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

Ulf Kulau, Felix Büsching and Lars Wolf
Institute of Operating Systems and Computer Networks
Technische Universität Braunschweig
Braunschweig, Germany
Email: [kulau|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 paper, we present a method to estimate the gain of DVS. We study DVS theoretically as well as in a practical manner, i.e., we using a prototype implementation of a DVS capable wireless sensor node. With the theoretical considerations we were able to optimize DVS for an 8-Bit ULP micro controller by a combination of DVS and dynamic power management (DPM). In addition, we proved the calculated results with an extensive practical evaluation of the concept using a prototypical implementation. Moreover, we analysed 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 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.

A simplified model for the power consumption of a sensor node (P_{node}) is given in equation 1.

$$\begin{aligned} P_i &= \frac{1}{T}\left(T_{i_{active}} P_{i_{active}} + T_{i_{sleep}} P_{i_{sleep}}\right) \\ \Rightarrow P_{node} &= \sum_i P_i \end{aligned} \tag{1}$$

The node is divided into i units (processing unit, transceiver unit, sensors, ...) with two power states each: An active state and a sleep state with a much smaller power consumption $P_{sleep} \ll P_{active}$. Furthermore, during a time T a component i can switch between these two states. While $T_{i_{active}}$ is the total time that a component spends in active mode, $T_{i_{sleep}}$ is the total time in sleep state ($T = T_{i_{active}} + T_{i_{sleep}} \ \forall i$). All in all P_{node} is the sum of the components power consumption.

Both the particular power consumption $P_{i_{active}}$ and the time that a component spends in active state T_{active} could be reduced, to minimize P_{node} in total.

As the transmission, respectively the reception of data is one of the most power consuming processes, a lot of existing approaches deal with an intelligent usage of the transceiver unit. Methods on protocol level [1], [2], [3] avoid long active states (reducing T_{active}), whereas the adaptation of the transmission power to the actual channel characteristics [4] influences P_{active} during this active time.

The processing unit has a relatively long active time (T_{active}), because it executes the software which is responsible for all other units of the sensor node. Typically, ultra low power (ULP) micro controllers provide a dynamic power management (DPM) in the form of several sleep states, to reduce the power consumption.

A more fundamental approach to save energy is based on the specific behaviour of the internal structure of integrated circuits. If we take a closer look on the typical hardware architecture of wireless sensor nodes, or other embedded ULP micro controller systems, we see that most of the used integrated circuits are realized in CMOS (Complementary Metal–Oxide Semiconductor) technology. The advantage of this technology is its theoretical non existing power consumption during static operation. An electrical current flow only occurs whenever the internal state of a CMOS gate is changed, so that the dynamic power consumption (P_{dyn}) dominates the total with an percentage of $\approx 75\%$. On the other hand the dynamic power consumption has a linear dependency on the clock rate (f_{cpu}) and a quadratic dependency on the voltage level (V). The parasitic capacity C_L depends on the quality of the manufacturing process [5].

$$P_{dyn} \quad = \quad C_L \cdot f_{cpu} \cdot V^2 \tag{2}$$

The switching delay of a CMOS gate grows with lower supply voltages V. To guarantee that a switching operation finishes within period $\frac{1}{f_{cpu}}$, a clock rate is assigned to a specific voltage level $V(f_{cpu})$.

$$V(f_{cpu}) \Leftrightarrow f_{cpu}(V) \tag{3}$$

This relationship is used in methods like dynamic voltage scaling (DVS), where the processing power of a processor is adapted to the actual workload. A higher clock rate needs a higher voltage level and contrariwise. Current 8 and 16 bit processing units in the area of wireless sensor nodes do not support DVS functionalities. Despite the fact that up-to-date ARM based micro controllers (e.g. [6]) may be equipped with a voltage scaling feature, this is only limited to the processor, disregarding all other parts and peripherals of the node. Thus, the effect of a scaled down voltage is locally bounded and other peripherals would not profit.

In this paper we present a theoretical yield estimation as well as a practical implementation of DVS on a sensor node. Through extensive evaluations we show the advantages of this approach and the achievable gains.

The outline of the paper is as follows: Related work is discussed in the next section. Section III presents a theoretical approach to design and optimize a DVS capable node architecture. Furthermore we show that the combination of DVS and DPM increases the efficiency of small ULP micro controllers. The implementation of a prototype, which is based on a conventional sensor node, is described in section IV. Beside the correlation between the preliminary considerations and the real-life behaviour, an extensive evaluation in section V shows the effect of voltage scaling on all parts of the sensor node, such as RF communication, sensors and memories.

II. Related Work

The absolute minimum voltage level for a wireless sensor node to be still usable depends on the highest minimum voltage level of all components. That means with regard to the equations 1 and 2, a lot of energy is wasted because of only one component. A scalable voltage supply offers the potential to reduce the overall power consumption of a sensor node by adjusting the voltage to the needed level. To support a user defined voltage level, additional hardware components are necessary. These components come with the overhead of a static quiescent current I_q and the non-ideal efficiency of voltage conversion.

