

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Development of a High-Resolution Wireless Sensor Network for Monitoring Volcanic Activity

José Chilo<sup>1</sup>, Andreas Schlüter<sup>2</sup> and Thomas Lindblad<sup>3</sup>

<sup>1</sup>*University of Gävle S-80176 Gävle, Sweden*

<sup>2</sup>*Institute of Computer Science, Freie Universität Berlin, Germany*

<sup>3</sup>*Royal Institute of Technology, S-106 91 Stockholm, Sweden*

## 1. Introduction

This chapter describes the design of a high-resolution wireless sensor network to monitor infrasonic signals from volcanic activity. A prototype system is constructed and tested. The system is based on the ultra low power microcontroller MSP430 with the requirements of energy-awareness and high sensor node autonomy. The infrasonic signals are measured at 200 Hz using 12 bit resolution and the result is buffered on SD cards in case of a lack of bandwidth. The implementation of a cost-table driven network routing protocol allows a radio sleep schedule of almost 97% when no data has to be transmitted. Furthermore, the sensors need to be time-synchronized for later event localization. This work shows that it is feasible to have a synchronization accuracy of less than 1 ms using a GPS receiver that is powered on only a few seconds per hour.

In recent years the installation of infrasound sensors at seismic measuring stations has become common and now researchers can obtain large and heterogeneous infrasound signal data-sets generated in near real-time. However, most current infrasound stations are still using expensive infrasound microphones and traditional data acquisition systems which limit the deployment of new infrasound stations. To improve on this, we propose in this work a wireless data acquisition system based on FreeWave FGR09CSU 900 MHz radio modem and a Wireless sensor networks (WSN).

Infrasound is defined as the range of frequencies between 0.001-20 Hz. It is generated by a variety of events, both man-made and natural. Among the latter type, active volcanoes are efficient sources of infrasound. Volcanic eruptions are characterized by the acceleration of hot fluids from subsurface reservoirs into the atmosphere generating acoustic waves in the 1-20 Hz frequency range. Infrasonic airwaves produced by active volcanoes provide valuable insight into the eruption dynamics and related phenomena. Infrasound also provides a special opportunity for the comparison of eruptive activity among different volcanoes because atmospheric pressure records are mostly independent of site-specific propagation effects (Chilo 2008).

However, infrasound propagating long distances is a complex phenomenon. It is strongly influenced by the detailed temperature and wind profiles. The infrasonic signal detected by

traditional infrasound systems contains the combination of the source's infrasound power spectrum and the distortions introduced by the atmosphere. In order to extract the source characteristics the data should be collected at close range: from a few meters to a few km distances. At short distances, the atmosphere is a homogeneous medium that preserves the infrasonic waveform as it is generated by the source. We need to seek new ways to enhance the capability of monitoring volcanic activity close to the source. Wireless sensor networks have the potential greatly benefit studies of volcanic activity.

A wireless sensor network is a collection of small devices having sensors, computational processing ability, wireless receiver and transmitter technology and a power supply (Culler, Estrin and Srivastava 2004). Typical WSNs communicate directly with centralized controller or a satellite, thus communication between the sensor and controllers is based on a single hop. Another kind of WSN could be a collection of autonomous nodes that communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner by forming an ad hoc network.

The last few years, the WSN has been used by a number of authors for volcanic eruptions monitoring. In references (Werner-Allen, Johnson, Ruiz, Lees and Welsh 2005; Werner-Allen, Welsh, Locrincz, Johnson, Marcillo, Ruiz and Lees 2006) a WSN was used together with infrasound microphones and seismometers for geophysical studies in the area of the volcano Tungurahua, Ecuador. The infrasound signals were sampled by 102 Hz, 10 bits resolution and transmitted over a 9 km wireless link to a remote base station. For the time synchronization a single GPS receiver was used in combination with the *Flooding Time Synchronization Protocol* (FTSP). The archived accuracy was 10 ms with an error of more than six milliseconds. The data transport was controlled by the remote base station.

In this approach we present a high-resolution WSN for long-term monitoring. The infrasonic signals are sampled and converted to digital data at a frequency of 200 Hz and a resolution of 12 bit. The proposed WSN requires a time stamp per infrasonic sample with an accuracy of one millisecond. Therefore, an algorithm was developed and evaluated which synchronizes the sensor nodes under the support of an equipped GPS device. The algorithm needs to be powered on the GPS device only a few seconds per hour. The collected data is handled under the concept of unlimited virtual data memory, whereby the data is swapped out to an SD card or, if the radio is switched on, transmitted towards the observatory. The used and evaluated routing protocol is the first implementation of the data-centric data dissemination protocol (D3). We extended this cost-based ad hoc routing protocol for the usage of radio time slots. This approach allows a sleep scheduled radio of almost 97% of the time.

The proposed WSN is planned to be deployed at University of San Agustin observatory station (ARE) in Southern Perú. The station is located 12.7 km south-east of Arequipa city, latitude S 16° 27'56.67443", longitude W 71° 29'35.23676" and elevation 2450 m. The ARE station provides a unique laboratory for studying regional infrasound and seismic wave propagation. The ARE station is located in the shadow of three giant volcanoes: Chachani (6075m), Misti (5821m) and Picchu Picchu (Chilo, Jabor, Liszka, Lindblad and Persson 2006). Furthermore, the most active volcano of Perú, volcano Ubinas, is situated 65 km from the ARE station which will be the focus for our infrasonic studies.

2. Hardware and requirements

The platform used is the Modular Sensor Board MSB-430 shown in Figure 1 (left). In this figure the top part (MSB-430S) is the sensor module; the middle part (MSB-430) is the core module; and the bottom part (MSB-430T) is the base module. The base module MSB430T carries three AAA batteries, has a JTAG and serial/USB socket and is available with a GPS receiver FALCOM Smart Antenna FSA01. It should be noted that this platform can easily be exchanged to a more suitable platform. The features and capabilities of the MSB430 concerning our proposal are summarized in Table 1.

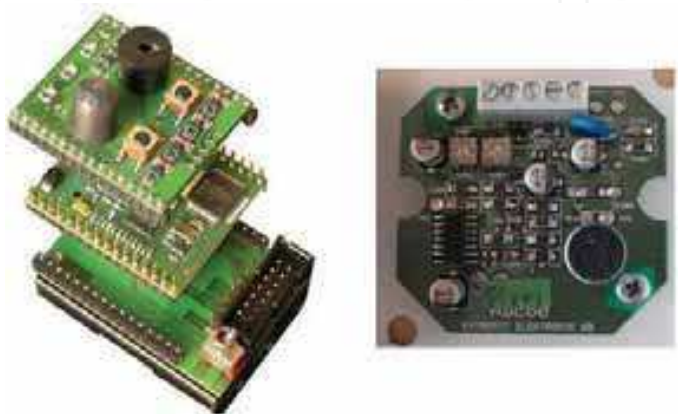


Fig. 1. Title Photo of MSB-430 (left) and the MCE-200 microphone with pertinent preamplifier and filter mounted on a PCB (right)

<b>Microcontroller MSP430F1612</b> 16bit RISC	
	100 kHz - 8MHz
	55 kB Flash-ROM
	5 kB RAM
	256B Infomemory
<b>Transceiver Chipcon CC1020</b>	
	868MHz
	8.6 dBm, max 1km (tuneable to more than 5 km)
	19.2 kbit/s using Manchester encoding
<b>Mass storage</b>	SD card (max 4 GB)
<b>ADC</b>	12 Bit, unipolar [0-3 V]
<b>Sensors on board</b>	
	humidity and temperature sensor Sensirion SHT11
<b>Supply voltage</b>	2.7-3.6V
<b>Energy</b>	Active Mode: 330 $\mu$ A at 1 MHz, 2.2 V
	Standby Mode: 1.1 $\mu$ A
	Off Mode (RAM Retention): 0.2 $\mu$ A

Table 1. MSB-430 features

For infrasound recording the electret condenser element microphone MCE-200 from Panasonic was used. The details listed in Table 2 are given by the manufacturer. The

mentioned frequency range is peculiar and seems not to cover the infrasonic range. Nevertheless, in contrast with the manufacturer specification, this microphone operates sufficiently in the infrasonic range according to the experiences of the authors (Chilo and Lindblad 2007).

