
Ismo Hakala, Ilkka Kivelä, Jukka Ihalainen, Jari Luomala, Chao Gao
University Of Jyväskylä
Kokkola University Consortium Chydenius
P.O.Box 567, FI-67701, Kokkola, Finland
{ismo.hakala, ilkka.kivela, jukka.ihalainen, jari.luomala, chao.gao}@chydenius.fi

Abstract—In this paper, we report the sensor function design and implementation of a wireless sensor network application for measuring environmental acoustic noise. The system is built on ATmega128 and CC2420 platform. The protocol stack is based on CiNet stack with a global synchronization scheme and supports multi-hop communications. Strict filtering function specified by ITU-R 468 (namely A-weighting) is followed. Both the indoor and outdoor test results were compared with standard sound level meters (CESV A SC-20c and Pulsar94) and showed a less than ±2dB error in both short-term and long-term measurement. Power consumption has been measured that a single AA-type battery can sustain the application. Comparing to the traditional noise measurement method, our wireless sensor network solution is much lower in cost, able to offer real-time data with sensed data timely coherent, and requests least attention after deployment.

Keywords-sensor node design; noise measurement; environment monitoring

I. INTRODUCTION AND MOTIVATION

In WSN (Wireless Sensor Network) development, sensor node design is never a trivial task due to the small size, low-cost, resource-limit nature of WSN applications. A requirement of measuring acoustic noise in both industrial and residential environments is present in Ostrobothnia area of Western Finland. In such a measurement system, a number of wireless sensor nodes must be scattered into a concerned area. Each node measures the noise level at its location and the data is collected by a sink node, which forwards the data to a web-based database. The network should be able to cover a large area such as a university campus, an industrial park, or a residential block.

In many countries the environmental noise has been regarded as a critical metric of working and living comfort. However, traditional way of conducting noise measurement exhibits great inconvenience: A test technician needs to carry a sound level meter to a measuring location and set the meter up for a specific measurement duration, which is usually several hours, and repeat this procedure for all the measuring points. The disadvantages of this method are obvious: 1) commercial sound meters are expensive, making large scale measurement very costly, 2) point-by-point measurement makes the result incoherent in time, 3) due to the lack of communication facility, the measured result is not available in real-time, and 4) the sound meter has to be fully attended, which increases the work load. These give the motivation of our WSN design. The goal of sensor node design is to minimize the necessity of peripheral components of a given microcontroller chip so that the cost of sensor is minimized.

The target WSN consists of a sink and a set of noise sensor nodes. The network topology is therefore a multi-hop tree as the sink node the root of the tree. This network can be deployed in both indoor and outdoor environments. The sensor nodes should keep measuring the noise level for e.g., a whole day [1]. Comparing to the traditional method, our system has the following significant advantages: 1) cost reduction in both sensing devices and workload; 2) real-time, multi-point, coherent measurement; and 3) least attention.

This WSN application has very special features because it demands excessive CPU power for data acquisition: the sensor nodes have to continuously sample the noise, thus leaving very short time for communication. The aim of this paper is to present the solution approaches to achieve accurate sensing and efficient communication for excessive data acquisition WSN applications.

In order to accomplish multi-hop communications, we built our application on CiNet platform [2], which is a IEEE802.15.4-based platform using ATmega128 microcontroller with a cross-layer protocol stack containing efficient routing, topology management, and synchronization functions [3]. The block diagram of sensor node architecture can be seen in Figure 1. This document mainly covers the sensor function design. The networking and data communication issues are briefly illustrated in Section VI. However, they are thoroughly presented in our technical report, which will be published later.

The rest of this document is organized as follows: A brief survey of related work is given in Section II; Background of sound level and its calculation is illustrated in Section III; Section IV covers the hardware design; Section V covers the software design and device calibration; Section VII shows the results of measurement and the comparison to a referencing device; Finally, Section VIII summarizes the design.

II. RELATED WORK

The work carried in [4] is probably the closest one comparing to our project. In their project, Tmote Sky
platform [5] were used, and the sensors were deployed to measure road traffic noise. Furthermore, from the measured data it is able to count the number and type of vehicles. The authors asserted that large-scale noise measurement using WSN solution is possible. However, the precision of measurement was not mentioned, the calibration of nodes was an open issue, and the sampling rate of sensor was set at 8kHz due to the CPU/ADC limit. In [6], the authors claimed that an environment monitoring system must have autonomy, reliability, robustness, and flexibility characteristics. They implemented a testbed network based on 8MHz TinyNode platform with 868MHz XE1025 transceiver to handle the acoustic data in a distributed way. However, detailed implementation was not included in the paper. In [7], a WSN was developed for coal mine monitoring to rapidly detect the collapse area and report to the sink node, (2) to maintain the system integrity when the sensor network structure is altered, and (3) provide a sound and robust mechanism for efficiently handling queries over the sensor network under unstable circumstances. The system was built on Mica2 motes platform and a prototype has been deployed to D.L. coal mine in China.