A passive voltage scaling (PVS) approach is proposed in [7]. This variant tries to maximize the average MIPS (Million Instructions Per Second) without any hardware modification of the sensor node. The node is supplied by battery directly and according to equation 3 the clock rate is set as a function of the batteries discharge curve $f_{cpu}(V_{batt})$. The advantages are no additional quiescent current and an efficiency of $\eta \approx 100\%$. However, PVS is not able to adapt the voltage level to the workload independent of the battery state.

To handle this problem of PVS, active voltage scaling can be used. The component aware dynamic voltage scaling (CADVS) [8] is based on an energy efficient step down converter. Compared to linear low dropout regulators (LDO), the step down converters have a higher conversion efficiency (e.g. up to $\eta_{sd} = 96\%$ [9]). Nevertheless I_q of a step down converter is much higher than the quiescent current of a LDO. Hence, we have the difficult situation, that the choice of the voltage converter depends on the later applications. For short duty cycles a LDO would be more efficient, which results from the lower quiescent current. A more detailed view on the efficiency of voltage converters is given in section IV-A.

For the required interface to control the voltage level, [8] proposes a digital potentiometer. All in all, the authors claim that CADVS saves up to 31.5% of the energy compared to a constant voltage supply.

The sensor node architecture in [10] is equipped with a fully integrated power management IC. Those fully integrated power management units, such as [11], offer all features which are needed to implement DVS on a sensor node. In [10] the usage of [11] is justified, because the considered processing unit is a powerful OMAP processor [12] to allow software defined radios. For typical sensor nodes with 8 or 16-Bit ULP micro controllers, a power management IC is not suitable due to its high quiescent current

Dynamic voltage scaling on wireless sensor nodes has also been proposed in [13], [14], [15]. But they all have in common that the architectures are only design proposals and the expected power savings are only based on calculations. Furthermore, the authors focus on the processing unit and do not consider effects on the RF communication or sensing units. As far as we know, our work is the first to study not only theoretical issues, but implement and evaluate DVS on a wireless sensor node including a practical analysis of the effects on various components.

III. Theoretical Considerations

An advantage when starting with a theoretical approach is, that optimum operation points can be indicated and proved.

A. Model Function

Unfortunately a datasheet of a micro controller usually does not provide much information about the relation between the current consumption I_{cc} and the voltage supply V respectively the clock rate f_{cpu}. Mostly, only a few data points are listed or measured characteristic curves are given. However, this information is sufficient to derive a model function and approximate the current consumption $I_{cc}(f_{cpu}, V)$ of a micro controller.

With equation 2 and an undefined factor $\varphi \widehat{=} C_L$ we can assume:

$$P_{dyn} \approx I_{cc} \cdot V \;\Rightarrow\; I_{cc}(f_{cpu}, V) = \varphi \cdot f_{cpu} \cdot V \tag{4}$$

To increase the accuracy, the model function can be divided into piecewise linear functions.

1) Define $i+1$ nodes at fixed frequencies f_{cpu_i} and derive functions $I_{cc}(f_{cpu_i}, V)$ by using curve fitting.
2) Use the equation of a line $y(x) = a \cdot x + b$ to model the frequency dependence
   $I_{cc}(f_{cpu}, V) \;\forall\; f_{cpu_i} \leq f_{cpu} \leq f_{cpu_{i+1}}$

$$
\begin{aligned}
a_i(V) &= \frac{\Delta I_{cc}(V)}{\Delta f_{cpu}} = \frac{I_{cc}(f_{cpu_{i+1}}, V) - I_{cc}(f_{cpu_i}, V)}{f_{cpu_{i+1}} - f_{cpu_i}} \\
b_i(V) &= I_{cc}(f_{cpu_i}, V) - a_i(V) \cdot f_{cpu_i}
\end{aligned}
$$

$$
\Rightarrow I_{cc}(f_{cpu}, V) = \begin{cases} a_1(V) \cdot f_{cpu} + b_1(V) & \forall f_{cpu_1} \leq f \leq f_{cpu_2} \\ a_2(V) \cdot f_{cpu} + b_2(V) & \forall f_{cpu_2} < f \leq f_{cpu_3} \\ \vdots & \\ a_i(V) \cdot f_{cpu} + b_i(V) & \forall f_{cpu_i} < f \leq f_{cpu_{i+1}} \end{cases}
$$

Our implementation is based on a Atmega1284p [16], so we derived a model function for this micro controller as described before. We took only 3 nodes at fixed frequencies $f_{cpu} = \{1\text{MHz}, 4\text{MHz}, 8\text{MHz}\}$ which results in two piecewise linear functions.

With the same method we created a linear function for equation 3 ($V(f_{cpu})$). Thereby the current consumption of the Atmega1284p is given as a function of the clock rate $I_{cc}(f_{cpu}, V(f_{cpu}))$.