The MCE-200 microphone is also available from Extronic AB (<http://www.extronic.se/>) mounted on a PCB, Fig 1, right part. A specially designed version includes a filter providing two bipolar output signals, one for infrasound and one for audible sound. The power supply is about 3-12 V. The PCB doesn't suit to be used as a sensor caused by high power consumption of more than 1 mA. We planned to use the microphone without this PCB, therefore, a tailor -made circuit needs to be designed including a low pass filter, a gain amplifier and an antialiasing filter.

Frequency range	20-16000 Hz
Sensitivity	7.9mV/Pa/kHz ±2 dB
Output impedance	1-2 kΩ
Signal-to-noise	ratio < 58 dB
Couple capacitor	0.1-4.7μF
Working temperature	0-40 °C
Power supply	1.5-10V DC /0.5mA

Table 2. Nominal MCE-200 Microphone specifications

The following incomplete enumeration briefly sketches the requirements of the application domain.

- 1) An event classification requires the complete signal opening of each infrasonic record.
- 2) The record of an infrasonic event per microphone shall be automatically transported from volcano Ubina to the University of San Agustín observatory station (ARE). The distance is about 65 km.
- 3) The operating time of the monitoring system is weeks or permanent.
- 4) The system shall automatically handle the failure of sensor nodes. The system will continue at least under a restricted operating mode following the malfunction of multiple nodes. It means that a complete system breakdown should be avoided.

The third requirement is in contradiction to the first one, because it forces a continuous AD conversion. Therefore, the sensor nodes will quickly run out of power. A trade-off is the continuous conversion by just a single node. The remaining nodes could use a more energy conserving comparator.

To analyze the wave propagation and in order to localize the infrasonic source six time synchronized signals recorded from different positions are required:

- 5) At least six microphones shall be deployed close to the volcano. They shall be spatially separated.
- 6) The chronological, accurate to a millisecond, and spatial position, accurate to 10 meters, of all records per microphone of an infrasonic event shall be available.



### 3. System design

#### 3.1 System overview

The data transport requirements (about 65km distance) affect materially the hardware selection and distribution. The transmission range of the MSB-430 mounted radio goes about 5000 m, but only under special circumstances like a clear line of sight. Hence the node has to be combined in a multi hop network a more powerful transmitting devices must be used. The FreeWave FGR09CSU 900MHz radio modem (<http://www.freewave.com>) could be used for this task.

According to the manufacturer specifications the FGR09CSU modem reaches a range of 95 km in a clear line of sight. If no line of sight is possible, then either a modem can be used as a repeater or additional intermediates allow a more suited gateway position. An example deployment using the FreeWave modem is sketched in Figure 2. In the example seven infrasonic sensor nodes (green) are used for the data acquisition, one gateway node (magenta) is connected to a workstation located in the ARE observatory and one intermediate node (blue) pass the data from the sources to the sink, the gateway node, which finally delivers the data to a workstation.

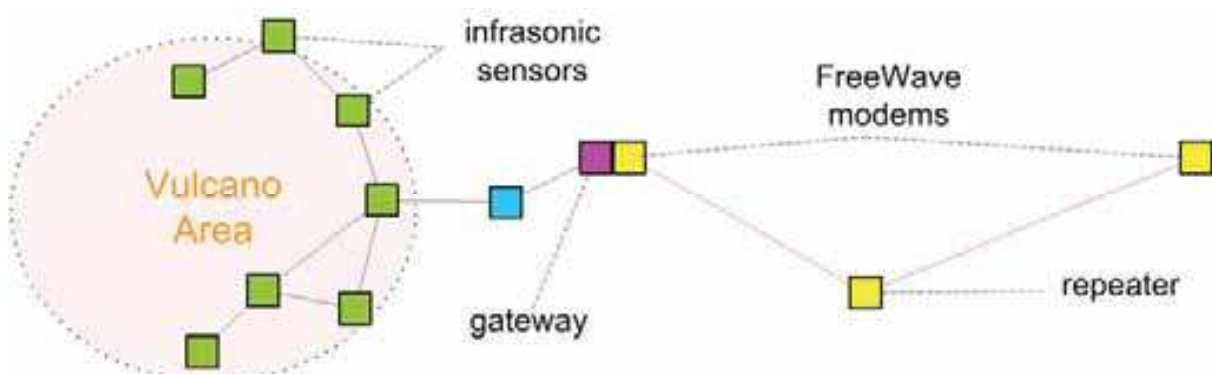


Fig. 2. Spatial system arrangement of a hybrid network; red line represents a long range link; black lines accords a radio coverage between two nodes

As aforementioned, Figure 2 contains seven infrasonic sensors, one more than needed. Not all sensors have to be simultaneously active. So, one can rest and maybe recover energy. Not every infrasonic sensor in the figure is inside radio coverage of an intermediate. In order to transport the data of the outer infrasonic sensors all of these must be intermediate nodes simultaneously. To assure sufficient CPU power for the data acquisition this double role needs to be controlled.

It is also vitally important that a failure of one or two nodes must not impact the whole system, thus the architecture shall actualize highly autonomous behaviour of the nodes not just for the day-to-day business but also for the handling of exceptions like malfunctions of some nodes. Furthermore, low power consumption and energy conservation strategies shall be taken into account, which is essential for a long term operation.

#### 3.2 Architecture

The applied method for the system decomposition covered both: the partitioning of the

system according to the system functionality (distinction of concerns), and to map the components to the required hardware.

Figure 3 shows the component dependencies of a sensor node. Time synchronization using GPS and periodical signalization form the Time component. For instance, the analog-digital-converter (ADC) is controlled by a periodical signal. The converted data is evaluated by the DataProcessing component (DP) in order to detect events of interest. The component gathers all needed data for an infrasonic record including the time and position information provided by the Time component and hands it to the Network component. Both components, DP and Network, require major RAM parts, as they handle the same data. The VirtualDataMemory component (VDM) manages RAM and SD card blocks. The current voltage level is periodically watched by EnergyController. Depending to the remaining energy and in collaboration with the neighboring nodes the component set the node into a low-power mode by stopping the Network, the Time and indirectly the DP component. The hardware abstraction functionality is pooled within the System component.

