Remote Monitoring of Geological Activity of Inclined Regions – The Concept

Jarosław Flak, Piotr Gaj, Krzysztof Tokarz,
Stanisław Wideł, and Adam Ziębiński

Silesian University of Technology, Institute of Informatics,
ul. Akademicka 2A, 44-100 Gliwice, Poland
{jaroslaw.flak,piotr.gaj,krzysztof.tokarz}@polsl.pl
{stanislaw.widel,adam.ziebinski}@polsl.pl
http://www.polsl.pl

Abstract. In the article authors consider the possibility of using current IT technologies to create a computer system dedicated to monitoring geological phenomena. Nowadays, there are a few technical solutions dedicated to monitoring various kinds of events. Some of them are standalone devices, other a little bit more complex systems. After seminar, realized together with geologists within the GIT project[1], the conclusion was born that there are no full-scale, comprehensive systems which can be able to perform a measurement acquisition at an field location, transmit information to a database, and process achieved data online and offline in an unaided way. Therefore, in the article the concept of such a system is presented. The system is dedicated to monitoring rock slides and similar geological activities. There are some techniques involved, which allows to reduce the processing and transmission time of data as well as power consumption and maintenance cost.

1 Introduction

Nowadays, information technology gives us an opportunity to gather a huge set of apparently useless information as cyclically collected measurements of various physical parameters. This information can be processed online as well, what is even most valuable, as offline. Thanks to that, we are able to achieve some regularities and correlations between values in time and next create predictions of a real behavior of the observed object. This could be useful in observations of geological activity of soil and rocks i.e. slope movement, rock avalanches and other dangerous phenomena occurrences on inclined regions. One of the most precipitous and perilous geological phenomena are rock slides. An occurrence of this phenomenon depends on many factors: rainfalls, pressures of water inside a

[1] Geology and Information Technology – seminar project organized by The Polish-Norwegian Research Fund, Silesian University of technology and Sogn og Fjordane University College.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2009, CCIS 39, pp. 292–301, 2009.

rock and soil, earthquakes and micro-tremors and many other. These phenomena should be and thanks to nowadays IT techniques can be monitored.

In this article authors are going to present a concept of a distributed measurement computer system based on the sensor networks idea, designed for collecting various data from a given local area, its local preprocessing, local controls and sending gathered data to the central database system. Gathering information and processing it is executed in a cyclic manner with a parametrisable period. The main part of such a system are local measuring nodes – sensors (LMN) and a local process and control node – field computer (LPCN). Additionally, the central database server (CDB) has to exist but it can be located in a distant place. Communication with this server is established via GPRS channel because of low cost of devices, installation and maintenance [4,5]. It is assumed the measurements are performed by proper sensors and signal from it comes to system nodes as analogue voltage or current signal. It means that the system nodes are equipped with standard inputs with AD converters. Thanks to that, nodes are independent from type of measurement and sensor and could act as sensors for various purposes. The nodes are powered by batteries because of lack of power supply sources in the system's working area.

Any measurements should be executed in time function in the way that one can detect changes of factors. Generally, observation of geological phenomena needs ages, years, months. However, in case of rapid and fierce phenomena this is necessary to perform monitoring and data processing in real time with time limitations of days, hours, minutes, seconds. Such an approach gives a possibility of informing about upcoming outpass of assumed tier values for each measurement or about similarity of set of measurements to alarming pattern. As a result, it gives an opportunity to produce a warning or an alert of possibility of an avalanche occurrence.

2 Local System

In the single local field area there are typically several measurement points spread along the given inclined region, i.e. a slope. The size of measured area is about several hundred square meters. The locations of measurement points are fixed and/or variable. The main task is to measure and analyze the physical parameters as pressure, humidity, rainfall, movement etc. There are four functional tasks. The first one is to measure something in the given point. The second one is to transmit data to LPCN by each LMN and collect it in LPCN memory. The third one is to process the local data in run-time mode and, depending on achieved results, to control the local actuators. The fourth one is to transmit gathered data to the CDB located in the offline analysis centre. So locally, there are measurement, transmission, and control functionalities. The transmission issues are similar to the ones discussed in previous chapter. However, in this case the construction of the system node could be quite different.

