Distributed Energy Measurement in WSNs for Outdoor Applications

Robert Hartung, Ulf Kulau and Lars Wolf
Institute of Operating Systems and Computer Networks
Technische Universität Braunschweig
Braunschweig, Germany
Email: [hartung | kulau | wolf]@ibr.cs.tu-bs.de

***Abstract*—Energy is one of the most critical resources in Wireless Sensor Networks (WSN). For any WSN deployment, being it for a productive use case or for a testbed, the ability to perform energy measurements is of major importance, e.g., to estimate the status of the WSN, the need for maintenance, or for remaining lifetime. This paper presents a distributed energy measurement system for outdoor applications. An existing testbed has been extended by the micro controller unit (MCU)-based oscilloscope PotatoScope. Measuring voltage and current of the sensor node, undervolting effects and energy consumption of realtime applications can be observed. The PotatoScope is capable of capturing the power dissipation precisely at a high temperature range of 100 K. Using high sample rates with up to 500 kHz, the system needs to be capable of storing and archiving the measured data. A final evaluation shows a 0.9% drift for voltage measurements and a 1.6% drift for current measurements over the range of 100K. Using an SD card, long-term measurements can be made with up to 500 kHz. Nevertheless, a live mode allows sample rates up to 2 MHz when using the PotatoScope with a computer.**

I. Introduction

Although the academic research on WSNs has been going on for more than 15 years, real-world experiences show that it is still challenging to operate WSNs reliably. This is why testbeds have been established enabling users convenient examinations of their applications for such networks. Many testbeds and management solutions for WSNs exist like [1], [2], [3], [4], [5], [6], [7]. However, all of these testbeds have in common that they are only located indoors. However, especially when WSNs are deployed outdoors, the hardware of the sensor nodes suffers from the environmental conditions. Recent publications have shown that temperature is still an underestimated factor. The efficiency of transceivers, the energy efficiency of the components and batteries as well as the aging of electronic parts is affected by the temperature.

In Bannister et al. [8] the impact of temperatures on the reliability of radio transceivers has been evaluated. They measured the Receive Signal Strength Indicators (RSSIs) of a real world WSN deployment and observed a drop of the RSSI and the packet reception rate (PRR) at higher temperatures. Boano et al. [9] describe variations of up to 56°C within 24h and propose to raise the transmission power to mitigate the adverse effect on the transmission unit [10].

Beside this, Kulau et al. [11] take advantage of the temperature dependency of Complementary Metal-Oxide Semiconductor (CMOS) gates and implement an active undervolting to extend the lifetime of wireless sensor nodes. The absolute minimum voltage level for MCUs gets therefore temperature dependent which allows lower voltage levels at higher temperatures. A basic study how this approach influences the energy efficiency of nodes deployed outdoors has been performed in [12].

In further consequence the energy efficiency and energy budget of a WSN is highly affected by its environmental condition.

To facilitate investigations on these effects and to evaluate power consumption of applications in realistic outdoor environments, we have build the PotatoNet testbed, consisting of 18 nodes, distributed over an area of almost 10000 square meters [13]. Measuring the energy profile of applications – including MAC and routing protocols – can help to increase the lifetime of WSNs significantly. Furthermore, knowing the current drawn from a node can help to optimize the discharge characteristics of batteries [14].

In this paper we propose a reliable, scalable and convenient distributed energy measurement of nodes by using a MCU-based oscilloscope – the so called PotatoScope. An essential requirement for the measurement is correctness within a widespread temperature range of 100 Kelvin. It has to be ruled out that changes in power dissipation due to environmental variances are not traced back on temperature dependencies of the PotatoScope itself.

The distribution of our nodes in our existing testbed requires an on-line measurement, this implies that we measure the energy at each node individually. A current mirror will not be used, as the current would be doubled and this would be a conflict to the goal to save energy in WSNs. Due to the low current consumption of nodes, we cannot use a inductive-based solution, as the magnetic field cannot be measured in all cases. Finally, our solution is using a shunt resistor as this is a common and widely used method.

Moreover, we have to ensure that the proposed approach is scalable. Hence, the PotatoScope should a) be inexpensive and b) implement a distributed storage of energy measurement results. The entire system has to be easy-to-use by the user and should provide transparent interfaces for further customizations.

The rest of this paper is structured as follows: The next

section will discuss related work and outlines differences to our solution. The existing testbed is presented in Section III. Then, Section IV will introduce the detailed architecture of the PotatoScope. A theoretical calculation of the measurement error will be performed in Section V. The evaluation of the PotatoScope is discussed in Section VI. We present our system for distributed energy measurements in Section VII before giving a final conclusion in Section VIII.

II. Related Work

In most setups it is hard to get a realistic picture of energy usage. You can hook up a single node to expensive lab equipment such as an oscilloscope to study it in detail. This, however, does not give an overview how a whole network is performing. Furthermore, it is impractical to apply in outdoor testbeds (cf. previous Section I). Often researcher rely on energy accounting mechanisms built into the operating system. For every transmission or other action they assume a certain fixed amount of energy to be used [15], [16]. The problem is that these mechanisms are not very accurate as the current consumption between different nodes varies significantly even under constant environmental conditions [17]. When individual and precise measurements are required, currents from a few micro-amperes up to several hundred milli-amperes have to be covered. This can usually not be realized using a single shunt resistor, but a series of different resistors is needed. Thus, the design of such energy meters is more complex and adds additional costs to the system as they require multiple stages to measure these currents [18]. In fact, due to the costs and the complex switch-over such systems are neither scalable nor suitable for outdoor applications.