B. Yield estimation

With the model function it is possible to estimate the yield of a variable supply voltage compared to a fixed one. The yield is a factor of how much the current consumption can be reduced, when DVS is used instead of DPM. With DVS, although the processing will take more time due to the lower clock rate, a lower current consumption can be achieved because of the lower voltage. We assume a scalable clock rate, so that f_{cpu} can be adapted for a given workload λ. Furthermore, we define a maximum clock rate of $f_{cpu_{max}} = 8\text{MHz}$, because this is the default clock rate of the Atmega1284p. From this it follows that a process which needs permanent $f_{cpu} = 8\text{MHz}$ to meet a deadline, implies a system load λ of 100% ($\lambda = 1$). For example, a system load of 50% equates $f_{cpu} = 4\text{MHz}$.

Usually in the area of small 8-Bit micro controllers, dynamic frequency and voltage scaling is not provided. The conventional way to save energy is to use DPM as described in section I. The average current consumption of DPM according to the system load is given in the following equation 5 (I_{sleep} is the current consumption during sleep state):

$$
I_{DPM}(\lambda) = \lambda \cdot I_{cc}(f_{cpu_{max}}, V(f_{cpu_{max}})) + (1 - \lambda) \cdot I_{sleep} \quad (5)
$$

Contrary to DPM, equation 6 shows the current consumption when the clock rate and the voltage level is adapted to the workload (DVS).

$$
I_{DVS}(\lambda) = I_{cc}\left(\lambda \cdot f_{cpu_{max}}, V(\lambda \cdot f_{cpu_{max}})\right) \quad (6)
$$

Figure 1 shows the yield $Y(\lambda) = \frac{I_{DPM}}{I_{DVS}}$ for dynamic voltage scaling ($Y(\lambda)DVS$). The yield's peak is at $\lambda \approx 50\%$, because for the Atmega1284p the minimum supply voltage supply is reached for clock rates $f_{cpu} \leq 4\text{MHz}$. To improve the yield for workloads $< 50\%$, classical DPM can be used. In this paper the combination of DPM and DVS is named modDVS. For

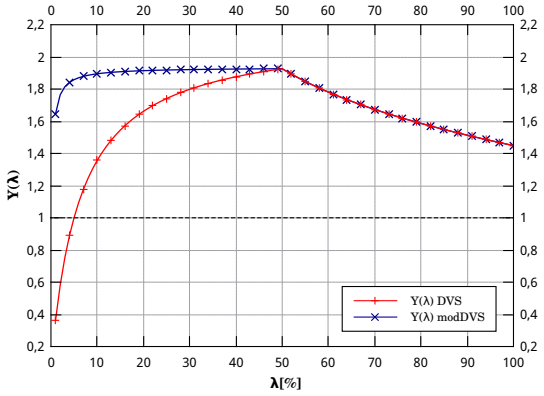


Fig. 1. Estimated yield of a scalable frequency and voltage mechanism as a function of the system load

that the clock rate is fixed for $\lambda \leq 50\%$ at $f_{cpu} = 4\text{MHz}$ and sleep states are used for idle times. The current consumption can be calculated as follows:

$$
I_{modDVS}(\lambda) = \begin{cases} I_{DVS}(\lambda) & \forall \lambda > 50\% \\ 2\lambda \cdot I_{cc}(\frac{f_{cpu_{max}}}{2}, V(\frac{f_{cpu_{max}}}{2})) & \\ +(1 - 2\lambda) \cdot I_{sleep} & \text{else} \end{cases} \quad (7)
$$

The estimated yield $Y(\lambda)modDVS = \frac{I_{DPM}}{I_{modDVS}}$ is also shown in figure 1. As we can see in figure 1, already DVS has nearly always a positive effect. Only for very low workloads lambda below approx. 5%, the yield is < 1. Using modDVS we can improve the yield for workloads below 50% and ensure positive gains.

It should be mentioned that modDVS is not limited to the Atmega1284p. The yield's peak is a general effect which results from the minimum supply voltage V_{min}. With equation 3 this peak is reached whenever $V(f_{cpu}) \leq V_{min}$. For instance, [7] describes this point for the msp430 micro controller.

IV. Implementation

The yield estimation has shown, that a dynamic voltage supply will improve the energy efficiency of the processing unit. To implement 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. For our prototype implementation we identified the following requirements:

- *Minimal overhead*
  The current consumption of the voltage scaling module should be as low as possible. Power management is one