There is a bluetooth-based solution for noise measurement available in market [9]. In this solution, a bluetooth piconet, which supports maximum 5 noise nodes, can be deployed. It does not support multi-hop communication and therefore the application is limited within a small scale.

III. BACKGROUND: SOUND LEVEL

Sound is a traveling wave, which is an oscillation of pressure transmitted through a solid, liquid, or gas, composed of frequencies within the range of hearing and of a level sufficiently strong to be heard, or the sensation stimulated in organs of hearing by such vibrations [10]. This wave energy can be picked up (i.e., heard) by the human ears. The human ears are very sensitive devices, which can hear sound power in a wide range of 12 to 13 decimal magnitudes [11]. Sound power level is thus defined as logarithm scale, denoted as decibels, and can be calculated by

\[ L_p = 10 \log_{10} \left( \frac{p_{rms}^2}{p_{ref}^2} \right) = 20 \log_{10} \left( \frac{p_{rms}}{p_{ref}} \right) \text{dB,} \quad (1) \]

where \( p_{rms} \) and \( p_{ref} \) are RMS (Root Mean Square) sound pressure and reference sound pressure, respectively, and \( p_{ref} = 20 \mu Pa \) (RMS).

The human ear is able to hear frequencies from 20 Hz to 20 kHz. However, it does not have a flat response to all the frequencies in the range, therefore in sound level measurement the sound signal is often frequency-weighted so that the measured level will match the ear-perceived level. Such a frequency weighting devices is also called weighting filter due to its frequency response feature. The IEC (International Electrotechnical Commission) has defined several weighting schemes, namely A, B, C, and D weighting. A-weighting is valid for relatively quiet sounds, which fits the application of our project.

A-weighting filter is a frequency-select filter picking up the frequency range around 3-6 kHz, to which the human ear is most sensitive, while attenuating very high and very low frequencies to which the ear is insensitive. The corresponding transfer function of A-weighting filter is given according to [12] as

\[ H_A(s) = \frac{7.39705 \times 10^9 \cdot s^4}{(s + 129.4)^2(s + 676.7)(s + 4636)(s + 76655)^2}. \]

Figure 2 shows the frequency response of A-weighting filter, in comparison with B- and C-weighting filter curves. Sound level which is measured after A-weighting filter is also denoted as \( L_A \).

The effective sound pressure is the RMS value of the instantaneous sound pressure over a given interval of time (a.k.a., time-averaging sound level). In general, it is called equivalent sound level over time \( T \) and denoted as \( L_{eqT} \). Three of these time-weightings have been standardized in IEC-61672(2003): 'S' \( (T = 1 \text{ second}) \) originally called Slow, 'F' \( (T = 125 \text{ milliseconds}) \) originally called Fast and 'I' \( (T = 35 \text{ milliseconds}) \) originally called Impulse. A general equation of \( L_{eqT} \) for discrete system is denoted as

\[ L_{eqT} = 10 \log \left( \frac{1}{N} \sum_{i=0}^{N-1} \frac{p_i^2}{p_{ref}^2} \right). \quad (2) \]
where $N$ is the number of samples taken in time $T$.

In our design, a set of digital and analog amplifications are introduced for a suitable ADC (Analog-to-Digital Converter) range. All these amplifications, together with $p_{ref}$, can be treated as a single constant and able to be retrieved by comparing the ADC readings with a standard calibrator.

IV. DESIGN APPROACH: HARDWARE

Research has shown that a long-term exposure to sound level over 85dB will cause hearing damage [13]. The average sound level of motorway traffic is 70dB and definitely regarded as an uncomfortable noise. Typical environmental or background noise level in residential areas ranges from 30dB to 80dB. Considering some measurement margins, this requires a dynamic range of 60dB more of measurement range. The focus of this chapter is to describe our hardware design that covers a dynamic range of 60+dB.

The overall block diagram of sensor circuits is given in Figure 3. The sound pickup device is Monacor MCE-400 high-quality electret omnidirectional microphone cartridge [14]. The frequency range of MCE-400 fits well into A-weighting frequency curve.