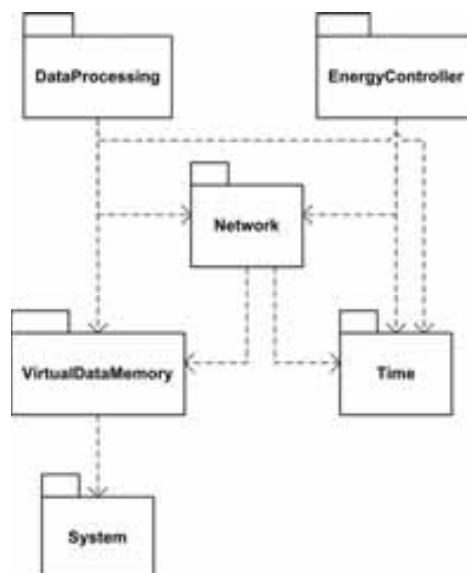


Fig. 3. The diagram gives an overview of the system architecture. Arrows symbolize interface access

## 4. Time synchronization

### 4.1 Hardware Mapping

The MSP430 version F1216 provides two timers, *Timer\_A* and *Timer\_B*. *Timer\_A* has three (TACCR0–TACCR2), *Timer\_B* has seven independent capture/compare registers (TBCCR0–TBCCR6). A timer register can be used in two different ways. Either the register's content is compared to the current counter value of the timer and creates an interrupt on equality (compare mode), or on an external signal, e.g. the rising edge of the GPS pulse, the current counter value is stored in the register (capture mode). *Timer\_A* is configured to count the oscillations of the sub-main clock (SMCLK), which itself will be fed by the digitally controlled oscillator (DCO, Figure 4) and *Timer\_B* counts the crystal watch oscillations

(ACLK, 32768 Hz). In order to provide a stable main clock the DCO is watched and controlled according to the more reliable crystal oscillator (dcoChecker).

The second Timer\_B register in compare mode is used to create the signal for the ADC. The third register is used to measure the GPS pps. In detail, the GPS signal captures the Timer\_B counter value, stores it into the third register and wakes up the CPU, i.e. an interrupt service routine (ISR) of the Time component.

For scheduling tasks like a periodically GPS synchronization the functionality of software timers is needed. Software timers shall run either after a defined delay or at a defined point in time. For an accurate scheduling two values are required, the number of Timer\_B cycles and the desired value of Timer\_B, i.e. ACLK cycles. When the defined Timer\_B cycles of a software timer expires, the desired watch oscillator cycles are written to the sixth compare register, which produces an interrupt if Timer\_B reaches the value and the software timer is executed.

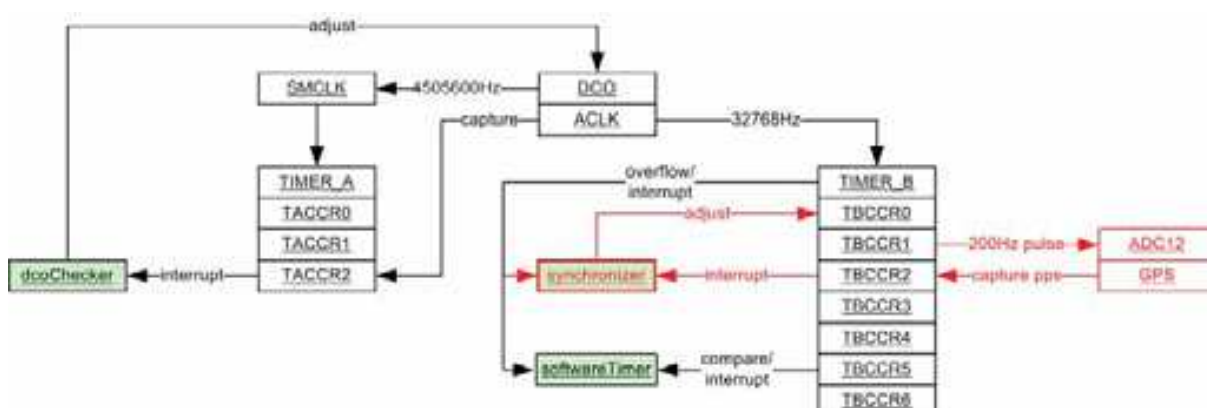


Fig. 4. The diagram shows time synchronization via crystal watch. Green objects are interrupting service routines; red objects are available only on sensor nodes

#### 4.2 GPS-ACLK synchronization

The idea of the ACLK synchronization is to let the timer count from zero to exactly the ACLK frequency decreased by one. Though, the exact ACLK frequency, about 32678 Hz, needs to be measured by using the GPS pulse-per-second (pps) as reference which provides an accuracy of 1  $\mu$ s. The real ACLK frequency depends to the environment temperature. The interrupt TBCCR0 (for TBR=32767) and CAP2 (for pps capture) occur simultaneously. This fact complicates the deviation detection. The execution of an interrupt service routine may not follow the real chronological event order. Therefore, it is wise to move the pps capture reference from zero to  $(TBCCR0+1)/2$ . Thus, the interrupts are separated by 0.5 s, but only if the system is synchronous. In other words, if the Timer\_B is synchronized, the pps is caught exactly by  $(TBCCR0+1)/2$ . Moreover, the ACLK ticks between the first and the second pps capture are counted, what should be close to 32768. This is the first synchronization stage.

To illustrate the procedure in detail we sketched a simplified example in Figure 5. In this figure each timer counting step is visible. One relevant detail is the fact that if the timer counts to TBCCR0 it altogether counts TBCCR0+1 ticks; as the timer value is in the range of  $[0..TBCCR0]$ . Before the first capture occurred, the timer was initialized by  $f_{ACLK_1} = 9$  or





As a corrective the places after the decimal point for the rational ACLK frequency are approximated by measuring the average deviation for  $\Delta seconds_n$ . Through the usage of the integer frequency  $f_{CLK}$  the deviation accords the fractional part of the rational ACLK frequency multiplied  $\Delta seconds_n$ . The fraction calculation is done within the second stage of synchronization. First after the third capture the GPS device is switched off and the captured values (captured ticks and local time in seconds) are stored. When for instance tens of minutes elapsed the GPS device is activated again and the fourth capture is awaited. Equation 3 calculates the average fraction, whereby  $\Delta ticks_n = capture_n - capture_{n-1}$ .

$$fraction_n = \frac{\Delta ticks_n}{\Delta seconds_n} \quad (3)$$

The frequency correction can be simply done by adding uniquely  $\Delta ticks_n$  every  $\Delta seconds_n$  to TBCCR0 and restoring the original value of TBCCR0 one second later. However, this approach would lead to unwanted steps within the time response. Much better is the method of equally distributing  $\Delta ticks_n$  within  $\Delta seconds_n$ . The resulting value for TBCCR0 is calculated on each timer overflow event like in equation 4. The variable  $t$  is the time in seconds after the last phase shift ( $t = 0s$ ).

As  $fraction_n$  always is within the range  $(-1..1)$ , the timer border  $TBCCR0(t)$  equals either  $f_{CLK_n} - 1$  or  $f_{CLK_n} - 1 \pm 1$ . In other words: the correction of  $\Delta ticks_n$  within  $\Delta seconds_n$  is implemented as an equal distribution of atomic corrections for the timer period.

$$TBCCR0(t) = f_{CLK_n} - 1 + \lfloor (t+1) fraction_n \rfloor - \lfloor t fraction_n \rfloor \quad (4)$$

On the fifth pps the captured values needed for the next GPS adjustment are stored and the device is powered down for maximum  $16 * \Delta seconds_n$  (on a constant frequency). One open point is the measuring of  $fraction_n$  while concurrently the timer is adjusted by  $fraction_{n-1}$ . Equation 3 needs to be extended to equation 5 to take a concurrent adjustment into account.

$$fraction_n = \frac{\Delta ticks_n + \lfloor \Delta seconds_n fraction_{n-1} \rfloor}{\Delta seconds_n} \quad (5)$$

First tests showed a non acceptable result. A failure analysis discovered a dependency of the watch crystal on the environment temperature. Within a couple of hours the oscillator slowed down almost linear. The third synchronization stage forecasts the expected ACLK frequency taking the gradient of the frequency response into account.