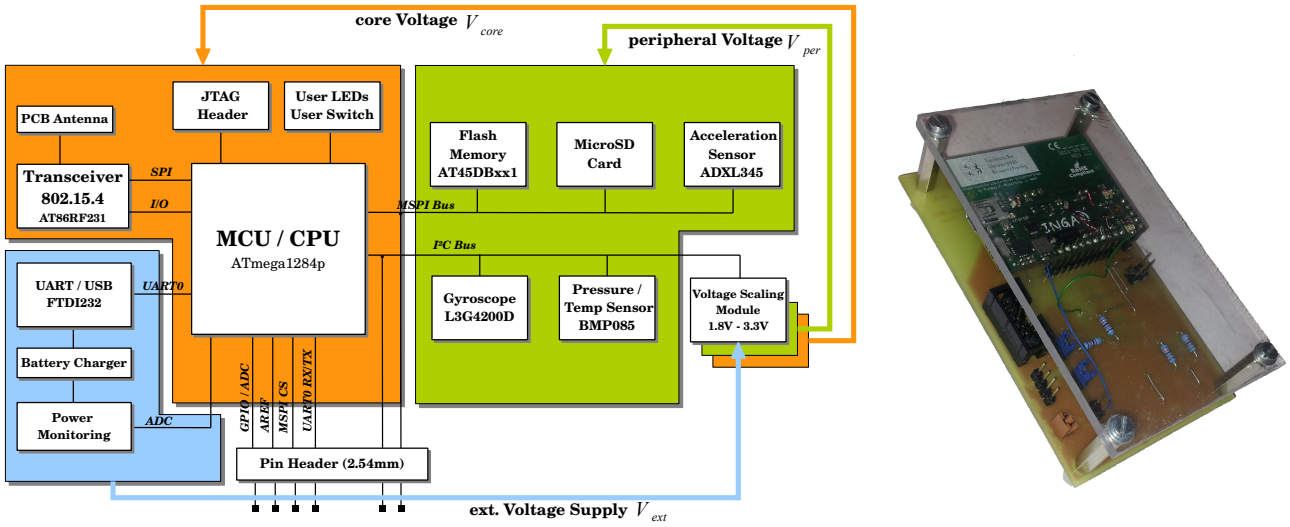


Fig. 2. Block diagram of the system integration and picture of the prototype implementation

part of a sensor node where sleep states are not recommended, so that every additional current consumption has to be compensated by the expected yield.

- *Two independent voltage paths*
  The core functionality, which includes the micro controller plus the transceiver unit, should have a separate voltage path. The average voltage level of these components is much lower than the required voltages of most of the peripherals. Besides a lower voltage level, the usage duration of peripherals compared to the micro controller is much shorter, so that on the one hand the micro controller and transceiver profit by a lower voltage level and on the other hand the peripherals can be switched off during idle times.
- *Scalable voltage levels*
  Both voltage paths, the core voltage V_{core} and the peripheral voltage V_{per}, should be scalable between $1.8V$ to $3.3V$.
- *No calibration*
  To allow for a larger number of nodes, calibration of each single node should be needless. The interaction between hard- and software should work from the scratch without any external adjustment.

LDO vs. step down converter: With the model function (cf. section III) we compared an LDO [17] with a suitable step down converter [9]. The power consumption as a function of the duty cycle is given in figure 3. The result is that the average power dissipation of the step down converter is $\approx 7.8\%$ lower. On the other hand the break-even for the LDO is reached at a duty cycle of $\approx 25\%$, which results from the significantly higher quiescent current of the step down converter [9] (≈ 200 times higher). Unfortunately, it is not possible to define a typical duty-cycle for WSN applications. [18] describes two different categories. For example, environmental sensing applications tend to result in low duty cycles, whereas structural health monitoring requires higher sampling rates with higher duty cycles. The concept of a voltage scaling module would work with both types of converters. We decided to take an LDO for the prototype implementation.

Software defined voltage level: The digital interface is implemented through a I^2C 8-Bit digital potentiometer [19] with two variable ohmic resistors. Every resistor controls an LDO [17] 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

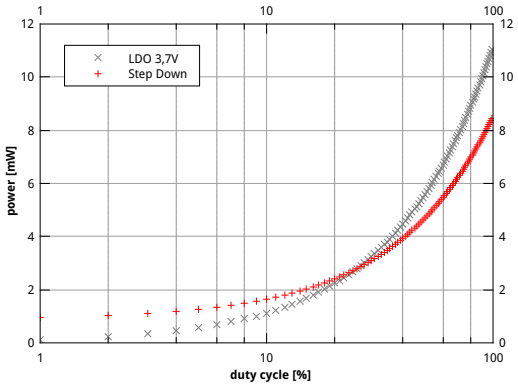


Fig. 3. LDO [17] vs. step down converter [9], power dissipation as a function of the duty cycle

expected and measured voltage levels is ≥ 0.999 for both voltage paths V_{core} and V_{per}.

Figure 4 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.

Overhead: The current consumption of the voltage scaling module depends on its usage. The lowest measured consumption $I_{cc}vs = 14.3\mu A$ is reached with disabled peripheral voltage $V_{per} = 0V$ and $V_{core} = 1800mV$, the highest one with $I_{cc}vs = 19.3\mu A$ at $V_{per} = V_{core} = 3300mV$.

Provided that a sensor node is already equipped with a voltage converter, the only extra costs for an active voltage scaling module would be an additional digital potentiometer.