Li et al. [17] present an approach that uses different motes to derive energy models. Evaluations have been performed in a lab environment to control the temperature easily. However, the impact of different temperatures on the measurement arrangement was not considered, but current consumption at different temperatures where recorded without any counter-check. This means that temperature effects resulting from the measurement circuit itself are not taken into account. An important note to take from their paper is that using a simple modeling approach is not enough. As mentioned above, even across identically constructed nodes there is a difference in power dissipation of up to 15%. As a consequence the model has to be customized for each node and for temperature effects as well.

A further approach performs an on-line measurement on every single node, to directly measure voltage and current consumption. Some *energy meters* have been presented in [19], [20], [21], [22]. Hergenroeder et al. [19] built an USB-based solution – the Sensor Node Management Device (SNMD) – that measures energy of a connected sensor node. It can also be used for performing maintenance tasks like programming the node. Measurements can be taken at a sample rate of up to 500 kHz in buffered mode and is limited to 20 kHz without a buffer on a single channel. The sampling resolution is 16bit and they support a high range for the voltage from 0 V-10 V and for currents of up to 500 mA. Using a sample rate of 500 kHz the buffer – that can hold up to 448000 samples – is filled up after only 1.1 seconds. As a consequence, long term measurements of multiple seconds or minutes can only be performed in the slow, unbuffered mode at 20 kHz. With costs of about $300 [23], slow continuous sample rates of 20 kHz and only a single measurement channel, the SNMD would not be applicable for use in our testbed.

Several more energy meters have been presented, like Multichannel Energy Measurement Device (MEMD) by Zhu et al. [20]. Their idea is to use a multi layer architecture to provide power for MCU, transceiver and sensor differently to get better knowledge of these parts. While the general approach seems solid, they only use a buffer of 1900 samples per channel which is filled fairly fast at samples rates between 5.5 kHz (unbuffered) and 150 kHz (buffered). The MEMD uses a serial interface for communication which automatically leads to slow data rates when saving data or using the unbuffered mode.

Finally, Prayati et al. [24] present the *Power Consumption Measuring System* (PCMS). This is a current-mirror based solution for creating a power consumption model of the TelosB processor board. It has six independent A/D convertes, that can sample at a rate of 50kHz. For measuring currents it uses a pretty expensive device that is placed between the measurement platform and a PC on the other side. Unfortunately this solution is pretty expensive as the device itself costs about $460.

The following Table I summarizes the features of the presented energy meters. A comparison to our PotatoScope is also given within this table.

TABLE I: Comparison of energy meters

Name	Channels	f_s [kHz]*	Resolution	Range [mA]	Duration
SNMD	1	400 (20)	16bit	0-150/200/500	1s
MEMD	4	150 (5.5)	10bit	0-40	$<$ 1s
PCMS	6	40	16bit	0-50	-
PS**	2	500	12bit	0-26.6	2.5h

*Buffered (Unbuffered)
**PotatoScope

Firstly, it can be seen that sampling for more than a few seconds is either not possible or limits the the user to very unacceptably low sampling rates. In addition, none of the previously presented energy meters is able to provide a reliable energy measurement for challenging environmental conditions – e.g. outdoor deployments. Up to now temperature effects have not been taken into account and to the best of our knowledge we present the first examination of temperature effects on energy meters in Wireless Sensor Network (WSN) testbeds.

III. PotatoNet testbed

This section gives a brief overview about our existing outdoor testbed PotatoNet [13]. The testbed was built in the light of Smart Farming applications, so that it is located

in a rural area with poor network connectivity and limited maintainability. Thus, PotatoNet was designed to be robust and to ensure that the testbed can survive for extended periods of times without needing local maintenance from specialists.

A central management server (Central Box – embedded Linux PC) is used to manage and monitor all connected nodes. The nodes themselves are connected to the Central Box via a WRTnode [1] – a $25 MIPS-based embedded Linux board with 680 MHz – that runs the embedded Linux distribution OpenWRT [2]. The WRTnode supports both an Ethernet and a Wi-Fi interface, of which we use the first one in combination with power-over-ethernet (PoE) to provide permanent power supply. The actual sensor node [25] is connected to WRTnode GPIOs providing a full-fledged In System Programmer (ISP). All nodes are connected via Ethernet cables between 30 and 80 meters to the Central Box. If needed, nodes can be concatenated by using the internal Ethernet switch of the WRTnodes. Redundant uplinks at the Central Box allow remote access to the nodes for maintenance or reprogramming. A condensed block diagram of the testbed is shown in Figure 1.

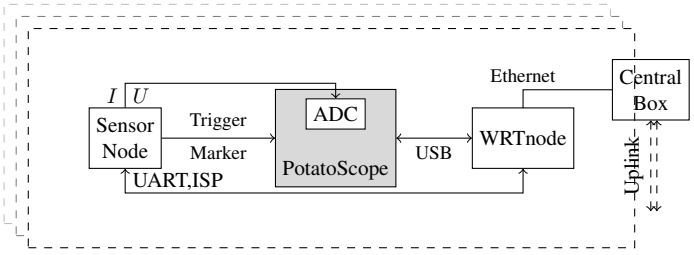


Fig. 1: PotatoNet architecture

In order to provide a distributed energy measurement the PotatoNet is extended by the PotatoScope – an MCU-based oscilloscope. After analog preprocessing the ADC of the PotatoScope's MCU is used to capture the node's voltage and current consumption. The sensor node itself can use dedicated GPIOs to start and stop (trigger) measurements. Moreover, GPIOs can be used as markers which allows tagging of the sensor node's source code. Finally, the USB connection between WRTnode and the PotatoScope is used for changing the configuration, starting measurements and data exchange. The potentially large bulk of data is stored on the PotatoScope itself and can be transfered via ethernet to the central box on request. A more detailed description of the PotatoScope is given in the following section.

