or only slightly rising shape in this frequency range. Although the active periods had various lengths depending on the clock frequency, the number of clock pulses needed to execute the task in the active state was the same. Because the supply voltage for frequency less or equal to 4 MHz had to be set to 1.8 V, the energy consumed in one task period was the same for all of these frequencies. Small energy consumption increase observed in Task_1 was probably caused by consumption of the clock generator and other clock circuits.

Clock frequencies higher than 4 MHz required higher supply voltage, causing remarkable increase of energy consumption.

VI. CONCLUSIONS

Results of our experiments showed that dynamic voltage scaling method is well applicable on the MSP430 microcontroller. As the minimal supply voltage 1.8 V guarantee the MCU operability with clock frequency 4 MHz, from this point of view there is no necessity to use lower frequency as the total energy consumption will not decrease.

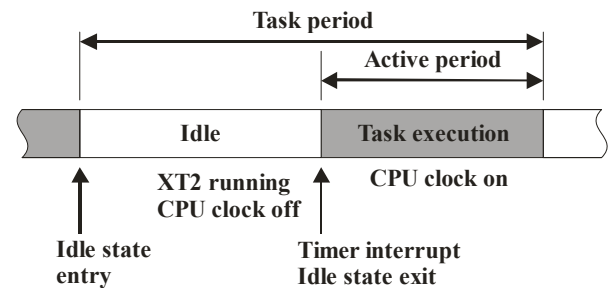


Fig. 11. The task period.

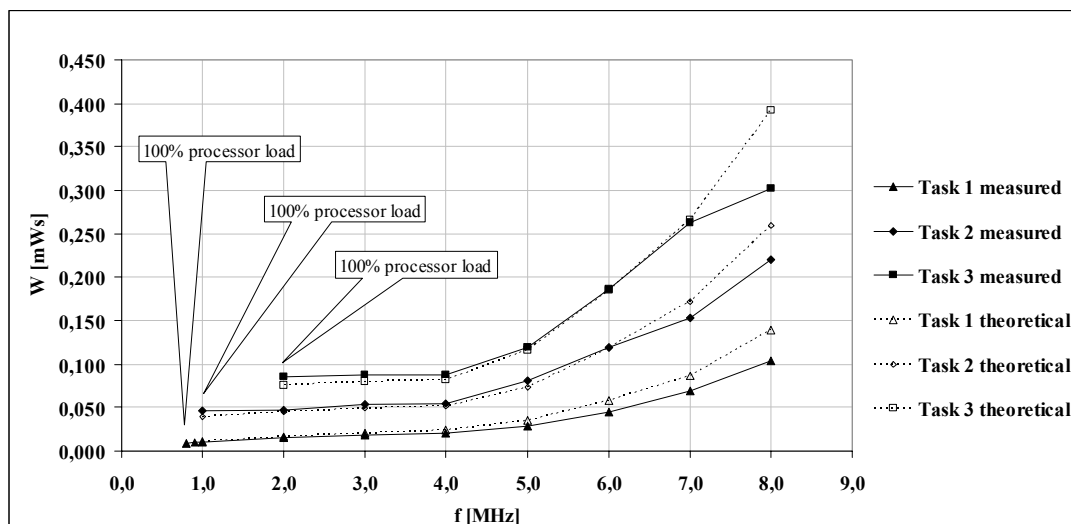


Fig. 12. Energy consumed in one task period.

For application of the DVS method with dynamic task planning, one more fact should be considered: To set the supply voltage to the level corresponding to the CPU operating frequency, the system has to be equipped with device (for example DC/DC converter) for supply voltage adjustment (see Fig. 13). Such devices are available on the market, but as the MSP430 is designed for single chip solutions and tiny systems, there is the question about overall effectiveness of such solution.

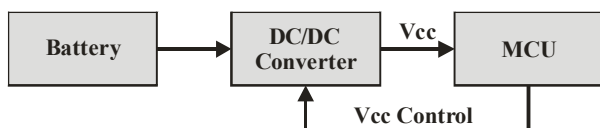


Fig. 13. Supply system for DVS application.

To save the energy in the CPU idle period, both HF oscillator and DCO can be turned off. The MSP430 has capability to restart DCO automatically if some interrupt occurred. Fast DCO restart adds only about 6 μ s to the interrupt latency time. However the power requirement of DCO clocked CPU are nearly twice the power requirement of HF oscillator clocked CPU. The appropriate procedure is to restart both oscillators simultaneously. After HF oscillator stabilizes, the software can set it as the clock source and turn the DCO off consecutively.

Many modern microcontrollers are designed for low power applications. We believe that our paper showed that detailed study of these microcontroller properties is

necessary if power saving modes have to be used with the maximal effect.

VII. FUTURE WORK

Measurements presented in this paper were done in the simple software loops, without any operating system. In the future work we will focus to the power requirements of microcontroller running tasks in the RTOS environment. The goal of our work will be to find rules for energy optimal task planning and clock system adjusting.

ACKNOWLEDGMENT

The work was granted by the Ministry of Education, Youth and Sport of the Czech Republic - "University spec. research - 1311".

REFERENCES

- [1] J. Jejurikar, R. Gupta, "Optimized slowdown in Real/Time Task Systems," *Proceedings of the 12th Euromicro Conference on Real/Time Systems ECRTS 2004*, pp. 155 – 164, 2004.
- [2] H.-S. Yun, J. Kim, "On Energy-Optimal Voltage Scheduling for Fixed-Priority Hard Real-Time Systems," *ACM Transactions on Embedded Computing Systems*, 2(3): pp 393.- 430, 2003.
- [3] D. C. Snowdon, S. Ruocco, and G. Heiser, "Power Management and Dynamic Voltage Scaling: Myths and Facts," *Proceedings of the 2005 Workshop on Power Aware Real-time Computing*, pp. 44 - 49. New Jersey, USA, September 2005.
- [4] "MSP430x1xx Family User's Guide," Texas Instruments Inc. 2003.
- [5] "MSP430x13xx, MSP430x14xx: Mixed Signal Microcontroller," Texas Instruments Inc. 2004.