The gradient is calculated in equation 6. The gradient goes into equation 4 and equation 5 to form equation 7 and equation 8.

$$gradient_n = \frac{(f_{ACLK_n} + fraction_n) - (f_{ACLK_{n-1}} + fraction_{n-1})}{\Delta seconds_n} \quad (6)$$

$$TBCCR0(t) = f_{ACLK_n} - 1 + \left\lfloor (t+1) fraction_n + (t+1)^2 gradient_n \right\rfloor - \left\lfloor t fraction_n + t^2 gradient_n \right\rfloor \quad (7)$$

$$fraction_n = \frac{\Delta ticks_n + \left\lfloor \Delta seconds_n fraction_{n-1} + \Delta seconds_n^2 gradient_{n-1} \right\rfloor}{\Delta seconds_n} \quad (8)$$

The synchronization stages are sketched in the activity diagram in Figure 6. Stage one happens after the second capture but only for the first time a node is activated. Between the third and fourth capture the GPS is powered down for a long time. For the best trade off between energy consumption and accuracy the time duration is determined by test series. On the fourth capture stage one and two are executed. The calculation of the gradient results zero, hence a prior fraction is missing in the current state. The next occurring capture is treated as the third capture again. The second time a *fourth* capture happens, the gradient can be determined.

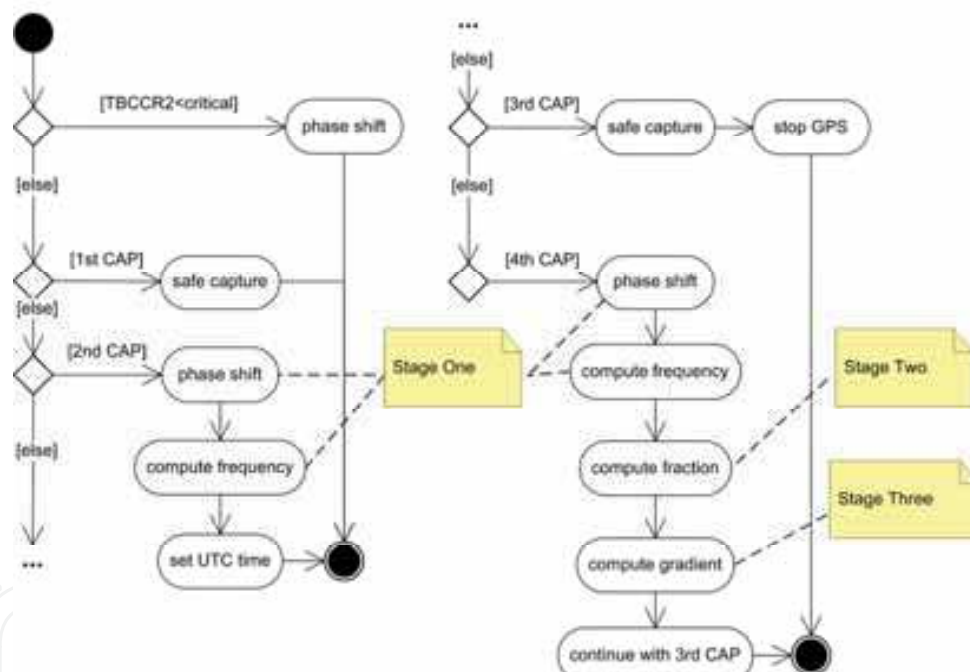


Fig. 6. Final activity diagram of handling CAP2 events

## 5. Data Acquisition

### 5.1 Infrasound data

The infrasound sensor is realized by the electret condenser element microphone (ECEM) MCE-200 from Panasonic. A PCB will connect the microphone to the ADC12. Therefore, the signal needs to be transformed to a unipolar signal up to 3V.

The complete data pipeline is arranged in a way that first the analog signal produced by the ECEM is filtered to pass only frequencies below the cutoff frequency and afterwards the

signal is gained to the required voltage, Figure 7. This approach avoids an overmodulation by disturbing frequencies.

To date there are not ideal low-pass filter, i.e. passing the signal below 20 Hz and eliminating the frequencies above. Real filters are approximations, for instance the *Bessel*, *Chebyshev*, and *Butterworth* filter.

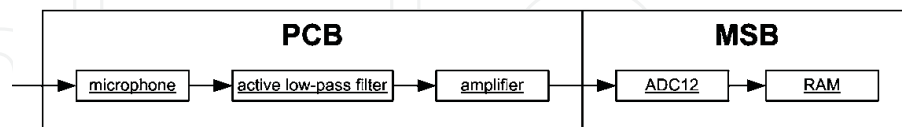


Fig. 7. Data acquisition pipeline

A Butterworth filter was designed in our case. The quality of an active filter, for instance a steeper attenuation, is improved by cascading filter stages by combining multiple single filters. A 4th order low-pass filter was realized which will require two operation amplifiers on the PCB.

## 5.2 Local event detection

To detect a supposed volcanic eruption a simple approach is watching the signal amplitude. If it passes a threshold one can assume an eruption occurred. The maximum of the amplitude is searched in time window  $T_w$ . For the detection of a 1 Hz signal  $T_w$  must be at least 500 ms. If the threshold is too high, the amount of false negatives, i.e. missing an event, increases. If it is too low, more false positives will be detected, for instance short-term fluctuations or wind.

Short-term fluctuations can be smoothed out by calculating a moving average. The EWMA-detector implemented by (Werner-Allen, Johnson, Ruiz, Lees and Welsh 2005) supplied more reliable events than their implemented threshold based detector. The EWMA (exponential weighted moving average) function

$$average_t = average_{t-1} + \alpha(sample - average_{t-1}) \quad (9)$$

computes the averages by fading older samples. The variable  $\alpha \in [0..1]$  determines the fading factor of old samples. A high  $\alpha$  fades off more. The elimination of long-term trends can be done by maintenance two averages, by name a short-term  $average_s$  and a long-term average  $average_l$ . If  $average_s$  exceeds  $average_l$  by the amount of  $div$  an event is triggered.

The state machine in Figure 8 maps the distributed event detection. To trigger the data acquisition two independent events must occur within a time window. The elapsing of a timeout without an event always leads back to the initial state.

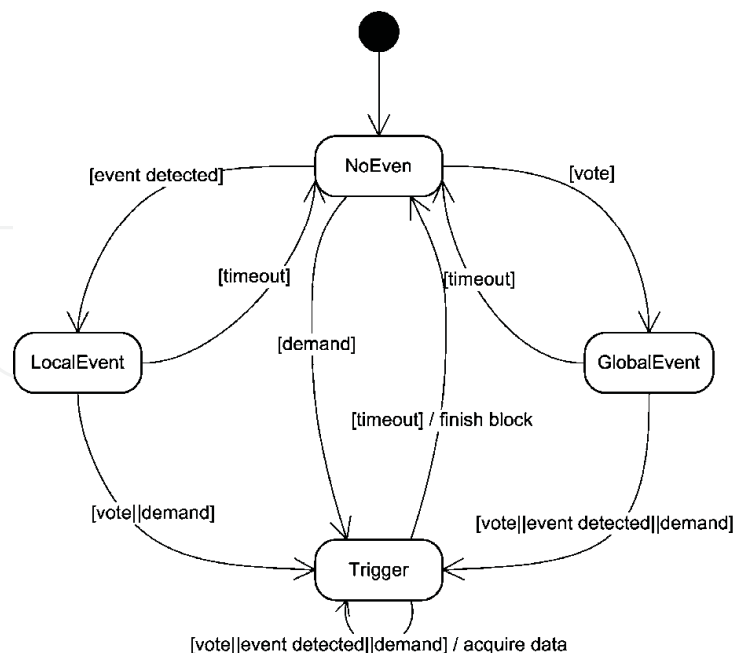


Fig. 8. State machine for distributed event detection

All state transitions are briefly described in Table 3. The demand message is put into brackets because of the following: if a sensor does not receive two votes it is probably too far away, that it could sample meaningful data, hence, the demand message could be discarded. However, the implementation efforts for this functionality are not high and may still finally benefit.