3 LMN – Local Measurement Node

Local Measurement Node is the device that can perform measurements of several environmental parameters, store the results and send them to the central station. The method chosen for sending the data is radio transmission using ISM frequency. There are some protocols available for such a transmission. One of such protocols is ZigBee [3]. This protocol is becoming a very popular solution for short distance radio transmission and is widely used in sensor networks. Other possibility is to use modules offering raw data channel or channel with some low level modulation. Using such modules can be a good choice where nodes do not have to play role of routers for the network. In the proposed system nodes communicate directly with the central station so there is no need for implementing routing protocol.

The module chosen as the radio transmitter in the node is based on the CC1000 chip made by Texas Instruments (Chipcon). It can work with 433 or 868 MHz frequency depending on the version and can transmit data with the speed up to 78.6 kbit/s. It offers raw or encoded channel working in NRZ or Manchester mode. The radio module has the connection for external antenna and two interfaces to connect the microcontroller. One of them is the three wire synchronous serial interface used for controlling the mode and parameters of operation of radio module. The second serial interface is used for data transmission. It can work in synchronous or asynchronous mode depending on the CC1000 current mode of operation. As the main part of the measurement node the ATmega32 microcontroller has been chosen. Because the radio module works with 3.3 V of power supply, the low voltage version of microcontroller has been used in the system. The node can be supplied with several power sources. One possibility is the accumulator charged by the solar panel. Unfortunately, during long-lasting poor weather conditions solar energy wouldn't be sufficient so charging must be performed periodically by maintenance personnel. Such solution is uncomfortable and generates additional maintenance costs. Another possibility is to use high capacity batteries. An example of such battery is LS33600. It is the 3.3 V lithium battery of capacity 16500 mAh. Such battery has been used in the measurement node.

Because of supplying the node with the battery the special care has been taken on the power management. Both radio module and microcontroller offer many advanced power down and sleep modes. Radio module is switched into power down mode when not used. In this mode it consumes less than 1 μA of the current. Microcontroller is switched into power save mode and automatically waken up every second to perform measurements and store the results. After collecting 10–50 of data samples it switches the radio module on for sending data packet to central station. The number of data packets needed for performing the transmission can be set during node configuration.

The node has 4 16-bit analog and 8 digital inputs. Therefore, maximally there are $M = 4$ analog measures and 8-bit discrete vector, so maximally $m = 9$ bytes of useful data. Any unused input is configured as disabled for saving the battery power. Analog inputs can measure the voltage from any electronic sensor.

Digital inputs provide information about the state of switches or two-state sensors. The node can communicate with advanced sensors using serial interfaces. Both synchronous and asynchronous interfaces can be used. Asynchronous interface is compatible with RS485, synchronous interfaces are compatible with TWI and SPI protocols. Because many nodes send data to one central station, some protocol for accessing the medium must have been developed. The protocol is very simple. All transmission must be acknowledged. If the node does not receive the acknowledgement frame, it resends the data after 1 s. If such a situation occurs three times, it is the information about possible collision in the radio channel. While detecting the collision, the node randomly changes the time of next transmission. After five such changes the node turns into low frequency operation which means that it will perform measurements and transmissions every 30 s. If the node receives the acknowledgement frame, it returns to normal operation.

4 LPCN – Local Processing and Control Node

As a main unit of LPCN, the unaided field computer of PC or SBC (Single Board Computer) class can be used as well as a dedicated computer device can be designed. This computer works without any rotated parts and is closed in a special case resistant to weather conditions. The acquisition module and supporting coprocessor are joined as a PCI board or are a part of the device. The details are presented in Sect 4.2.