A. Signal Amplification and Dual Channel for Dynamic Range

As mentioned previously, one of the design objectives is to measure the noise level in a range of [30,90+]+dB, which requires the data sampling and conversion system to cover a dynamic range of 60+dB. ATmega128 comes with a 10-bit build-in successive approximation ADC. 10-bit resolution can compensate a dynamic range of $20\log_{10}2^{10} \approx 60$dB only, which does not meet the requirement.

In order to solve the problem, we used two different pre-amplification gains to measure the sound levels in different ranges. As shown in Figure 3, two MAX4465 pre-amplifiers are in parallel arranged after the microphone. When the input level is low, the high-gain channel is selected; otherwise the low-gain channel is selected. A MAX4524 4-channel multiplexer is applied to select proper channel with the selection controlled by PD4 output pin of ATmega128. We here denote the two gains, $G_H$, $G_L$, respectively. In our design, $G_H = 390$, $G_L = 10$. Therefore, the dynamic range is increased by $10\log_{10} \frac{G_H}{G_L} = 31.82$dB correspondently.

The sampling program starts from measuring the high-gain channel input and keeps tracing the ADC input level. When a threshold is reached, PD4 is pulled low and the ADC input is switched to the low gain channel. Figure 4(a) is the oscilloscope snapshot of channel selection moment. It can be seen, that when the high-gain input gets saturated, PD4 is pulled down immediately and the low-gain channel is selected. When the low-gain channel is selected, and the sound level goes below a threshold, the system will switch back to the high-gain, as shown in Figure 4(b).

B. A-Weighting Filter

According to the transfer function $H_A(s)$, the A-weighting filter can be approximated by a cascading of 3 high-pass plus 2 low-pass filters. The frequency response of such a filter is also plotted in Figure 2. Need to mention that theoretically such a filter can be implemented by software as a digital filter. However, a digital filter involves excessive floating point calculation which definitely surpasses the limit of 8-bit ATmega128.

V. DESIGN APPROACH: SOFTWARE

The software is designed to be able to perform both slow and fast measurements: the RMS power calculation is done every 125ms and 1s corresponding to F/S time-weightings in IEC 61672, respectively.

A. 10-bit ADC setting

The 10-bit ADC is set in differential channel mode at 33kHz sampling rate. Such a sampling rate has frequency response from 0 to 16.5kHz, which covers the range of A-weighting filter well, therefore most of sound power is
picked up by ADC. The conversion formula is given as [15]:

\[ ADC = \frac{(V_p - V_n) \cdot GAIN \cdot 512}{V_r} \]  

(3)

where \( V_n = 1.25V \) is the negative input pin of ADC0, \( V_r = 2.56V \) is the selected voltage reference, and \( GAIN = 10 \) is the ADC’s internal gain. These three are constant values. \( V_p \) is the instantaneous voltage of positive input pin ADC1 fed by the A-weighting filter. According to (3), \( V_p \) must be in range of [1.0,1.5V] so that \( ADC \) output is in range of [-512, 511].

When \( V_p \) is at boundaries of range [1.0,1.5V], the strongest sound pressure level is seen at the microphone input, and in this case the low-gain channel \( G_L \) is selected. Thus we can accept an instantaneous peak voltage to the reference of \( V_{p,\text{peak}} = 1.5 - V_n = V_n = 1.0 = 0.25V \). If we assume that the input sound wave is a sinusoidal for the first order approximation, the RMS voltage of microphone input \( V_{rms,\text{mic}} = V_{p,\text{peak}} / G_L / \sqrt{2} = 17.7mV \). The MCE-400 has sensitivity of 7.9nV/Pa, so 17.7mV corresponds to \( p_{rms} = 2.24Pa \) sound pressure. Applying equation (1), it indicates that our system is able to measure sound level up to 100.98dB.

### B. Signal sampling and averaging

The basic algorithm is to take samples and square each of them. At the end of each 1ms, if \( G_H \) is being applied, in case the signal is too strong and clip has been observed, the MUX is turned into \( G_L \) channel; if \( G_L \) is being applied, in case the signal is weaker than a threshold \( \Delta \), the MUX is turned into \( G_H \). For every 125ms, an RMS average is calculated using equation (2). Algorithm 1 shows the algorithm in details, and Table I is the variable description for the algorithm.

### C. Calibration

Calibration was done with CESVA SC-20c, which is a high performance Type I integrating-averaging sound level meter. SC-20c is able to send measurement data to a host PC via RS232 interface and the data can be stored in file for further analysis.

The calibration acoustic source must be 1kHz sine wave. The \( ADC_{rms} \) values are compared with CESVA measurement in range of [30,90] decibels and an empirical curve fitting equation is drawn as

\[ L_{PA} = 8.1522 \times \ln(ADC_{rms}) + 32.7148(dB) \]  

(4)

The calibration curve can be seen in Figure 5.