State	Description
Transition	
<i>local event</i>	local event detected
<i>vote</i>	dialog message received by a neighbour who detected an local event
<i>(demand)</i>	dialog message forces the start of data acquisition
<i>timeout</i>	time without any event elapsed

Table 3. List of state transitions used in Figure 8

## 6. Data transport

The implemented routing protocol D3 (Ditzel and Langendoen 2005) is a cost table and data centric routing protocol, i.e. no network addresses are used and the data is “floating” down a gradient. Therefore, only broadcasts are used. In respect to energy conservation, the medium is accessed in time slots, small ones for negotiation and large timeslots for the data transfer whereby only the participating nodes are active. The problems of hidden and exposed stations are no more solvable by the classical MACA protocol. Instead we used three different data timeslots: TXDATA for transmission, RXDATA for reception, and IDLE for a powered down radio. An example is given in Figure 9. All nodes with the same hop-



count distance to the data-sink, i.e. cost, need to use the same slots.

Figure 10 shows the general idea of the D3 protocol. First the gateway, the sink, broadcasts an interest message, Figure 10(a). Each node updates its own cost, i.e. it stores the smallest received cost plus one and forwards one-time the INT message with the node's own cost. The data sink labels the interest message by a sequence number. If a node receives an INT message with a new sequence number, it overwrites its cost value even with a higher cost. In this way a changing node topology can be recognized by regular INT messages.

Before a node transmits a data message, it advertises the data by broadcasting an ADV message in the ADV slot before the next TXDATA slot, node E in Figure 10(b). The message contains the cost of the node (here two for node E) and the segment\_key\_t values of all unconfirmed packets in the memory. In the original D3 protocol the message also includes the time of data transmission, i.e. here the duration to the next data slot in ACLK ticks. However, the transmitted duration should be ideally the time between the first byte received by an interested node and the beginning of the data slot. To approach this value as close as possible the time of transmitting the first ADV byte needs to be measured by the initiating node and the duration should be computed according this value. This is only possible by transmitting the duration by a different message immediately after an ADV message; called *Time* message. To map a Time message to the responding ADV message, both messages need the node network address.

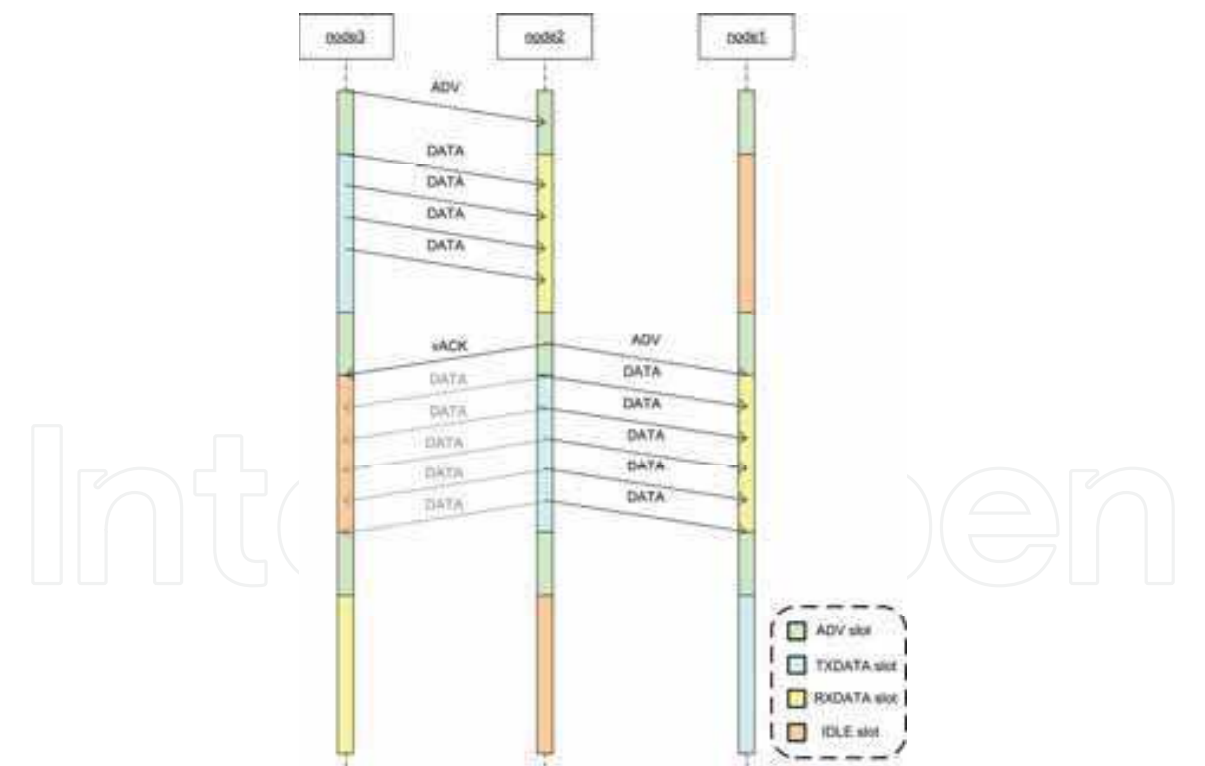


Fig. 9. Sequence diagram of D3 timeslots

Each neighbour with lower costs than the cost of a received ADV message schedules the data reception (node C and D). All other nodes just switch off their radio in the next RXDATA slot (node F and G). The data is transmitted at the beginning of the RXDATA slot, Figure 10(c).

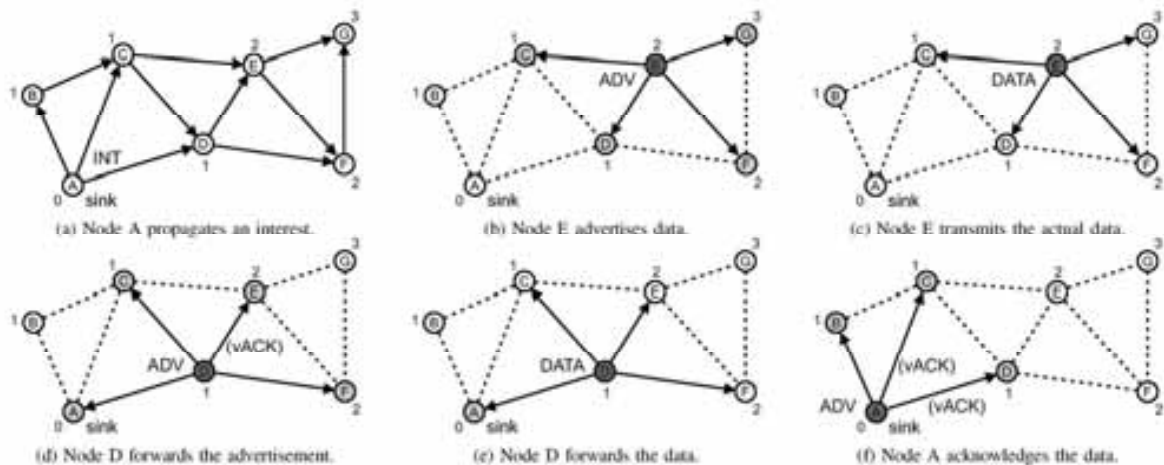


Fig. 10. The D3 routing protocol. Initially the gateway floods an INT message. A node forwards data by advertising the new data and after transmitting the data itself

The next data slot is an IDLE slot for the prior transmitter (node E) and a TXDATA slot for the prior receiving nodes (node C and D). Those nodes initiate a random backoff delay at the beginning of the ADV slot foregoing its TXDATA slot. The node for which the backoff delay expires first, node D in Figure 10(d), immediately broadcasts an ADV, which lets all other nodes with an advertisement intention cancel their backoff delay. Again, the message contains the nodes own cost (one for node D) and all segment identifications of the packets in the memory, which includes also the quite recently received packets. Furthermore the prior forwarding node (node E) takes that ADV message as a virtual acknowledgment (vACK).

