Demo: A Node's Life – Increasing WSN Lifetime by Dynamic Voltage Scaling

Ulf Kulau, Wolf-Bastian Pöttner, Felix Büsching and Lars Wolf
Institute of Operating Systems and Computer Networks
Technische Universität Braunschweig
Braunschweig, Germany
Email: [kulau|poettner|buesching|wolf]@ibr.cs.tu-bs.de

***Abstract*—In mobile applications, like wireless sensor networks, it is often inevitable to use a location-independent source of energy. Wherever the batteries capacity or inefficient energy harvesting limits the lifetime, other solutions have to be implemented to increase the energy efficiency.**

Dynamic Voltage Scaling (DVS) is a well known technique with the ability to adapt the voltage level to the actual system load to save energy. While it has been used in many application areas, DVS has not been studied sufficiently for wireless sensor nodes. In this demo, we show the benefits of DVS. We were able to optimize DVS for an 8-Bit ULP micro controller by a combination of DVS and dynamic power management (DPM). Moreover, we analyzed the effect of voltage scaling on other components of the sensor node (transceiver, sensors and memory).

***Index Terms*—wireless sensor node, energy efficiency, voltage scaling, hardware architecture, hardware design, dvs.**

I. Introduction

In Wireless Sensor Networks (WSNs), usually batteries are used for energy supply to enable flexible placement of sensor nodes. Unfortunately, due to the limited capacity of batteries $\left(\frac{Wh}{Kg}\right)$, the lifetime of the WSN is limited, too. Thus, to increase the uptime of a wireless sensor node and the overall network, the energy efficiency of each node has to be improved. Especially whenever the used energy source is not able to meet desired lifetimes of specific nodes.

In [1] the detailed motivation, implementation and evaluation for this work is given.

In this demonstration we present the practical implementation of DVS on a sensor node.

II. Implementation

In [1] we have shown, that DVS will improve the energy efficiency of the processing unit. To implement a combination of DPM and DVS (modDVS) on a ULP micro controller, we need to be able to control the clock rate and the voltage level by software.

A. Voltage scaling module

The processing unit is not the only component of a sensor node. Whenever the transceiver, sensors or external memories require a higher voltage level than the micro controller, the potential of DVS is wasted. Hence, one simple adjustable voltage regulator which supplies all components is not suitable for a wireless sensor node.

The digital interface is implemented through a I^2C 8-Bit digital potentiometer [2] with two variable ohmic resistors. Every resistor controls an LDO [3] to realize two independent voltage paths. The linear voltage regulators generate an internal reference voltage. This allows a generic description of the output voltage level as a function of a resistance ratio. Thereby, the software implementation can calculate register settings for the digital potentiometer to achieve a given voltage level. To validate the accuracy, a target voltage level between $1800mV \leq V_{core}, V_{per} \leq 3300mV$ with a step size of $\Delta V = 100mV$ were set by software. The average correlation between all expected and measured voltage levels is ≥ 0.999 for both voltage paths V_{core} and V_{per}.

Figure 2 shows the implementation of the voltage scaling module. The straightforward design allows it to equip other existing nodes with an active voltage scaling. As the digital potentiometer comes with additional digital output pins, one pin is used to either enable or disable the peripheral voltage V_{per} by software.

B. Clock rate adjustment via software

The concrete method to adapt the clock rate depends on the micro controller and clock source. The ATmega1284p micro controller in our prototype generates the main clock with an internal RC oscillator [4]. As every ATmega1284p has a unique calibration register *OSCCAL*, the claimed clock rate f_{claim} has to be calibrated with an external reference clock f_{ref}. This reference clock offers a known and constant