### VI. NETWORKING & SYNCHRONIZATION

The application requires multi-hop communication to send sensed data back to a sink, and a global synchronization is necessary to offer time-coherent noise information. We proposed a SYNC2SINK application-oriented protocol to accomplish the task. In this section we briefly describe SYNC2SINK mechanism. The details of networking and
Algorithm 1 Sound Level Calculation Algorithm

\[
\text{for } k=0 \text{ to } N_1-1 \text{ do} \\
\quad S_1 = 0 \\
\text{for } i=0 \text{ to } N_m - 1 \text{ do} \\
\quad S_m = 0, c = 0 \\
\text{for } j=0 \text{ to } N_{1ms} - 1 \text{ do} \\
\quad \text{readADC to } x_i \\
\quad \text{if } x_i > H \text{ then} \\
\quad \quad c++ \\
\quad \text{end if} \\
\quad S_m = S_m + x_i^2 \\
\text{end for} \\
\text{if } \text{HIGHGAIN} \text{ then} \\
\quad \text{if } c \geq C \text{ then} \\
\quad \quad \text{set LOWGAIN} \\
\quad \text{end if} \\
\text{else} \\
\quad S_m = S_m \times (G_H/G_L)^2 \\
\text{if } S_m < \Delta \text{ then} \\
\quad \text{set HIGHGAIN} \\
\text{end if} \\
\text{end if} \\
\text{end for} \\
\text{RMS}(S_m) = \sqrt{\frac{1}{N_m N_{1ms}} S_m} \\
\text{record max(RMS(S_m)) and min(RMS(S_m))} \\
\text{S}_1 = S_1 + \text{RMS}(S_m) \\
\text{end for} \\
\text{mean}(S_1) = \frac{S_1}{N_1} \\
\text{record mean(S_1), max,min}
\]

Figure 5. Calibration and Curve Fitting

synchronization issues such as throughput control over multi-hop and clock skew control are presented in another document.

SYNC2SINK is built on CiNet protocol stack [3] (Fig. 1). SYNC2SINK works periodically and each period consists of three phases: SYNC broadcasting, noise sampling, and data communication with durations denoted as \( T_{syn}, T_{sen}, \) and \( T_{com} \) respectively (Figure 6). Each period is started by the Sink node broadcasting a SYNC frame, which contains a monotonically increasing sequence number and the current time of the sink. Every node re-broadcasts the SYNC after receiving a copy of it, and sets the sender of that SYNC as its \textit{predecessor to sink}, thus a \textit{passive route} to Sink is established for every node throughout the network. Then the node goes into noise sampling phase which has been described in previous section. During this phase, the node turns off its radio transceiver so that the RF electromagnetic interference to the sampling circuits is eliminated.

Each period is 5 seconds. The SYNC-broadcasting phase is no longer than 20ms and negligible. Sampling phase is 4 seconds and totally 32 125ms “Fast” RMS values are produced, leaving the last second for data communication. Each data frame contains 64 bytes RMS values (2 bytes for each 125ms) and some application layer overhead, resulting in a total of 96 bytes of payload are pushed to the IEEE 802.15.4 MAC layer.

VII. MEASUREMENT RESULT & COMPARISON

The final finish of our sensor node can be seen in Figure 7. The nodes are sealed by metal cover and water-proof material so that they can be deployed outside.

A. Power consumption

Instantaneous power consumption was tested by adding a 5.25\( \Omega \) resistor next to the power supply. We measured the voltage over the resistor and calculated the current using Ohm’s rule. Similar method has been used in [16], [17]. In order to indicate the phases mentioned above, an output pin is triggered when there is a phase transition.

In each period \( T \) the \( \mu \)C keeps reading ADC value and calculates the sound level. At the end of each period the radio module is turned on and a data frame is sent to the sink. When a sensor node is not scheduled to do radio communication, the radio module CC2420 is turned off to save energy. In this state the node consumes a constant, small amount of energy as shown in Figure 8(a). In general, the sensor nodes have sleep schedules. However, in this application the sensor nodes must keep sensing the noise thus no sleep mode is applied to \( \mu \)C. Nevertheless, we have observed less than 1mA current drain during the sleeping mode. We are interested in the real-time power consumption when the node is active and its radio is turned on. Totally five phases are needed to send a frame [18]:

\[
L_{PA} = ( olig(ADC) + 4.0135)/0.1227 \text{ (dB)}
\]
Figure 7. Final finish of the sensor node

1) Initialize radio module: the radio transceiver is turned on and the oscillator starts to oscillate. This phase takes a few milliseconds.