Finally, the data message reaches the gateway, Figure 10(e). The gateway also advertises its data in its according ADV slot, Figure 10(f). Of course, in its following TXDATA slot it does not forward the data by the radio, but by the COM port to a gateway client.

The usage of explicit acknowledgments (eACK) is need fully to avoid the retransmission of plenty big data packets. The eACKs are transmitted immediately when a neighbour node advertises already confirmed packets.

Another scenario is the resetting of a node or the adding of a new node to the network. The ADV messages do not suffice for the integration of the node into the topology, because the following could happen: Let's assume a new node has the cost of five, but before the node doesn't receive an ADV message of a neighbour with the cost four, its cost is unknown. Now, this node receives an ADV message with the cost six and assumes a cost seven, i.e. the cost plus one. This lets also confirm all advertised packets in its history. In the case the node receives an ADV message with cost four; it updates its cost to five. If now the node who first advertised data needs to advertise the data twice, the new node assumes the data to be already confirmed and wrongly transmits an eACK. The data packets will be lost. A solution is the usage of interest request messages, which are transmitted the first time a node enters the network or frequently until the node is added to the network. All neighbours answer on an interest request with its last received interest message. However, the requesting node can use the ADV messages together with the time messages to synchronize to the network time slots, so it could transmit the request within an ADV slot.

## 7. Results and conclusions

### 7.1 Data processing component

A prototype node is equipped with a specially designed PCB containing the infrasonic microphone, a 4th-order Butterworth active low-pass filter, and an amplifier. For the power supply a 3 V source is used. A function generator is connected to the couple capacitor and ground. It produces a sinus wave with an amplitude of  $U_{i-max} = 344$  mV. The PCB output pin and ground is connected to a voltage oscillograph. The amplitude of the PCB output signal  $U_{o-max}$  is measured. The output signal swings around 1.5 V.

The gain  $G_{PCB}$  of the circuit depending to the input frequency  $f_i$  is computed by the equation 10, Figure 11, green curve.

$$G_{PCB}(f_i) = \log\left(\frac{U_{o-max}(f_i) - 1.5V}{U_{i-max}}\right) * 10dB \quad (10)$$

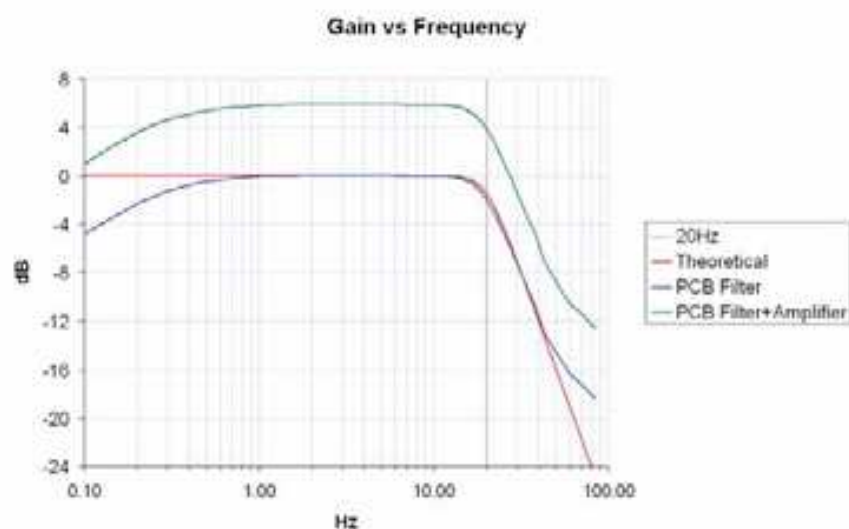


Fig. 11. The response of the amplifier. Red, theoretical 4<sup>th</sup>-order Butterworth; blue, filter without amplifier and green, filter plus amplifier

To measure the gain of the AC amplifier we used a frequency which is passed by the filter with unity gain. In other words we searched the output voltage maximum depending to the input frequency, what is by  $f = 2.2$  Hz and  $U_{o-max} = 2.844$  V. The gain at this point is:

$$G_{amp} = \frac{2.844V - 1.5V}{U_{i-max}} = 3.95 \quad (11)$$

The calculated value is  $G = 4.25$ . To get the gain  $G_{filter}$  assigned to the filter the amplifier gain must be undone, Figure 11, blue curve:

$$G_{filter}(f_i) = \log\left(\frac{U_{o-max}(f_i) - 1.5V}{3.95U_{i-max}}\right) * 10db \quad (12)$$

The amplitude response of a 4th-order Butterworth filter is plotted for comparison, Figure 11 red curve. Two obvious deviations are visible. The first one below 1 Hz is explained by the high-pass effect of the couple capacitor and maybe even by the high-pass effect of the capacitor of the AC amplifier. The second one close to 100 Hz can be a measurement error caused by the fact that the output voltage in this range is nearly the offset DC of 1.5 V.

The cutoff frequency is defined as the frequency for which the filter returns  $\sqrt{1/2}$  of the pass-band voltage. The cutoff frequency  $f_c$  can be experimentally determined by finding the frequency which fulfils following equation:

$$\log(\sqrt{1/2}) * 10dB = -1.51dB = G_{filter}(f_c) \quad (13)$$

The realized cutoff frequency is  $f_{c-low} = 19$  Hz. The cutoff frequency of the high-pass effect can be determined in the same way and is  $f_{c-high} = 0.14$  Hz which is acceptable.

## 7.2 Time component

The three staged GPS synchronization allows a switched off GPS device for long periods ( $t_{GPS\_OFF}$ ). The time period impacts not only the energy consumption but the RTC accuracy. To find the best trade-off between energy consumption and accuracy test series with different  $t_{GPS\_OFF}$  values are accomplished. The board is simply made of a single sensor node equipped with a GPS device. It is configured to start the stage one synchronization immediately, stage two after ten minutes, and stage three after further ten minutes. Firstly now, the time period  $t_{GPS\_OFF}$  is used. Hence, the GPS device needs between 45 s and about 165 s to provide a valid pps signal, the synchronization period is  $t_{GPS\_OFF}$  plus the fluctuating GPS startup time.

Figure 12 shows the result for  $t_{GPS\_OFF} = 45$  min. The test runs more than 15 hours. One can extract different assumptions by interpreting the responses. The first fact is that within the first three hours the synchronization is unacceptable. The reason could be the temperature difference between the office and outside, thus the node needs some time to acclimatize to the outside temperature. Furthermore, the outside temperature declined in the evening hours.



Fig. 12. State 15 hour test result of the RTC deviation; three staged GPS synchronization happened about every 45 minutes

Incontestable is the dependence of the deviation between the expected and the real frequency response to the RTC deviation. A bend in the frequency response leads to a deviation. However, the idea to take the second derivation into account would worsen the prediction of the frequency. One can appreciate this, if the expected frequency response is imaginary extended by the turns of the real curve. The mathematically unpredictable bends can maybe be physically predictable, if for instance the temperature is monitored. A future investigation of dependence of the frequency and temperature could pay off.