B. Clock rate adjustment via software

The concrete method to adapt the clock rate depends on the micro controller and clock source. For example the 16-Bit MCU of the Tmote Sky (msp430) [20] has special registers to set the frequency. Other platforms may use external programmable clock sources. The ATmega1284p micro controller in our prototype generates the main clock with an internal RC oscillator [16]. 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 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.

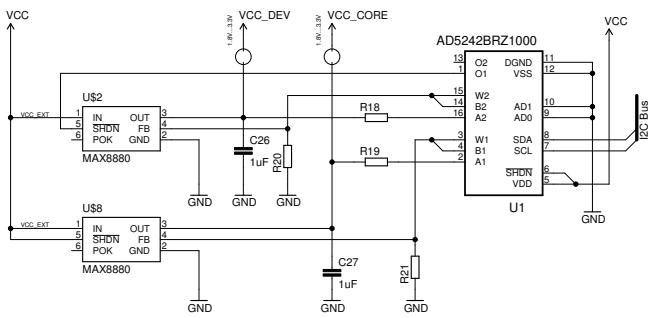


Fig. 4. Schematic of the I^2C voltage scaling module

C. System Integration

The prototype implementation is based on the INGA sensor node [21], which is an open hardware project[1]. The modified architecture is shown in figure 2. 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 [22], 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 $2\mu A$ [22].

V. Evaluation

We evaluated the effect of DVS and modDVS for a wireless sensor node by using the shown prototype. As the processing unit is only one part of a sensor node, we analysed the impact of a variable voltage levels for the transceiver unit and the peripherals as well. For comparison purposes, we also include according measurement results of an original INGA node.

A. Processing unit

The goal of this evaluation is to validate the theoretical approaches of section III. The model function $I_{cc}(f_{cpu}, V)$ respectively the yield estimation can be proved with measured data. Moreover the evaluation shows the benefit of DVS and modDVS compared to a fixed voltage source and clock rates (original INGA $8\text{MHz}@3.3V$) in practice.

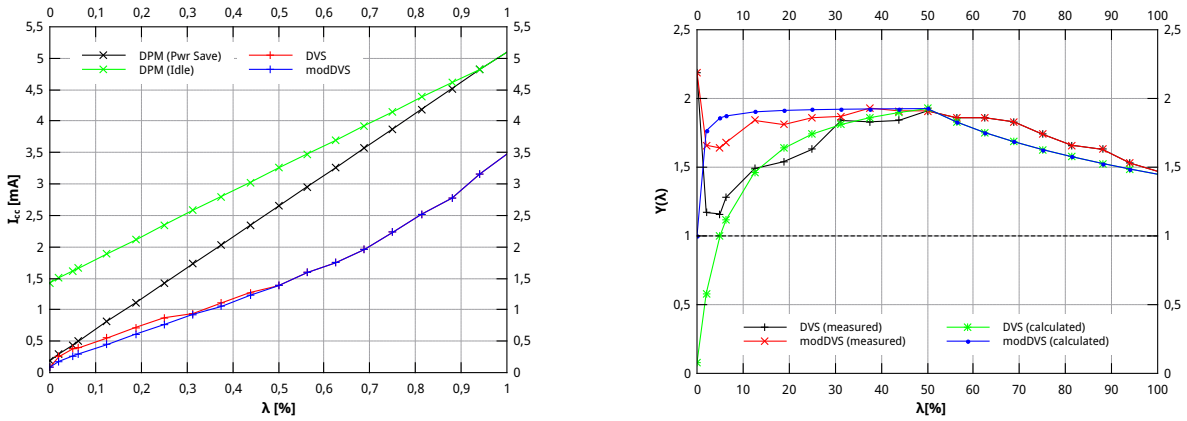
Figure 5(a) and 5(b) show the numerical results and the yield as a function of the system load λ. In figure 5(a) the DPM with two different sleep modes (Power Save, Idle) of an original INGA (fixed voltage, fixed frequency) is compared to DVS and modDVS of the prototype.

The yield $Y(\lambda)$ in figure 5(b), is given as the ratio between DPM(Power Save) and DVS respectively modDVS. It can be seen that the calculations correlate with the measured values for system loads $\lambda > 6.3\%$. The deviation in the area of $0\% \leq \lambda \leq 6.3\%$ is caused by a simplification within the model function, as well as a wrong estimated sleep current. Nevertheless the practical result shows, that it is possible to reduce the current consumption by an average factor of $\overline{Y} = 1.6551$ (DVS) respectively $\overline{Y} = 1.7829$ (modDVS).

B. Peripherals

The original INGA works with a clock rate of $f_{cpu} = 8\text{MHz}$. We have to consider here what happens with respect

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



(a) Supply current I_{cc} as a function of system load λ

(b) Yield through DVS and modDVS compared to DPM

Fig. 5. Impact on the current consumption of the processing unit ATmega1284p

to the peripheral components when the clock rate is reduced ($f_{CPU} = 4\text{MHz}$ as default clock rate for modDVS). In the worst case the SPI communication between micro controller and SPI peripherals when using modDVS is two times slower, because the SPI clock is bounded by the main clock f_{cpu}. Compared to the SPI clock, the I^2C clock is much lower ($f_{i2c} \leq 400\text{kHz}$), so that it makes no difference whether the micro controller runs either with 8MHz or 4MHz. This information is important, because we used the energy for our comparison and not only the pure supply current.