IV. PotatoScope

On the one hand the PotatoScope should provide an accurate, fine-grained and temperature invariant sensing of the node's energy consumption. On the other hand the costs for one PotatoScope must be relatively inexpensive to equip as many nodes as possible. Hence, a well balanced selection of the involved components is needed to fulfil these opposing requirements.

[1]http://wrtnode.com
[2]http://openwrt.org

The used MCU is an ARM Cortex M3 implementation from ST's STM32F205 series [26]. It keeps the costs low while allowing the attachment of an external, temperature-stable reference voltage $V_{ref} = 2.5V$. For communication with the WRTnode, we use the external USB 2.0 Phy *Microchip USB3300*[3].Current and voltage measurements can be taken in parallel, using two different ADC channels. The detailed architecture of the PotatoScope is shown in Figure 2.

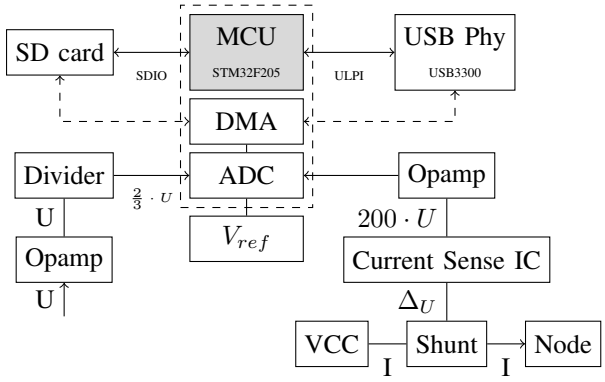


Fig. 2: Detailed architecture of the PotatoScope

The analog front-end of the PotatoScope is designed to exhibit a low temperature drift.

To prevent drawing current from the node, the voltage input channel is connected to an operational amplifier (impedance converter). A downstream voltage divider with a factor of $\frac{2}{3}$ is used to divide the input voltage of 3.3V to match the reference voltage.

For the current implementation we use a high precision shunt with $0.47\,\Omega$. Thus a current consumption from $0\,\text{mA}$ to $26.6\,\text{mA}$ can be measured. This range can be changed by replacing the shunt resistor. The VCC is passed through the scope and the shunt so we can measure the voltage drop Δ_U at the resistor. The potential difference is amplified by a current sense Integrated Circuit (IC) that outputs the voltage with a gain of 200. Due to the current drain of the ADC's sample-and-hold stage, a small current will be drawn for every single conversion from the input channels. Hence, another impedance converter is used to avoid small but measurable drifts at different sample rates.

A. DMA Capabilities

For data transfers involving the SD card and the USB interface, we use the integrated Direct Memory Access (DMA) Controller. Thus, a deterministic sampling is guaranteed and the workload of the MCU is reduced. In parallel to ongoing measurements the MCU is able to receive and handle control messages from the USB interface.

B. Trigger and Markers

As shown in Figure 1, the nodes themselves are able to trigger measurements by using GPIOs. However, these signals

[3]http://www.microchip.com/USB3300

cannot only be used to start or stop a measurement, but can also be used as *markers*. The node pulls these signals *high* or *low* to tag a position within a running measurement. This can be used to identify states in the node's software, e.g. radio on/off and allows a fine-grained analysis of a nodes power consumption down to the level of specific lines of code. Marker positions are highly precise as they are saved subsequently after conversion whenever one of the signal has been changed.

C. Analog-Digital Converter

The ADC operates in the *dual regular* mode. Both ADC channels that are used for measuring voltage and current of the node are measured in parallel and temporally stored in a 4 bytes wide special register. The upper half-word is used for ADC channel 2 and the lower half-word is used for holding the data of channel 1. After the end of conversion (EOC), the DMA controller transfers the register's data to a circular SRAM buffer. When either half of the buffer is full (or the measurement has been stopped), a DMA transfer will be started, to store the data on the SD card. At the moment we use a circular buffer with a total size of 30720 bytes or 7680 samples. The seamless forwarding of data from the buffer to the SD card allows continuous sampling over long periods of time.

A dedicated timer is used to sample at a fixed rate. Overall there are 24 predefined sample rates (1 Hz - 500 kHz) that are preconfigured within the PotatoScope. Selecting a sample rate will automatically configure the timer and set the optimum hold (cycle) time for the ADC. As a result the noise is reduced and the most accurate measurement at a given sample rate is achieved.

The internal ADC has a resolution of 12 bit. This means that our reference voltage of 2.5V will be divided into 4096 levels with steps of $\frac{2.5V}{2^{12-1}} = 0.611mV$ between them. The resolution for the voltage is hereby given by $0.611mV \cdot \frac{3}{2} = 0.917mV$. The resolution for the current measurement is equal to $\frac{0.611mV}{200 \cdot 0.47\Omega} \approx 6.5\mu A$. Additionaly, the ability to measure at high sample rates allows oversampling to achieve even higher resolutions by software.

D. Data Storage

Measuring the energy consumption at relatively high sample rates leads to a large amount of data. As the system is distributed and should scale with an increasing number of nodes, the data will be stored locally using an SD card. At the moment SD cards with a capacity of 8GB are used, which lasts for almost 2.5 hours of continuous measuring at a sample rate of 500 kHz. We do not use a file system like FATFS as it is not applicable for real-time applications [27].

In our implementation, we use one SD card block called *Data Description Block (DDB)* to describe subsequent measurement data. The data format is shown in Figure 3. This block is initially stored in block number 1 on the SD card and can describe 62 consecutive measurements. For each measurement, we store the number of samples and the sample rate. Finally, if 62 measurements have been taken, a new DDB will be stored after the data and the *Next DDB* field will be updated. Storing the absolute block number of the next DDB makes it easy to iterate over measurement information.