2) Frame formatting: a frame is copied to the radio buffer from \(\mu\)C. This phase duration varies, depending on the frame length.

3) Waiting RSSI: the radio module waits for a clearance of channel. This phase duration varies, depending on the channel situation.

4) Random backoff: IEEE 802.15.4 uses CSMA/CA medium access control (MAC), so a random backoff will reduce the chance of collision.

5) Sending the frame: the frame is sent out. The duration is proportional to the frame length. 

The above phases can be seen in Figure 8(b).

As the time transitions given by the time-indication-pin in Figure 8(b), the duration \(t_2 - t_1\) is the radio initialization phase, which takes approximately 5ms. There is an impulse power consumption at the beginning of this time when the radio circuits oscillator starts to run.

The second duration \(t_3 - t_2\) indicates the local communication between ATmega128 and CC2420. In our test, a total of 37 bytes (including MAC header 9 bytes, IP header 20 bytes and payload 8 bytes) were copied to CC2420. SPI (Serial Peripheral Interface) was set at a bus clock rate 7.7MHz/16 so that 37 bytes will take 0.615ms to transmit. Together with SPI bus commands, we have observed approximately 1ms for this phase. The power consumption raises 5mA from a floor of 30mA during this time.

The third time is short. This is the time that CC2420 acknowledges to the \(\mu\)C that frame buffer is ready.

The next duration \(t_5 - t_4\) is the channel access time. CC2420 must find a valid RSSI (Received Signal Strength Indicator) before the attempting of transmission. During this time the radio module keeps listening to the channel so a few power peaks are observed.

The duration \(t_6 - t_5\) is the random back-off time of CSMA/CA, and \(t_7 - t_6\) the frame transmitting time. During the transmission, the power consumption stays on a floor of 60mA.

At last, during \(t_8 - t_7\) the radio module is turned off.

In average, sending a frame consumes 36.74mA in approximately 10ms, and the entire cycle average is
26.90mA. Consider a typical AA-type battery with capacity of 2500mAh [19], it can support the node for $2500 \times \frac{3600}{26.90} = 334570$ seconds, which is nearly 93 hours, which satisfies the demand. Our field test experience has also proved this conclusion. Actually by network control/synchronization, the sensor nodes can be put into sleep mode, and wake up on demand. The network life time can be significantly increased by doing so.

B. Node cost

The cost of one sensor node is about 100 Euros. Comparing to the price of commercial sound meters in market, with the typical price for a class I sound meter 2000 Euros or more, this WSN application is very competitive.

C. Indoor accuracy test

All the nodes have been tested with Pulsar94, a type II sound meter, in a noise-immune room. The test configuration can be seen in Figure 9. Nodes are tested and their microphone pick-up points are very close to the sound meter 1.5m away from the sound source. The sound source is traffic noise recorded at Colin McRae Gathering down the M6, Sandbach [20]. In order to reduce the error, we put Pulsar94 at different places in the array. The test duration is 30 minutes with each configuration 5 minutes. The result is shown in Figure 10. The result shows that our nodes give very close data to that of Pulsar94. The mean standard deviation among our nodes is 0.91dBA throughout the all test phases.

D. Outdoor accuracy test

The field test was conducted on an open ground close to a motorway. We arranged 5 noise nodes in a line perpendicular to the motorway with 10 meters separation one each other, and Node 1 being the closest to the road. We put Cesva SC-20c together with each node and measured for 5 minutes. Because there is no synchronization between our network and SC-20c, we gave an impulse sound to mark the start of measurement. Figure 11 shows the results. We can see that our nodes have less than 2dB error for all the nodes.

VIII. CONCLUSION & FUTURE WORK

In this paper, the design of wireless sensor node for environmental noise measurement is described. Technical challenges during the design are reported and corresponding solutions are explained. The comparison with standard Type 1 sound meter is made and the objective of design is achieved. The power consumption is small enough so that a typical AA-type battery can support the nodes for several days of measuring without sleep. ATmega128 build-in A/D converter is enough for environmental noise measurement thus the cost of hardware platform is minimized.

The sensor node platform is built on ATmega128L with 4kB RAM, and its internal 10-bit ADC can operate a peak sampling rate of 33kHz. Both of these two resources have been pushed to their limits in our design. A better alternative should definitely have wider ADC range and more RAM. For this reason, we have decided our next development will
be continued on Jennic JN5139 platform, which has a 32-bit architecture, 16MHz clock, 12-bit ADC, and 96kB RAM.

REFERENCES


[18] TI CC2420 2.4GHz IEEE 802.15.4/ZigBee-ready RF Transceiver Datasheet, Texas Instruments.