The measurement results, which are given in figure 6, include the energy of the micro controller during a peripheral read, so that on the one hand the yield is pushed by a more energy efficient processing unit, on the other hand the relation between peripherals supply voltage V_{per} and the yield compared to a fixed voltage (original INGA $3.3V$) is noticeable.

The AT86DB161 memory device of INGA [21] shows a stronger dependency on the SPI clock, so that it is less efficient to run the micro controller with $f_{cpu} = 4\text{MHz}$. For this reason we added a boost method to the software controller of the prototype which uses a prescaler to double f_{cpu} in a short time ($\approx 750\mu s$) so that the yield of the memory is based on a lower core plus peripheral voltage.

Beside the benefit of a scalable peripheral voltage, the design (cf. section IV) offers the possibility to switch off the whole peripheral part of the sensor node.

C. RF Communication

The transceiver chip of INGA is connected to the core voltage V_{core} (cf. figure 2), but according to [23], this component does not profit so much from a lower voltage level. For this reason it is more important to analyse how a lower clock rate of modDVS will effect the cost per bit for RF communications compared to original INGA. We used Contiki OS for this evaluation. Considering, in this realm, the RF communication as a platform independent process we comprised the Tmote Sky [20] into the evaluation. This sensor node is based on a msp430 micro controller, for which [8] theoretically analysed the impact of a dynamic voltage scaling (cf. section II).

Furthermore, it is possible to use active voltage scaling (respectively modDVS) on a this node in practice. The voltage scaling module of section IV-A could be easily connected, as the Tmote Sky is powered through a battery directly (without any voltage conversion).

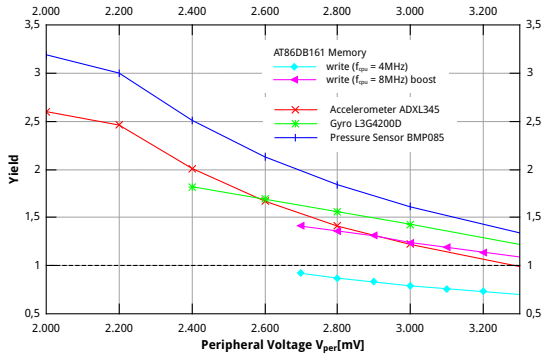


Fig. 6. Yield compared to peripherals with fixed supply voltage

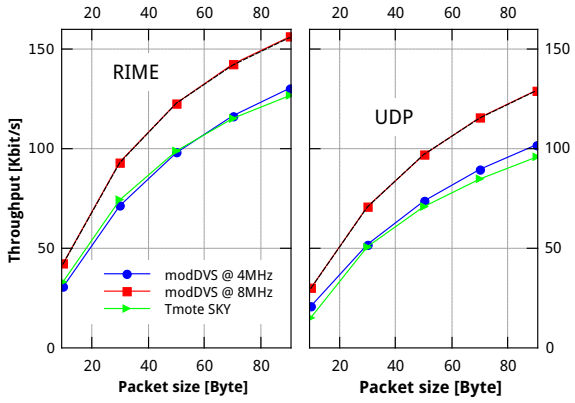


Fig. 7. Throughput of UDP and RIME under Contiki OS with TmoteSky and modDVS INGA at different clock rates

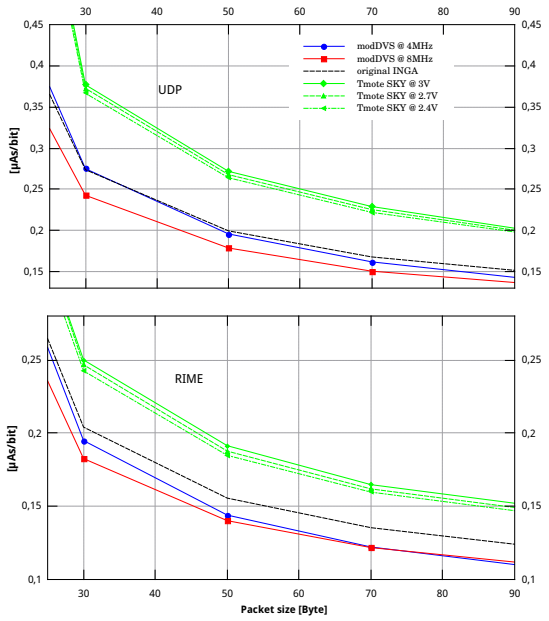


Fig. 8. Comparison of the cost per bit of UDP and RIME under Contiki OS with TmoteSky and modDVS INGA at different clock rates

Figure 7 shows the throughput of UDP and RIME under Contiki OS. Although the TmoteSky and the modDVS INGA have a clock rate of $f_{cpu} = 4\text{MHz}$, whereas the original INGA works with $f_{cpu} = 8\text{MHz}$, the throughput is not halved. This means, that the minimal clock rate for modDVS ($f_{cpu} = 4\text{MHz}$) does not influence the cost per bit efficiency as severe as expected.