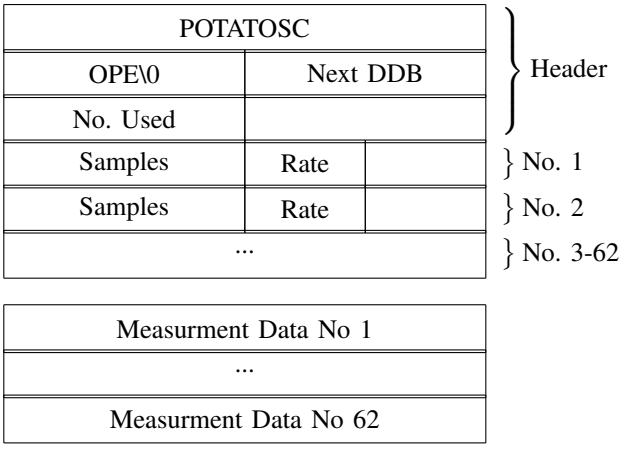


Fig. 3: Low Level Data Structure

Besides storing measurement data on the SD card, we use the first block to store calibration results for reducing offset errors, as the processor does not have an integrated EEPROM.

E. Measurement Modes

The PotatoScope can be used in different operating modes. As described above, the *triggered* mode allows nodes to control energy measurements on their own. Triggers can be used to start right in time at an important event, e.g. package reception. The *single shot* mode can be used to arm a trigger, when waiting for an event. Triggering can be done from the node itself or manually by the WRTnode. Last but not least, a *live* mode allows even higher sample rates of up to the maximum ADC frequency of 2 MHz and the use of the PotatoScope with a regular computer outside the testbed.

F. Command Line Interface

Basically a simple libusb-based *command line interface* (CLI) is used to control the PotatoScope or read status information. The CLI is cross-compiled for the WRTnode, while its main purpose is to hide the complexity of the underlying USB interface from the user and to make it easy-to-use. In total three USB interfaces are implemented: requesting and reading data, changing settings and a third one that is responsible to send notification to the USB host to indicate the end of a measurement run. These interfaces are separated so that they can be used simultaneously, because USB requires an interface to be claimed (locked) before being used.

V. Error estimation

Due to requirements of an inexpensive oscilloscope and a temperature stable sensing, components have to be chosen wisely. Maintaining good quality of results while using low-cost components is a balance between accuracy and cost. Therefore, this section gives a short overview of how to model

the theoretical error of the PotatoScope. Based on the chosen components we give an estimate of the PotatoScope's error.

As shown in Section IV the external voltage reference and the ADC is used in both measurements. Almost all errors can be expressed as a general offset or tolerance ϵ_{offset} and the temperature drift per 100 Kelvin ϵ_{drift}. The error of the voltage reference V_{ref} is given in the following Equation 1.

$$\begin{aligned} \epsilon_{vref} &= \epsilon_{offset} + \epsilon_{drift} \\ &= 0.04\% + 0.06\% \\ &= 0.1\% \end{aligned} \tag{1}$$

According to the datasheet of the MCU, the ADC has an accuracy of 5 LSB.

With an ADC resolution of 12 bit this is equal to $\epsilon_{ADC} = \frac{5}{4095} = 0.122\%$. Subsequently the error for the analog frotend can be calculated. The error of the operational amplifier is given below:

$$\begin{aligned} \epsilon_{opamp} &= \epsilon_{offset} + \epsilon_{drift} \\ &= 3.2mV + 0 \\ &\approx 0.13\% \end{aligned}$$

The estimation for the voltage measurement path is given in Equation 2:

$$\begin{aligned} \epsilon_{voltage} &= \epsilon_{vdiv} + (\epsilon_{opamp} + \epsilon_{ADC} + \epsilon_{vref}) \\ &=: \epsilon_{offset} + \epsilon_{common} \\ &\approx 2.36\% \end{aligned} \tag{2}$$

The voltage error $\epsilon_{voltage}$ consists of the common error of the operational amplifier ϵ_{opamp}, the ADC error ϵ_{ADC} and the voltage reference error ϵ_{vref}. Finally, the voltage divider error ϵ_{vdiv} is added to the common error – that is simply equal to 2% as we use two resistors with a precision of 1%.

$$\begin{aligned} \epsilon_{current} &= \epsilon_{shunt} + \epsilon_{ic} + (\epsilon_{opamp} + \epsilon_{ADC} + \epsilon_{vref}) \\ &=: \epsilon_{offset} + \epsilon_{common} \\ &\approx 2.21\% \end{aligned} \tag{3}$$

The error of the current input channel uses the same common error and adds the tolerance of the shunt ϵ_{shunt} and the gain error of the current sense IC ϵ_{ic} to it. In conclusion the final error is shown in Table II. In total it is 2.36% for the voltage measurement and about 2.21% for the current measurement. It should be mentioned that the temperature drift $\frac{ppm}{K}$ for 100 K is already considered for each estimation.

To reduce the errors ϵ_{offset} and ϵ_{opamp}, a one-time calibration is performed. To get the digital value from an analog voltage, a linear relation between input and output value can be formulated. The calibration uses this fact to calculate the two parameters of a line of the form $l(x) = m \cdot x + b$. As there might be noise on the measurement, three points are taken to determine the slope and intercept more precisely. Due to the calibration process, the total error could be reduced to $< 1\%$ for both channels.

TABLE II: Summary of the estimated errors (without calibration)

	ϵ_{vref}	ϵ_{ADC}	ϵ_{opamp}	ϵ_{offset}	**Error**
$\epsilon_{voltage}$	0.1%	0.12%	0.13%	2.00%	2.36%
$\epsilon_{current}$				1.84%	2.21%

VI. Evaluation of the PotatoScope

Before being deployed in PotatoNet, the PotatoScope has been evaluated in terms of accuracy and temperature invariance.

As introduced in Section V, the maximum theoretical error of about 2.5% can be reduced using a one-time calibration. Our calibration shows comparable results across the nodes and results in a measurement error $\epsilon_{voltage} \leq 0.87\%$ and $\epsilon_{current} \leq 1.56\%$.

The ADC accuracy is a lot worse than 5 LSB, as measured by ST Microelectronics (STM) itself [28]. Their measurements show dispersion of values of up to 21 LSB.

A. Noise

We applied a voltage to the circuit and observed a remarkable ADC dispersion. Results show that ADC values vary within a range of up to 60 LSB. This random noise is caused by several sources, e.g. lab conditions, power supply or other components of the PotatoScope itself. However, this means that values scatter ± 30 LSB around the expected value which corresponds to an error of 0.7% when considering a resolution of 12 bit. From this it follows that we cannot reduce errors below 0.7% due to noise in the signal chain.

B. USB 2.0 Interface

Evaluating the USB 2.0 interface is important for knowing limits of potential data transfer rates. Especially in the live mode, the data throughput might be a bottleneck. Figure 4 shows the results taken from several measurements.

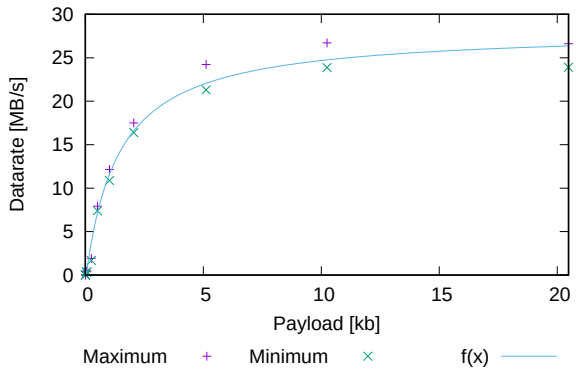


Fig. 4: Measurement of USB 2.0 throughput at different payload sizes

For this evaluation data of different payloads were transfered to the host. The total number of bytes received within a second were counted. Our evaluation shows a data rate of up to 26.5

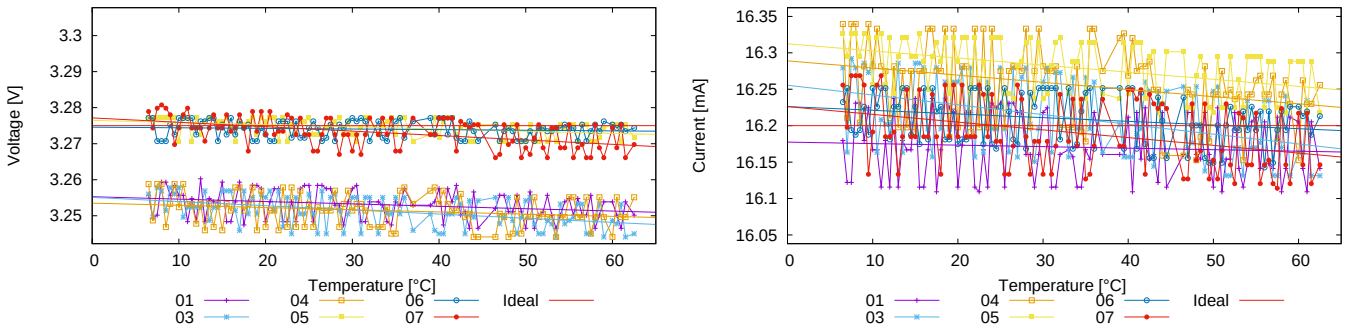


Fig. 5: Temperature drift of a set of nodes

MByte/s at a payload of 20kB. This is only about 45% of the theoretical maximum transfer rate of 480 MBit/s of USB 2.0, but still more than enough to transfer data at transfer rates of up to 2 MHz which requires a data rate of only 8MB/s.

C. Temperature Drift

The operation of WSNs in outdoor applications can lead to extreme temperature variations. Thus, a reliable measuring for a widespread temperature range has the highest priority. By using a climatic chamber, we also measured the impact of the temperature on the current and voltage measurement in practice. The results are depicted in Figure 5, where every node was calibrated beforehand. A small drift downwards within a range of 50 °C can be observed. The linear regression allows to approximate the total drift over a temperature range of 100K. The results for six nodes are shown in Table III.

TABLE III: Voltage and current drift over a range of 100K

Node		Δ_V		Δ_I
01	6.632mV	0.2%	0.020mA	0.13%
03	11.579mV	0.35%	0.134mA	0.83%
04	6.126mV	0.18%	0.098mA	0.61%
05	6.015mV	0.18%	0.097mA	0.60%
06	1.705mV	0.05%	0.050mA	0.31%
07	12.235mV	0.37%	0.106mA	0.66%

The maximum drift for voltage measurements is 0.37% or 12.235mV. For the current, the drift is bigger with 0.83% and 0.134mA.

VII. Distributed Energy Measurement

When measuring across a large number of nodes, data amount becomes larger with an increasing number of nodes. A theoretical consideration of the data rate R as the function of the sample rate f_s and the number of nodes n is given im Equation 4.

$$R(f_s, n) = f_s \cdot 4byte \cdot n \left[\frac{byte}{s}\right] \tag{4}$$

Considering the data rate of $R = 100Mbit/s$ and a sample rate of $f_s = 500kHz$. As a consequence, only $n = 6$ nodes could be used simultaneously before the entire network would be congested.