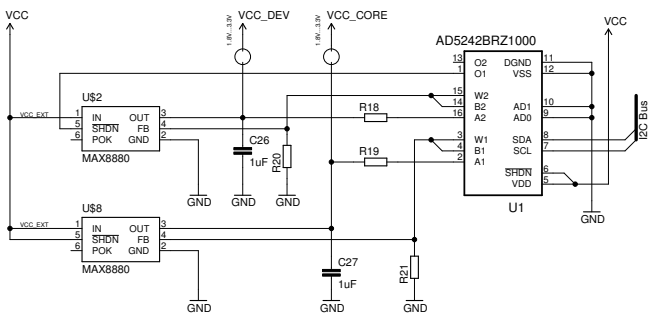

Figure 2. Schematic of the I^2C voltage scaling module

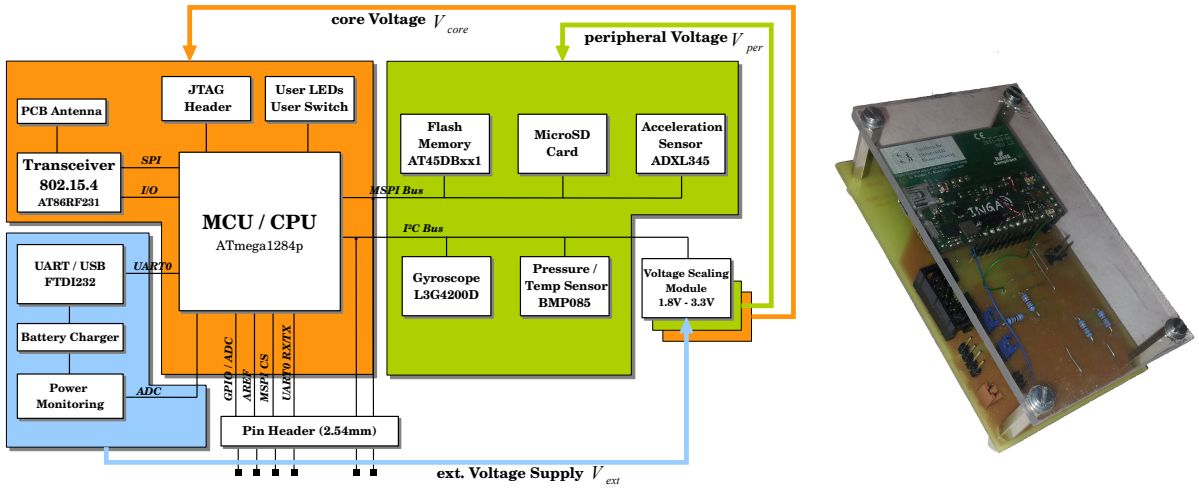

Figure 1. Block diagram of the system integration and picture of the prototype implementation

time slot $T_{ref} = \frac{1}{f_{ref}}$. During one period of T_{ref}, the processor ticks are counted and added up to $\rho = T_{ref} \cdot f_{cpu}$. If $\rho < T_{ref} \cdot f_{claim}$ the clock rate of the micro controller is too low and *OSCCAL* has to be increased. Otherwise, *OSCCAL* is decreased if $\rho > T_{ref} \cdot f_{claim}$. Instead of a iterative clock adjustment, we implement a binary search algorithm which finishes in $O(log(n))$. Nevertheless a single iteration lasts a few milliseconds, because the reference clock is generated by an external crystal with $f_{ref} = 32768Hz$. We conclude that it is feasible to adjust the internal clock rate of the ATmega1284p in software, but only with a significant time overhead. An improvement would be a lookup table, where previously calibrated register values are stored, so that some dedicated clock rates are available in a few processor ticks.

C. System Integration

The prototype implementation is based on the INGA sensor node [5], which is an open hardware project[1]. The modified architecture is shown in figure 1. A challenge with different voltage levels on one sensor node are the communication interfaces like I^2C and SPI bus. The peripherals are interfaced with V_{core} but are supplied with V_{per}. The I^2C components are very tolerant with unequal interface and supply voltages, so only minor changes were required. The SPI bus in contrast was equipped with a level translator [6], which acts like a bridge between the voltage levels. As the digital potentiometer of the voltage scaling module is supplied with the external battery voltage (V_{ext}) but interfaced by the micro controller (V_{core}), the electrical potential between V_{ext} and V_{core} might harm the components, so that another level translator was integrated. The voltage translators need a supply current of $I_{cc} \approx 130\mu A$ each, which causes additional overhead. As a voltage conversion is only needed during an SPI or I^2C transaction, the voltage translators are put into standby mode for idle times to reduce the overhead to less than $2\mu A$ [6].

[1]INGA is an ongoing open source project. All resources can be downloaded at http://www.ibr.cs.tu-bs.de/projects/inga/

III. Demonstration

Voltage and current of two wireless sensor nodes are monitored and displayed continuously. While one node is just an original INGA [5] sensor node, powered through a 3,3 Volt linear dropout converter (LDO), we have modified another INGA sensor node to be capable of DVS and DPM (see Figure 1). These nodes perform the same tasks and run the same duty-cycle schedules and, by that, can be directly compared regarding their individual current consumption. Additionally, the normal function of the nodes can be observed via an PC, displaying the nodes measured sensor data.

It can be seen that using DVS and DPM (modDVS) leads to a significantly lowered power consumption of a node, while all nodes are working correctly. In a temperature sensing scenario a node would live ≈ 2.7 times longer when using modDVS instead of a fixed voltage supply.

References

[1] U. Kulau, F. Büsching, and L. Wolf, "A Node's Life: Increasing WSN Lifetime by Dynamic Voltage Scaling," in *The 9th IEEE International Conference on Distributed Computing in Sensor Systems 2013 (IEEE DCoSS 2013)*, Cambridge, USA, May 2013, accepted for publication.

[2] Analog Devices, "I2c-compatible, 256-position digital potentiometers," 2012. [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD5241_5242.pdf

[3] Maxim, "12v, ultra-low-iq, low-dropout linear regulators with pok," 2000. [Online]. Available: http://www.datasheetcatalog.org/datasheet/maxim/MAX8880-MAX8881.pdf

[4] Atmel, "Atmega164a/164pa/324a/324pa/644a/ 644pa/1284/1284p," 2010. [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/doc8272.pdf

[5] F. Büsching, U. Kulau, and L. Wolf, "Architecture and Evaluation of INGA - An Inexpensive Node for General Applications," in *Sensors, 2012 IEEE*. Taipei, Taiwan: IEEE, oct. 2012, pp. 842–845.

[6] Maxim, "15kv esd-protected, 1μa, 16mbps, dual/quad low-voltage level translators in ucsp," 2007. [Online]. Available: http://datasheets.maxim-ic.com/en/ds/MAX3372E-MAX3393E.pdf