The electrical charge per bit, as a function of the packet size for UDP and RIME, is given in figure 8. The results show, that the efficiency of the modDVS INGA $f_{cpu} = 4\text{MHz}$ converges with a growing packet size to the modDVS INGA with $f_{cpu} = 8\text{MHz}$. However, compared to the original INGA (dashed line), both DVS variants benefit from an active voltage scaling during data transmission. The reason why higher clock rates are more efficient is because of the higher throughput. The time to transmit data is reduced which results in a lower charge per bit $[\mu As/Bit]$. The TmoteSky is less efficient, because of older and therefore less efficient components [24], [25]. Nevertheless the effect of a downscaled voltage supply on this sensor node is noticeable too. For both UDP and RIME, the electrical charge per bit $[\mu As/Bit]$ is $\approx 3\%$ lower when the supply voltage of the tmote Sky is decreased from $3.0V$ to $2.4V$.

Usually, a transceiver provides transmission power control to save energy, but during the idle listening it is not possible to reduce the power of the receiver. So another question is, if the current consumption during idle listening can be decreased through lower voltage levels. The results are listed in table I. DVS can be used to reduce the current consumption of the sensor node during idle listening. As mentioned before, the biggest part of this profit depends on the more energy efficient micro controller (cf. [23]). Nevertheless, the current consumption of a transceiver can be affected by the external voltage supply.

TABLE I
CURRENT CONSUMPTION DURING IDLE LISTENING

Platform	modDVS INGA		original INGA	TmoteSky
	4MHz	8MHz	8MHz@3.3V	4MHz@2.4V
$I_{cc}[mA]$	14.77	17.67	20.41	19.06

Overall we can conclude that the proper adjustment of the voltage supply can decrease the total current consumption significantly.

VI. BATTERY LIFETIME IN PRACTICE

The results shown in the previous section V demonstrated the effect of DVS by considering the components and analysing the processing unit, the transceiver and the peripherals. For the overall effect, a kind of an integrated system test has been done to investigate whether and how much the battery lifetime really increases when using modDVS. For this purpose we took the following two example applications, which were implemented as a Contiki OS protothread.

1) *Sample and send*
   This application measures the temperature with a period of $T = 2s$ and sends the data via RIME to a central node. The central node sorts the values by node and package id. The transceiver is only activated when data is transmitted, whereas the micro controller stays active all the time.
2) *Sample and send on request*
   This application measures the temperature too, but it samples and sends the results only after a request of a central node. The request has a period of $T = 1s$ and the idea behind this test is, that the sensor node should spent most of the time with idle listening.

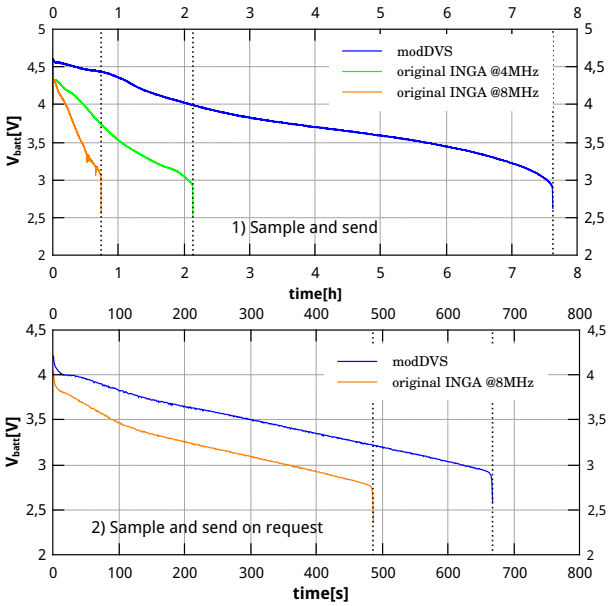


Fig. 9. Lifetime extension of two example applications when using modDVS

Every node was equipped with brand-new batteries. To evaluate the lifetime in an adequate short time, we selected batteries with a very low capacity [26]. We declared the lifespan of a node from the first to the last received package. The observation of the discharge curves, which are plotted in figure 9, was done by the nodes themselves. An ADC of INGA was used to sample and then send the battery voltage levels together with the temperature values.

Both applications *1)* and *2)* profit from the active voltage scaling and it is clear that the battery lifetime enlarges significantly. In application *1)*, we have a factor of 10.348 between the original INGA and modDVS on the prototype implementation. The permanent usage of the transceiver limits the extension of the lifetime of application *2)*. As explained in section V-C, the transceiver does not profit from a lower voltage level. Nevertheless, in case of application *2)* the lifetime of INGA with modDVS is still $\approx 37\%$ higher compared to an original INGA.