Thus transferring data directly to the Central Box is not an option. The solution is to leave data on the SD card of each PotatoScope and keep an index of all measurements across the nodes on a central server. This requires a database – like SQLite – to be used for indexing. Actual measurements can be downloaded from the nodes on demand by using the central database. The advantage is, that amount of storage available increases with every node added to the system.

A. General Architecture

Figure 6 shows the general architecture of the distributed energy measurement. All PotatoScopes are connected to the central server via the WRTnode. The question might come up why we do not use a SSH tunnel to interface the PotatoScope by using the CLI. Unfortunately the delay of a SSH connection is relatively high (few seconds), as the WRTnode is less performant. This would not allow a convenient handling of the energy measurement e.g. configuring or starting measurements immediately.

Another fundamental requirements are reliability and fail-over solutions as e.g. cables might suffer from external damages. When there is no active connection between the WRTnode and the central server, a fall-back solution is to keep a local log file of finished measurements. The list of these measurements will be transfered once the connection is (re-)established again. With this technique, pre-configured measurements can be performed, even if the central server is not reachable for some reason.

The core architecture consists of a simple client-server-architecture. Instead of the SSH tunnel, a more lightweight TCP connection is used to reduce the transmission delays between PotatoScope and central server. A ping message is

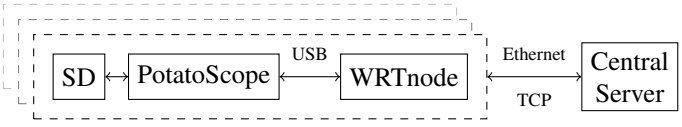


Fig. 6: Architecture of the distributed energy measurement system

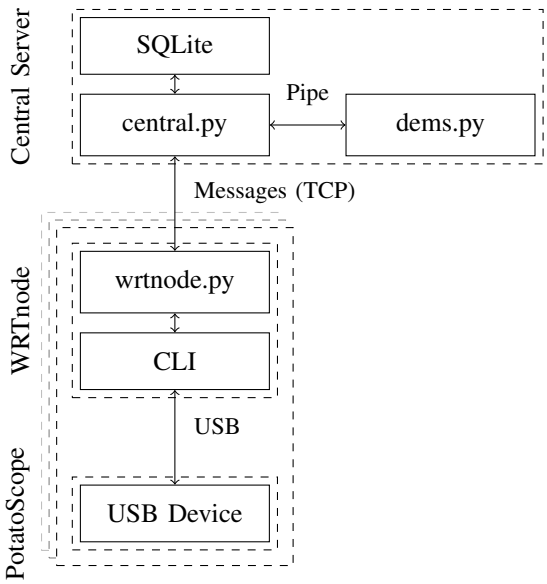


Fig. 7: Software architecture of our system

send to each node periodically to check whether a PotatoScope is still available.

B. Software implementation

The detailed software architecture of the entire system is shown in Figure 7.

On the bottom layer, the PotatoScope acts as a USB device to the WRTnode. The CLI, as presented in Section IV, is used to interface the PotatoScope by the WRTnode. On the one side a python script (wrtnode.py) is wrapped around the CLI to be responsible for calling the CLI on the WRTnode. On the other side the same script establishes, handles and maintains the TCP connection to the central server. As a measurement can end at every point in time, the notification interface is used, to wait for notifications permanently. Due to the fact that different USB interfaces are implemented (cf. Section IV), waiting for notifications and interaction with the PotatoScope e.g. reading data, can be performed concurrently.

The Central Box executes the central server where the nodes are connected to. This is where the user accesses the system. A further python script (central.py) is responsible for handling notifications, e.g. finished measurements. Moreover it provides a pipe for communication with the user interface. The communication uses serialized python classes, as the overhead is reasonable, but the gain of easy extendability is much higher. New types of messages can be implemented very easy, without knowledge of the underlying system.

The third script (dems.py) provides the distributed energy measurement system (DEMS) to the user. Using a pipe for communication, the dems can access the TCP connection indirectly and thus send data to the nodes, such as requesting the current configuration or starting measurements. Sampled data can be requested directly or a combination of command line options can be used for starting a measurement and reading the data afterwards. A simple call to the system allows to query used space of the SD cards, or a list of measurements across all nodes. For graphical representation of measurements *gnuplot*[4] into the system. This allows for plotting results directly on the central server. Downloading raw measurement data can take fairly long, as the capacity of the uplink might be limited.

As an extension the grouping of nodes is supported. This features makes it easy to start (different) measurements on a predefined subset of the nodes. Nodes are identified over the hostname. Using groups of nodes, different users can use the testbed at the same time, or nodes can be grouped by functionalities or applications running on them.

Two separate SQLite databases allow to parallel storage of measurement data and groups of nodes. This allows that groups can be exchanged, while keeping the list of data when different measurements are performend within the testbeds. A user can simply use its own group of nodes, upload applications and go ahead with measuring data.

The following listing demonstrates how a measurement can be taken on a group of the nodes. The first three parameters configure the scope to sample in single shot mode, at 10kHz sampling rate and to stop after 10000 samples (1 second). Finally, when the measurement has been completed, the data will be drawn into a figure.

```
dems -m S -r 10000 -l 10000 --plot -g group
```

Throughout the design of the system, we took special care to use clear interfaces. This enables us to exchange parts of our testbed. For example, management nodes or messages within the system can be easily extended or replaced.

C. Evaluation

To perform initial evaluations of our system, we emulate a smaller version of the testbed in our lab. A schematic view of the setup is depict in Figure 9, while a photo of this test arrangement is shown in Figure 8a.

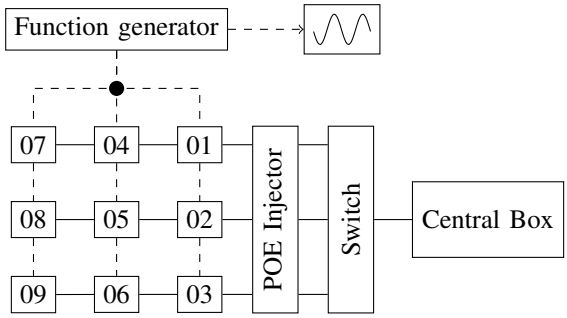
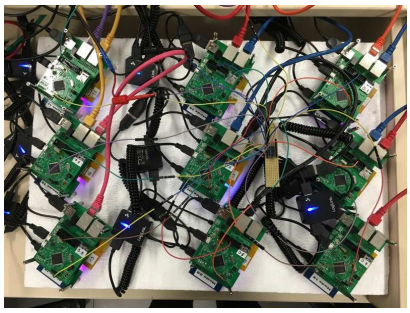


Fig. 9: Setup for the evaluation of our system

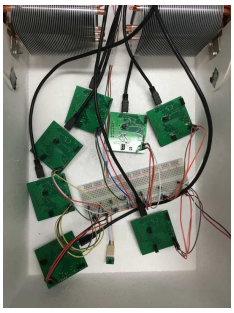
Instead measuring the energy characteristics of a sensor node, the PotatoScopes are connected to a signal generator. The advantage of an alternating test signal is, that we are

[4]http://www.gnuplot.info/



(a) Indoor test

(b) Central box with cables connecting the nodes



(c) Climatic chamber

Fig. 8: Pictures before and after installation

therefore able to measure differences in the amplitude and in the phase shift between nodes. A total of nine nodes are used for this evaluation. Using a star connection provides the same test signal to all nodes. For power supply, we also use a POE injector here that provides a constant power supply and the Ethernet connection to the nodes. Three nodes are concatenated as groups of three. Finally the Central Box is connected to the nodes by using an ethernet switch.

For the evaluation we use a low frequency sine wave. Our evaluation result is shown in Figure 10. It shows a small offset in amplitude, that has also been seen in the temperature drift measurements. The phase difference is surprisingly small with only 30ms, considering the fact that we do not use a common triggering mechanism. The triggering happens by sending a custom message to all nodes using the TCP connection. Nevertheless, further synchronization between measurements can be achieved by using the markers (cf. Section IV-B).

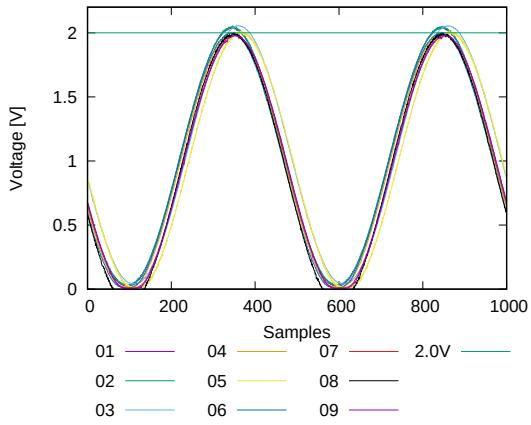


Fig. 10: Result of the evaluation of our distributed energy measurement system

VIII. Conclusion

In this paper, we presented a solution for distributed energy measurements in WSN testbeds. To the best of our knowledge, we are the first presenting an outdoor testbed, that supports testing applications of many kinds in actual outdoor environments while reliably measuring energy for research purposes. Most important was a dependable method for measuring energy on a high number of nodes. For this purpose, we designed the PotatoScope, a low-cost, high precision and temperature invariant MCU-based oscilloscope. PotatoScope can measure voltages from $0\,\mathrm{V}$ to $3.75\,\mathrm{V}$ and currents in the range of $0\,\mathrm{mA}$ to $26.6\,\mathrm{mA}$. Using sample rates of up to $500\,\mathrm{kHz}$ is absolute sufficient to characterize the energy consumption of WSN applications.

An theoretical estimation of measurement accuracy as well as practical evaluations within a climatic chamber showed that the PotatoScope is well suited to be used in outdoor applications. The total error plus the drift of energy measurements due to widespread temperatures within a range of $100\,\mathrm{K}$ is quite small ($< 1.56\%$).

A convenient distributed energy measurement system (dems) was implemented on top of the distributed PotatoScopes. The advantage, that the storage of measurements is also distributed among the nodes leads to a highly scalable system design.

Finally the entire system has been rated by measuring test signals across all nodes in our lab. Recently the distributed energy measurement system was applied to our testbed (cf. Figure 8b) to be used for ongoing research in WSN outdoor applications.

References

[1] K. Yi, R. Feng, N. Yu, and P. Chen, "Pared: A testbed with parallel reprogramming and multi-channel debugging for wsns," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, April 2013, pp. 4630–4635.

[2] P. Hurni, M. Anwander, G. Wagenknecht, T. Staub, and T. Braun, "Tarwis – a testbed management architecture for wireless sensor network testbeds," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, April 2012, pp. 611–614.

[3] G. Werner-Allen, P. Swieskowski, and M. Welsh, "Motelab: a wireless sensor network testbed," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, April 2005, pp. 483–488.

[4] A. Dludla, A. Abu-Mahfouz, C. Kruger, and J. Isaac, "Wireless sensor networks testbed: Asntbed," in *IST-Africa Conference and Exhibition (IST-Africa), 2013*, May 2013, pp. 1–10.

[5] J. Albesa, R. Casas, M. Penella, and M. Gasulla, "Realnet: An environmental wsn testbed," in *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on*, Oct 2007, pp. 502–507.

[6] C. Boano, M. Zuniga, J. Brown, U. Roedig, C. Keppitiyagama, and K. Romer, "Templab: A testbed infrastructure to study the impact of temperature on wireless sensor networks," in *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on*, April 2014, pp. 95–106.

[7] J. Sheu, C. Chang, and W. Yang, "A distributed wireless sensor network testbed with energy consumption estimation," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 6, no. 2, pp. 63–74, Jul. 2010. [Online]. Available: http://dx.doi.org/10.1504/IJAHUC.2010.034321

[8] K. Bannister, G. Giorgetti, and E. K. S. Gupta, "Wireless sensor networking for âĂIJhot âĂİ applications: Effects of temperature on signal strength, data collection and localization."

[9] C. A. Boano, H. Wennerstrom, M. A. Zúñiga, J. Brown, C. Keppitiyagama, F. Oppermann, U. Roedig, L.-A. Norden, T. Voigt, and K. RoĬLmer, "Hot packets: A systematic evaluation of the effect of temperature on low power wireless transceivers," 2013.

[10] C. A. Boano, N. Tsiftes, T. Voigt, J. Brown, and U. Roedig, "The impact of temperature on outdoor industrial sensornet applications," *Industrial Informatics, IEEE Transactions on*, vol. 6, no. 3, pp. 451–459, 2010.

[11] U. Kulau, F. Busching, and L. Wolf, "Undervolting in wsns: Theory and practice," *Internet of Things Journal, IEEE*, vol. 2, no. 3, pp. 190–198, June 2015.

[12] U. Kulau, S. Schildt, S. Rottmann, and L. Wolf, "Paint it black: Increase wsn energy efficiency with the right housing," in *Proceedings of the 6th ACM Workshop on Real World Wireless Sensor Networks*, ser. RealWSN '15. New York, NY, USA: ACM, 2015, pp. 3–6. [Online]. Available: http://doi.acm.org/10.1145/2820990.2820992

[13] U. Kulau, S. Schildt, S. Rottmann, B. Gernert, and L. Wolf, "Demo: PotatoNet – Robust Outdoor Testbed for WSNs: Experiment Like on Your Desk. utside." in *Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks*, ser. CHANTS '15. Paris, France: ACM, Sep. 2015, pp. 59 – 60. [Online]. Available: http://doi.acm.org/10.1145/2799371.2799374

[14] L. Feeney, C. Rohner, P. Gunningberg, A. Lindgren, and L. Andersson, "How do the dynamics of battery discharge affect sensor lifetime?" in *Wireless On-demand Network Systems and Services (WONS), 2014 11th Annual Conference on*, April 2014, pp. 49–56.

[15] J.-P. Sheu, C.-C. Chang, and W.-S. Yang, "A distributed wireless sensor network testbed with energy consumption estimation," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 6, no. 2, pp. 63–74, 2010.

[16] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, "Powertrace: Network-level power profiling for low-power wireless networks," 2011.

[17] Q. Li, M. Martins, O. Gnawali, and R. Fonseca, "On the effectiveness of energy metering on every node," in *Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems*, ser. DCOSS '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 231–240. [Online]. Available: http://dx.doi.org/10.1109/DCOSS.2013.67

[18] R. Zhou and G. Xing, "Nemo: A high-fidelity noninvasive power meter system for wireless sensor networks," in *Information Processing in Sensor Networks (IPSN), 2013 ACM/IEEE International Conference on*. IEEE, 2013, pp. 141–152.

[19] A. Hergenroeder, J. Wilke, and D. Meier, "Distributed energy measurements in wsn testbeds with a sensor node management device (snmd)," in *Architecture of Computing Systems (ARCS), 2010 23rd International Conference on*, Feb 2010, pp. 1–7.

[20] N. Zhu and I. O'Connor, "Energy measurements and evaluations on high data rate and ultra low power wsn node," in *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*, April 2013, pp. 232–236.

[21] A. Kopke and A. Wolisz, "Measuring the node energy consumption in usb based wsn testbeds," in *Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems Workshops*, ser. ICDCSW '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 333–338. [Online]. Available: http://dx.doi.org/10.1109/ICDCS.Workshops.2008.53

[22] X. Jiang, P. Dutta, D. Culler, and I. Stoica, "Micro power meter for energy monitoring of wireless sensor networks at scale," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, April 2007, pp. 186–195.

[23] P. Hurni, B. Nyffenegger, T. Braun, and A. Hergenroeder, "On the accuracy of software-based energy estimation techniques," in *Proceedings of the 8th European Conference on Wireless Sensor Networks*, ser. EWSN'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 49–64. [Online]. Available: http://dl.acm.org/citation.cfm?id=1966251.1966257

[24] A. Prayati, C. Antonopoulos, T. Stoyanova, C. Koulamas, and G. Papadopoulos, "A modeling approach on the telosb wsn platform power consumption," *J. Syst. Softw.*, vol. 83, no. 8, pp. 1355–1363, Aug. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2010.01.015

[25] 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. [Online]. Available: http://www.ibr.cs.tu-bs.de/papers/buesching-sensors2012.pdf

[26] STMicroelectronics. (2015-11-20) Stm32f205xx and stm32f207xx datasheet. [Online]. Available: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00237391.pdf

[27] S. Schildt, W.-B. Pottner, F. Busching, and L. Wolf, "Ratfat: Real-time fat for cooperative multitasking environments in wsns," in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, May 2013, pp. 388–393.

[28] STMicroelectronics. (2015-11-20) How to improve adc accuracy when using stm32f2xx and stm32f4xx microcontrollers. [Online]. Available: http://www.st.com/web/en/resource/technical/document/application_note/DM00050879.pdf