Summarized, 33.7% of the time the RTC exceeds the allowed limit (ignoring the first three hours, 22.1%).

For the experimental assurance of the desired accuracy the time  $t_{GPS\_OFF}$  needs to be defined according to the highest gradient, what is = 648 s, equation 14, ignoring the first three hour.

$$t_{GPS\_OFF} = 0.5ms / |gradient_{max}| = 648s \quad (14)$$

The average gradient, equation 15, of the deviation takes both into account: long periods of small deviations and the percentage part of peaks of high deviations. A fixing of the time  $t_{GPS\_OFF}$  according to the average gradient, equation 16, allows an exceeding of the allowed deviation for a small time. For the average gradient without the first hour the result is  $t_{GPS\_OFF} = 2150s$ .

Very optimistic but energy conserving is to take only 75% of the smallest gradients into account. The resulting value according to the average gradient of 75% of the best values is  $t_{GPS\_OFF} = 4168s$ .



$$\overline{\text{gradient}} = \frac{\sum_{\min < i \leq \max} \frac{|\text{deviation}_i|}{\Delta t_i}}{\max - \min} \quad (15)$$

$$t_{\text{GPS\_OFF}} = \frac{0.5 \text{ ms}}{\overline{\text{gradient}}} \quad (16)$$

An adjustment every 30 minutes, Figure 13, doesn't show a considerable improvement. It needs two hours to acclimatize and to calculate the gradient to the realistic temperature response. After the first two hours the time of exceeding the limit deviation of 0.5 ms is about 19.9%. A shorter time period for the GPS adjustment doesn't solve sufficiently the deviation problem caused by the turns in the frequency response.

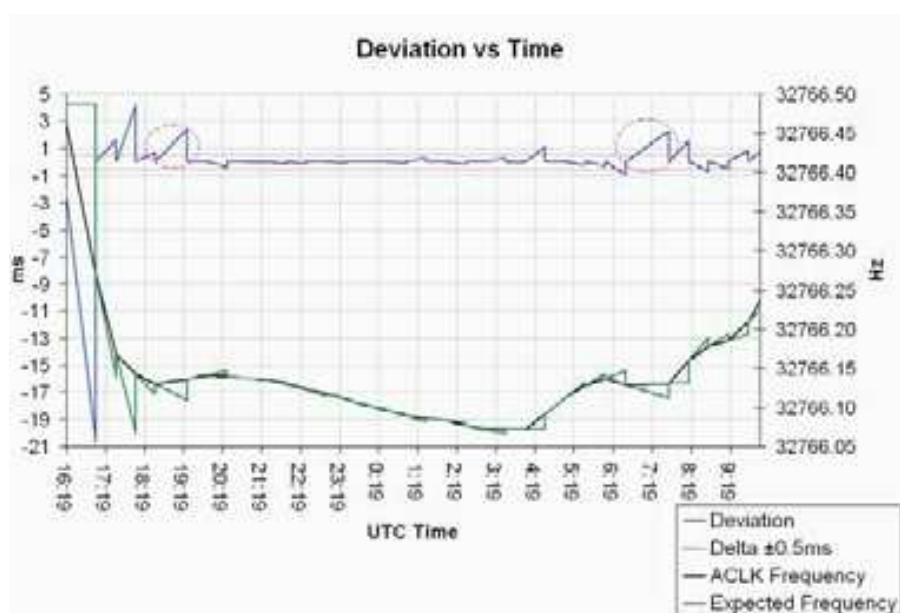


Fig. 13. 18 hour test result of the RTC deviation; three staged GPS synchronization happened about every 30 minutes; pink cycles - GPS timeout caused by rain

For a complete evaluation of the measurement the knowledge of an outstanding fact is required. The pink cycles mark long time periods of a deficient GPS signal, which lasts respectively about an hour. Both GPS timeouts were caused by rain. Rain impedes the synchronization process twice. On the one hand, it induces GPS timeouts and on the other hand, it comes together with an increasing of temperature, which itself speeds up the ACLK. Nevertheless, excluding these values would result a violation of the accuracy requirements for about 12% of the time.

### 7.3 The network layer

The issues for the network layers are the capacity, the reliability, and the robustness. Two types of test scenarios were accomplished: the transmission of a continuous data stream and a sparse transmission of data records over a long time period.

For both tests the command *cost* was implemented on the gateway node in order to simulate a cost distribution. The first parameter of the command addresses a node. The second one specifies the desired cost of the node. By receiving a dialog message containing the command the destination node adjusts its cost accordingly. While the test application is running, the node ignores messages transmitted by nodes whose cost difference is greater than one.

Unfortunately, a second node equipped with a second GPS device was not timely available. Due to the fact that the network time slots are seeded by time synchronized nodes, the tests can only be done with a single data source. However, the intermediate nodes don't care for the packet originator and multiple data sources are not essentially required to test the functionality.

Four different configurations for the breadboard were realized. Different network depths and different numbers of nodes on the same level were tested. Of course the data sink was always on level zero. The data rates are measured by the GatewayClient.

data rate [kbit/s]	Volume [kB]	cost level 1	cost level 2	cost level 3
2.35	530	one data source		
2.26	502	one intermediate	one data source	
1.37	292	two intermediates	one data source	
2.09	524	one intermediate	one intermediate	one data source

Table 4. Test results of routing a continuous data stream

Table 4 shows the data rates measured for the different network topologies. The packet loss rate (not listed in the table) for all topologies was 0%. The rate for the two node topology (line one in the table) complies exactly the theoretical value of equation 17 (one RAM block of 512 Bytes contains four network packets; one network packet carries a payload of 110 Bytes). The data rate is the measured value for two intermediates on the same cost level. Obviously, the small value is reasoned by collisions. This indicates a small value for the used backoff delay. Anyhow, the results still fulfil the requirements.

$$datarate_{MAX} = \frac{4blocks * 4packets * 110Byte}{3 * t_{ADV} + 3 * t_{DAT}} \approx 2.35kbit/s$$

(17)

By the following test case the reliability of the network synchronizations is measured. During the periods of no network traffic, no synchronization of the time slots happens. The network topology was linear with a depth of three, i.e. the data source was on cost level three. The size of the records was randomly created.

test duration [min]	record separation [min]	volume [kB]	records
124	15	51	8
185	30	33	7

Table 5. Test results of routing periodically single records

A delay of 15 minutes between the record transmissions results a stable time slot behavior of the network, Table 5. However, for the delay of 30 minutes the networks got asynchronous after the 7th record. Another test with a delay of one hour (not listed) failed after the first record. It implies, that the time slot deviation after 30 minutes can exceed at least the duration of the half of the ADV slot duration, i.e. 94 ms or 3072 ACLK ticks. In other words, if the crystal oscillators of the intermediates deviates about 3.4 ticks per second, the ADV slot duration is exceeded after 30 minutes.

In order to increase the allowed deviation the ADV slots could simply be extended. Even, periodical transmissions of ADV messages would synchronize the nodes again. However, both approaches would increase the network energy consumption.

Another solution is the flooding of the measured frequency and its expected trend by the sensor nodes immediately after a sensor node synchronized to the GPS device. Therefore, it is assumed, that the oscillators of all nodes work under the same conditions.

Considering very long time periods without any network traffic, following strategy is promising: If the nodes got asynchronous it firstly matters in the case of transmission purposes. To synchronize again the acting node transmits its slot time multiple times with a delay, which assures that the message is received in at least one ADV slot by a child node. A delay of about the half of the ADV slot duration would suit. The acting node is *scanning* the time slots.