VII. Conclusions

In this paper we presented a theoretical approach to estimate the yield and the ideal usage of Dynamic Voltage Scaling on a wireless sensor node. Based on this we derived modDVS, which is a hybrid of DVS and DPM. With modDVS it is possible to improve the classical DVS method by $\approx 7.7\%$ for a ULP micro controller. Although the presented methods are based on a specific micro controller [16], other processing units can be analysed, respectively optimized in the same way.

In addition to the theoretical considerations, we developed a sensor node architecture with DVS functionalities. The prototype is a modification of a conventional sensor node [21] where only COTS parts were used.
With an evaluation we proved the preliminary considerations and showed the advantage of modDVS compared to a static supply voltage. Furthermore, if not only the processing unit is considered but a separate scalable voltage path is provided for the peripherals, we can improve the energy efficiency of the whole node further. The corresponding evaluation showed, that other components like transceivers, sensors or memories benefit from a dynamic power supply too.

All in all active voltage scaling on small wireless sensor nodes can help significantly to increase the WSN lifetime. We showed that a temperature sensing node lives ≈ 2.7 times longer when using modDVS instead of a fixed voltage supply.

References

[1] R. Jurdak, P. Baldi, and C. Lopes, "Adaptive low power listening for wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 8, pp. 988 –1004, aug. 2007.

[2] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys '06. New York, NY, USA: ACM, 2006, pp. 307–320. [Online]. Available: http://doi.acm.org/10.1145/1182807.1182838

[3] G. Shafiullah, A. Thompson, P. Wolfs, and S. Ali, "Energy-efficient tdma mac protocol for wireless sensor networks applications," in *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on*, dec. 2008, pp. 85 –90.

[4] F. Pianegiani, "minitp: A protocol for the minimization of the transmit power in wireless networks," in *Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE*, may 2011, pp. 1 –6.

[5] U. Tietze and C. Schenk, *Electronic Circuits: Handbook for Design and Application*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[6] STMicroelectronics, "STM32 L1 Ultra Low Power," 2012. [Online]. Available: www.st.com/stm32l

[7] Y. Cho, Y. Kim, and N. Chang, "Pvs: passive voltage scaling for wireless sensor networks," in *Low Power Electronics and Design (ISLPED), 2007 ACM/IEEE International Symposium on*, aug. 2007, pp. 135 –140.

[8] L. B. Hoermann, P. M. Glatz, C. Steger, and R. Weiss, "Energy efficient supply of wsn nodes using component-aware dynamic voltage scaling," *Wireless Conference 2011 - Sustainable Wireless Technologies (European Wireless), 11th European*, pp. 1 –8, april 2011.

[9] Texas Instruments, "15v, 75ma high efficient buck converter," 2010. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps62120.pdf

[10] J. Shen, X. Yuan, L. Shen, D. Yao, and H. Liu, "An energy-efficient node for wireless sensor networks using software radio," in *Electro/information Technology, 2006 IEEE International Conference on*, may 2006, pp. 245 –249.

[11] Texas Instruments, "Power and battery management ic for li-ion powered systems," 2005. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps65010.pdf

[12] ——, "Wireless handset solutions : Omap mobile processors." [Online]. Available: http://www.ti.com/lsds/ti/omap-applications-processors/technologies.page

[13] W. Tuming, Y. Sijia, and W. Hailong, "A dynamic voltage scaling algorithm for wireless sensor networks," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 1, aug. 2010, pp. V1–554 –V1–557.

[14] T. Hamachiyo, Y. Yokota, and E. Okubo, "A cooperative power-saving technique using dvs and dms based on load prediction in sensor networks," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, july 2010, pp. 7 –12.

[15] W. Dargie, "Dynamic power management in wireless sensor networks: State-of-the-art," *Sensors Journal, IEEE*, vol. 12, no. 5, pp. 1518 –1528, may 2012.

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

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

[18] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*, ser. IPSN '07. New York, NY, USA: ACM, 2007, pp. 254–263. [Online]. Available: http://doi.acm.org/10.1145/1236360.1236395

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

[20] Moteiv Corporation, "Tmote Sky : Datasheet," 2006. [Online]. Available: http://www.snm.ethz.ch/pub/uploads/Projects/tmote_sky_datasheet.pdf

[21] 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.

[22] 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

[23] Atmel, "At86rf231, low power 2.4 ghz transceiver for zigbee, ieee 802.15.4, 6lowpan, rf4ce, sp100, wirelesshart and ism applications," 2009. [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/doc8111.pdf

[24] Chipcon, "Cc2420 2.4 ghz ieee 802.15.4 / zigbee-ready rf transceiver," 2012. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2420.pdf

[25] Texas Instruments, "Msp430f15x, msp430f16x, msp430f161x mixed signal microcontroller," 2002. [Online]. Available: http://www.ti.com/lit/ds/symlink/msp430f1611.pdf

[26] renata batteries, "364, sr621sw (sr60), watch battery." [Online]. Available: http://watchbatterycrossreference.com/pdfs/R364.pdf