Two of the tasks presented previously could be realized in ways which guarantee better system performance related to computing power and power consumption. The first one is local processing. There is a necessity that some data sets have to be processed locally and this processing have to be performed in the fast and real-time way. The main goal for this is to produce appropriate control signals dedicated to local actuators, i.e. traffic lights, buzzers, automatic bars, etc. Taking into consideration the size of monitored areas, it is requested to install a relatively big number of LMNs. Let's assume k nodes for further considerations. So registration and processing a big amount of measures is necessary. Additionally, prediction algorithms used for local control are quite complex and they have time constraints, so there is a necessity to use powerful processing devices with low power energizing. The second task is a long distance transmission to CDB. Since data can be collected in a LPCN memory and local processing exists, the transmission between LPCN and CDB could be executed within a dynamic cycle, according to achieved results from local processing and a data buffer capacity. However, the transmission is considerably power-consuming. So there is a justification for finding out a method of reducing the power consumption related to radio activity time. The good answer for both problems above is FPGA technology [1,2]. Authors consider the use of this kind of circuit in order to use it in LPCN. There is also extra utility; it allows to update a hardware application many times according to changes in the coprocessor project.

The hardware solutions can be realized in various technologies executed in integrated circuits of the following types: ASIC, DSP, specialized processors, and also a programmable array. The hardware solutions as a rule allow to make calculations of one or even two orders of magnitude more quickly than the software solutions. The achieved speed of making calculations is evidently dependent on the method of the implementation and the applied technology.

Additionally, the application of a programmable array to the implementation of computational algorithms, in the relation to different technologies, enables us to change the hardware configuration of the programmable array. Consequently, one can exchange the working hardware application for the one realizing a different function or acting more quickly. The increase of the calculation speed can be held by making the calculations in parallel or executing the operation in the pipeline. Moreover, achieved results in the case of the pipeline solutions are comparable with the quickest solutions executed in the different hardware technology. However, utilization of the VHDL allows to apply the implementation of the hardware application in various programmable arrays. This mechanism is described in Sect. 4.3.

4.1 Remote Communication

An LPCN has to be a single, management and repository device connected locally with LMNs and remotely with a distant CDB. It has to be installed near measured regions, so local environment may be various but mostly not urban. The communication infrastructure in a given location is generally unknown. Thus, from the universality point of view, an LPCN has to service at least a few methods of distant communication. It can be a local area network via either cable or wireless connection, radioline based on radio modems, GPS/GPRS transmission, commutative or direct telecommunication line, etc.

The hardware and software platform described in Sect. 4.2 allows to use almost any communication solution dedicated to distant transmission. However, there is a point to creating a special supporting mechanism which is able to reduce time of transmission. The first reason is a time characteristic of the whole system running. The system has to work with time constraints, because depending on the results achieved from calculations the safety actions are locally performed, and remotely user is informed. The second one is related to power supply. If energizing source is not permanently available, power consumption is important because of battery lifetime. The radio, especially the transmitter, consumes more power when transmission is occurring then when it is in sleep mode. The last one is related to maintenance cost. In some solutions the cost of transmission depends on transmission time or data size (i.e. GSM/GPRS), especially when packet size of useful data is relatively large. All of this leads to the reduction of transmission time. If transmission speed is fixed, there is one simple way to diminish it. Useful data can be compressed and selected protocols should be simple, without control data overload. The FPGA technology mentioned previously gives an opportunity to use it also in order to compress data very fast and without significant increasing of power consumption. This mechanism is described in Sect. 4.4.

Finally, the distant transmission can be integrated with Internet, especially when CDB is connected to it. If the TCP/IP stack can also operate with an LPCN, the proper protocol set has to be used, mainly in transport layer. There are two ways. The UDP protocol uses less control data but transmission is unreliable and user has to control transmission from application layer. The TCP protocol delivers a reliable channel but adds more control data. In order to select appropriate solution, the analysis of channel has to be performed for each particular case.

4.2 Main Unit

In a prototype version, the main unit is a computer host station consisting of elements presented in Fig. 1 and based on any hardware supported by OS. To evaluate this version, the Virtex-5 FXT FPGA ML507 Evaluation Platform [11] with PCIE interface is used. The OS platform can be Linux. Linux seems to be the best choice. Linux is an open source, operating system, developed originally for home PCs. Today, Linux runs on a variety of CPUs including PowerPC, MIPS, Alpha, Sparc, M68000 and ARM. The most important criterion of an embedded system, such as in LPCN case, is timing constraints reliability.

From version 2.6, Linux kernel task can be preempted [8,9]. In the general-purpose operating systems, the process scheduler during execution of a system call is forbidden. This possibly leads to the situation when one task may block the processor until the system call returns, no matter how long that can take. From version 2.6 the Linux kernel, during the execution of system calls, periodically tests delays between the preemption point, and then the kernel task can be interrupted.

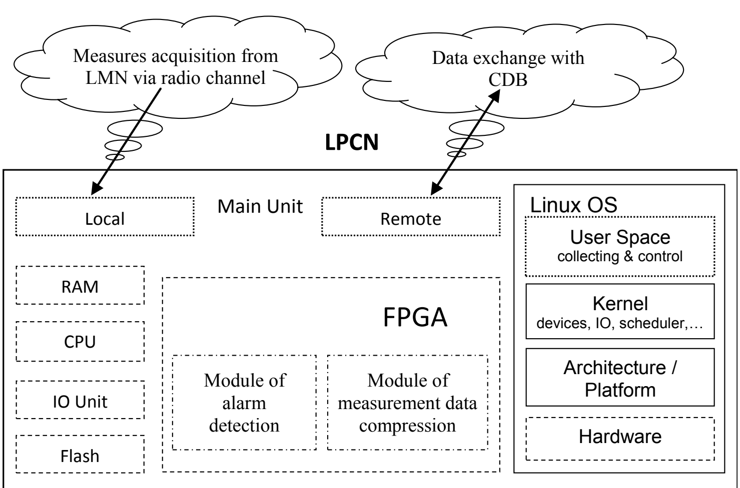

Fig. 1. The components of LPCN main unit

Embedded applications are customized to solve the problem in an original way. This requires customizing Linux to support the specific hardware design. It can be done easier when the system has the modular design with well defined components and API's which can be replaced or customized with minimal impact on other components. Figure 1 also shows the modularity architecture of the Linux 2.6 kernel design. Porting the system from different hardware is possible and convenient during moving the application from development to production platform.

One unquestionable advantage of the Linux is the scalability from the system that provide exceptionally large resources and high-throughput multiprocessor to the microcontrollers. Version 2.6 bringing microcontroller support into the Linux mainstream. It is done mostly by the adoption of the ţClinux distribution. The original ţClinux was a derivative of Linux 2.0 kernel intended for microcontrollers without Memory Management Units (MMUs). However, the Linux/Microcontroller Project has grown both in brand recognition and coverage of processor architectures. Today's ţClinux as an operating system includes Linux kernel releases for 2.0, 2.4 and 2.6 as well as a collection of user applications, libraries and tool chains. Thanks to Linux cross-platform mobility, after preparing the LPCN prototype and running it with the rest of the system, the LPCN as the SBC (i.e. TS7300 [10]) can be designed together with software and hardware projects prepared earlier.

4.3 Processing Support

The modules of alarm detection (Fig. 2) will begin the work after detection of signal informing about the end of collecting a the single package of the measuring data. This information is passed on to the suitable processing modules, which executes the operations on the current given measuring. The modules informing about the crossing of the threshold value for the given measurement will be the first processing modules. The threshold value should be so well-chosen to be able to affirm the possibilities of the pronouncement of the threat. On the example, one can define the threshold value of the pressure of water in soil, from level on which sliding of the ground can occur.

The second processing module informs about crossing the threshold value of differences between K of next measurements for the given sensor. In effect, one can notice violent growth of the signal for the given measuring sensor and inform that the considerable change of the measurement happened in the given place in the short period of time. The last processing module informs about crossing the threshold value of differences between L of measurements from the given class of measurements. In effect, one can get information about the difference of the value of measuring signals from all measuring sensors from the given class. This information will be essential for generating the spatial presentment of the value of measuring signals in the given time. This will let define in what area the threat may happen. I.e. in which place and what size a slope may slide.

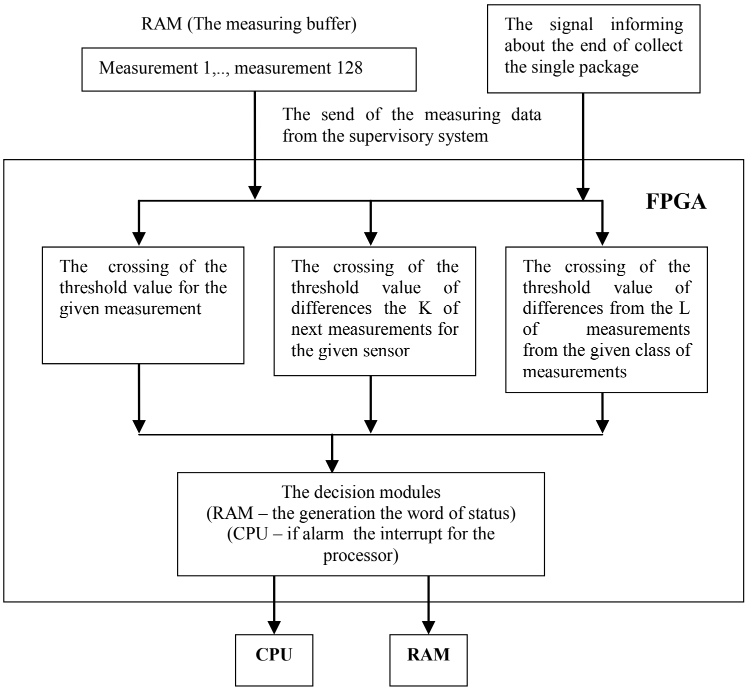

Fig. 2. The modules of alarm detection

As a result of execution of the modules is generation a status word which informs about alarm pronunciation for a given measurement sensor as well as, in case of triggering alarm, generation an interrupt for processor.

4.4 Data Compression

There are several streams of data to send over the net in the measuring system. The data is acquired by sensors (LMN) and transmitted without compression to a host station (LPCN). In the host station the data is buffered, processed and transmitted to the main station (CDB). Before the transmission the data should be compressed using lossless compression method. The gain of that compression is reducing the amount of the data sent, without loss of any information. It will reduce the time of the radio activity, the consumed power, and save its batteries.

Let us assume for example, that there are $k = 32$ sensors, each of them acquires $M = 4$ measured values per second using 16-bit resolution (so $m = 64$ bytes). This gives stream of data $k \times m = 32 \times 4 \times 16 = 2$ kbits/s. This stream is buffered in a host station and should be sent to the main station CDB, using one transmission process per hour. It gives 7 Mbits of data to be compressed for one compression process, so the off-line compression algorithm may be used with the best possible compression factor.

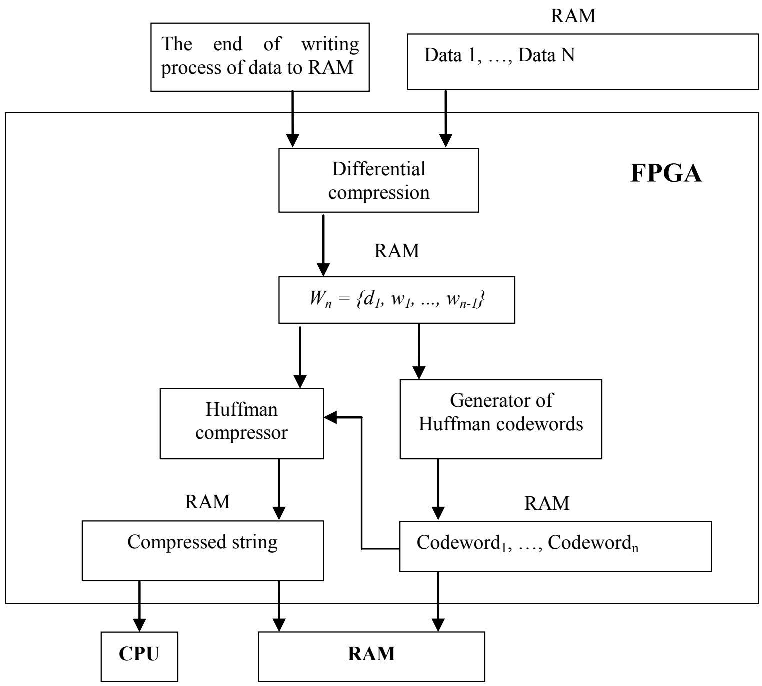

Fig. 3. The scheme of compression in FPGA structure

The compression algorithm must be implemented in FPGA, so it should be simple and not use a lot of memory. The compression process begins when the compression module receives the signal of completing the data in RAM. The first step of the compression process is to separate streams of data containing particular measured values. Those values are physical parameters, for example temperature, pressure, precipitation, measured periodically, so the consecutive values of each parameter are self-similar.

Motivated by self-similarity property, the next step is using differential coding method for decorrelation of the data for every stream. For the given data string $D_n = d_1, \ldots, d_n$, the difference between previous and current value is calculated, for each value. The result is string $W_n = d_1, w_1, \ldots, w_{n-1}$, where $w_i = d_{i+1} - d_i$.

The next step of the compression process is statistical coding of string W_n. For simplicity it can be the Huffman algorithm in off-line version [6,7]. The generator of Huffman codewords makes and writes to the RAM the book of codewords. Huffman compressor takes symbols from W_n string and codes them using codewords from RAM. The compressed string is written to RAM and waits for sending. The scheme of compression in FPGA structure is presented in Fig. 3.

In the case of emergency situation, the data should be sent immediately without buffering and, furthermore, the data may not be in strong correlation. The solution for the compression in that case is to use some kind of the fast on-line compression algorithm or send the emergency data without any compression.

5 Conclusions

The presented idea of a distributed computer system is only a concept. However, authors have wanted to prove that currently available IT techniques allow to design and build a comprehensive system, applicable to the area mentioned above. This article can be a good study for designers. Additionally, in order to put the idea into effect, more detailed projects and research have to be done. All of the authors have experience with issues regarding parts of the described system.

The presented solution is quite cheap, based on common components and some interesting improvements are involved. Using the FPGA technology gives a possibility to build a universal, multiplatform coprocessor and achieve high speed computing with low power consumption. Specialized compression can be used also between measurement nodes and the host station. However, nodes should be able to support it. The quite important advantage is that the system can be integrated with existing devices coming from various producers, both on a measuring and on a computing level.

References

1. Ziebinski, A., Sroka, W.: Realizacja funkcji statystycznych w sterowniku przemyslowym z wykorzystaniem ukladu FPGA. In: Mazur, Z., Huzar, Z. (eds.) Modele i zastosowania systemow czasu rzeczywistego, WKL (2008)
2. Ziebinski, A., Znamirowski, L., Sroka, W.: Implementacja wybranych funkcji sterownika przemyslowego w ukladzie programowalnym. In: Systemy Czasu Rzeczywistego, Metody i zastosowania. WKL (2007)
3. ZigBee Proffessional Guide. A Webcom Publication (2008)
4. Gaj, P.: Zdalne monitorowanie obiektow z wykorzystaniem sieci komorkowych. In: Napedy i Sterownie. No 2(94), Poland, Drukart, pp. 50–52 (2007) ISNN 1507-7764
5. Gaj, P., Holodok, A., Bigewski, Z., Kwiecien, A.: Zdalny monitoring z wykorzystaniem sieci GSM. In: Techniczne i spoleczne problemy z zastosowania Internetu, Poland, WKL (2005)
6. Huffman, D.: A method for the construction of minimum redundancy codes. Proceedings of the Institute of Radio Engineers 40, 1098–1101 (1952)
7. Hankamer, M.: A modified Huffman procedure with reduced memory requirement. IEEE Transactions on Communication 27, 930–932 (1979)
8. Aseem, R.: Linux Kernel 2.6: The Future of Embedded Computing. Part I (2004), http://www.linuxjournal.com/article/7477
9. Linux 2.6 for Embedded Systems, closing in on real time, http://www.lynuxworks.com
10. TS7300 manual rev. 1.5, Technologic Systems. USA (2008)
11. Virtex-5 FXT FPGA ML507 Evaluation Platform, http://www.xilinx.com