Though the unsatisfying result the implemented D3 routing protocol works in principal. A detailed analyzes and thereby the adjustment of the protocol parameters is the direction of future work, as well as large scale tests.

7.4 Conclusion and outlook

The system presented in this chapter has the potential of a full-fledged application. The analysis of the requirements, as well as the design of the system architecture, was done using common and well proven software engineering techniques. Whereby, the balancing act between the reutilization and the easy interchange ability in opposite to the high application awareness succeeded for the most parts. Though the not entirely satisfying results and the open questions, the design is reasoned and self contained.

In order to connect the microphone to the MSB a sensor module was created. The circuit covers a fourth-order Butterworth low-pass filter and an amplifier. Furthermore, the signal is transformed from bipolar to unipolar. The comparison of the measured frequency response and the expected theoretical response shows acceptable results. A direction for future work is to design a digitally controlled amplifier. This would open the possibility to remotely adjust and fine tune the event detection on hardware level. In the same way, the problem of a missing infrasonic reference source for the best suiting amplifying factor could be avoided.

The time synchronization was one of the big challenges. We showed the feasibility of achieving the required accuracy by a rare use of the GPS device. The implemented approach comprises a high potential to get improved. In view of the measurement results, the prediction of the crystal oscillator frequency by taking the current temperature into account is promising. Even the usage of an external and more stable oscillator would be an enhancement. The maximum achievable accuracy is determined by the GPS receiver and lies in the range of microseconds.

The time synchronization greatly benefits the data acquisition. On the one hand, the synchronized hardware timer controls the sampling frequency through the pulse-width modulation, what assures the desired sampling frequency. On the other hand, the timer determines the beginning of the sampling. So, the mapping of the samples to the UTC time is reliably done.

The conflict of the high CPU demands of both, the data acquisition and the data transmission, is solved by applying a mutex for the CPU. Anyhow, considering a long-term usage it is recommended to implement a hardware trigger and start the data acquisition only if a threshold is passed. Therefore, the requirement of the signal opening needs to be discarded. A trade-off is the continuously sampling by just a single node.

The presented strategies for the local and the distributed event detection can be applied and expediently tested firstly, when several infrasonic sensor nodes are available. Again, the missing of an adequate reference signal complicates a contingently ongoing development. Future work should concentrate on the remote parameterizing of the thresholds. The event detection is an own field of research. An improvement here would payoff by reducing the power consumption of the network.

Another big challenge was the implementation of the routing protocol in respect to the time slots. The procedure was very time consuming and the test results didn't show a sufficient behaviour. Anyhow, we showed the functioning in principle and submitted possible adjustments. The most promising approach is the scanning of the time slots in the case of asynchrony. The benefit of the sleep scheduled radio for almost 97% of the time still reasons such an approach. A detailed analyze and proving of the routing protocol can fill an own study.

The implemented concept of the virtual data memory was highly optimized to the demands of the accessing components. The SD card read/write operations are only performed when it is absolutely required. For the data acquisition a free memory block is always assured. The Network component finds enough free memory in the reception time slot, while in the transmission time slot the memory is filled by the packets to be forwarded. In order to serve gateway requests of missing packets a history maps the packets to SD card addresses. The thereby required trade-off between the history size and the SD card address limitation can be solved by considering the swapping out the entire history to the SD card.

Finally, designing a circuit for the regeneration of chargeable batteries by solar panels could be the essential step towards a real long-term usage.

Wireless sensor networks present many exciting opportunities. The developed system is easily to customize in order to operate in different applications. For instance, the orthogonal usage of the infrasonic microphones can be used to measure the resonance frequency of high buildings. The implementation of a burglar alarm is another example, as well as tracing of

moving objects by the time synchronized nodes. Nevertheless, exchanging the microphones to other sensors opens a broad spectrum of possibilities.

## 8. References

- Chilo, J. (2008). Low-Frequency Signal Classification: Filtering and extracting features from infrasound data, Verlag Dr. Muller, ISBN 978-3639113419.
- Chilo, J. ; Jabor A. ; Liszka L.; Eide A. J.; Lindblad & Persson L. (2006). Infrasonic and Seismic Signals from Earthquake and Explosions in Arequipa, Peru, Proceedings of Western Pacific Geophysics Meeting, Beijing China, 2006.
- Chilo J. & Lindblad Th. (2007). A Low Cost Digital Data Acquisition System for Infrasonic Records, International Workshop on Intelligent Data Acquisition and Advanced Computing Systems, pp. 35-37, IEEE Catalog Number 07EX1838C.
- Culler D.; Estrin D. & Srivastava M. (2004). Overview of wireless sensor networks, IEEE Computer, Special Issue in Sensor Networks.
- Ditzel M. & Langendoen (2005). D3: Data-centric Data Dissemination in Wireless Sensor Networks, IEEE Computer Society, The European Conference on Wireless Technology.
- Werner-Allen G.; Johnson J.; Ruiz M.; Lees J. & Welsh M. (2005). Monitoring Volcanic Eruptions with a Wireless Sensor Network, Second European Workshop on Wireless Sensor Networks, EWSN'05.
- Werner-Allen G.; Welsh M.; Locrincz K.; Johnson J.; Marcillo O.; Ruiz M. & Lees J. (2006). Deploying a Wireless Sensor Network on an Active Volcano, IEEE Computer, Sensor-Network Applications.

IntechOpen





## **Geoscience and Remote Sensing**

Edited by Pei-Gee Peter Ho

ISBN 978-953-307-003-2

Hard cover, 598 pages

**Publisher** InTech

**Published online** 01, October, 2009

**Published in print edition** October, 2009

Remote Sensing is collecting and interpreting information on targets without being in physical contact with the objects. Aircraft, satellites ...etc are the major platforms for remote sensing observations. Unlike electrical, magnetic and gravity surveys that measure force fields, remote sensing technology is commonly referred to methods that employ electromagnetic energy as radio waves, light and heat as the means of detecting and measuring target characteristics. Geoscience is a study of nature world from the core of the earth, to the depths of oceans and to the outer space. This branch of study can help mitigate volcanic eruptions, floods, landslides ... etc terrible human life disaster and help develop ground water, mineral ores, fossil fuels and construction materials. Also, it studies physical, chemical reactions to understand the distribution of the nature resources. Therefore, the geoscience encompass earth, atmospheric, oceanography, pedology, petrology, mineralogy, hydrology and geology. This book covers latest and futuristic developments in remote sensing novel theory and applications by numerous scholars, researchers and experts. It is organized into 26 excellent chapters which include optical and infrared modeling, microwave scattering propagation, forests and vegetation, soils, ocean temperature, geographic information , object classification, data mining, image processing, passive optical sensor, multispectral and hyperspectral sensing, lidar, radiometer instruments, calibration, active microwave and SAR processing. Last but not the least, this book presented chapters that highlight frontier works in remote sensing information processing. I am very pleased to have leaders in the field to prepare and contribute their most current research and development work. Although no attempt is made to cover every topic in remote sensing and geoscience, these entire 26 remote sensing technology chapters shall give readers a good insight. All topics listed are equal important and significant.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jose Chilo, Andreas Schluter and Thomas Lindblad (2009). Development of a High-Resolution Wireless Sensor Network for Monitoring Volcanic Activity, *Geoscience and Remote Sensing*, Pei-Gee Peter Ho (Ed.), ISBN: 978-953-307-003-2, InTech, Available from: <http://www.intechopen.com/books/geoscience-and-remote-sensing/development-of-a-high-resolution-wireless-sensor-network-for-monitoring-volcanic-activity>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China

[www.intechopen.com](http://www.intechopen.com)

51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

IntechOpen

IntechOpen

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen