ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

The extraction of physical quantities of the processor using the LabVIEW software package

Stefan Koprda, Martin Magdin
Department of Informatics, Faculty of Natural Sciences, Constantine the Philosopher University in Nitra, Nitra

Abstract: The article presents the issues of modelling and simulation in the graphic environment LabVIEW from the firm National Instrument. The possibility to simulate real processes offers many advantages to designers and advance designers from various spheres, such as time saving and costs minimization. The paper deal to create a block diagram in the environment LabVIEW which will show the entry about the temperature of the processor and it will be possible to use it, we had to find an appropriate way which will allow it and will be usable for a great number of computer equipments.

Keywords: processor; temperature; modelling; LabVIEW; programming

I. INTRODUCTION

Nowadays society lives in a modern world of information. People cannot imagine their lives without personal computers and information technology. Digitizing is getting more deeply into most of people´s everyday actions. New technology brings a big number of advantages and actualization simplifications of many processes and technological operations. Phrases such as computer modelling and computer simulation are used very often now. They may occur in manufacturing companies, scientific groups. They occur in places where before the real actualization of some processes it is needed to model the system first, by the use of computer simulation test the system´s behaviour carefully and evaluate its results. Just then the model can be applied to real system. There are various software for process simulation, among them there are programming languages e.g. LabVIEW and MATLAB.

II. MODELLING AND SIMULATION

Under the term "model" we generally understand a system, which is a certain simplification of the original of the modelled system. Between the original and its model there exists a homomorphous relationship of the display, while we differentiate between abstract models, which we can logically ponder over, and simulation models, on which we can perform simulation experiments [1], [2].

Modelling and simulation are becoming an option for the general public in every field. It is a discipline with its own structure of knowledge, theory and research methodology. The core of it is the idea that models approximate the real world. First, the approximation model is created followed by simulation which allows repeated observation of the model. [3].

Modelling is a multidisciplinary activity, since the knowledge of mathematics and physics, theory of systems, probability theory, informatics, cybernetics or cognitive sciences, operation research, and others, can take a share in. Modelling serves not only to solve practical problems, but it is also designed for the realization of certain research and experiments [4], [5], or to simulate phenomena and processes.

Verification of the model occurs in the moment when the model creator designedly tests seemingly correct version of the model in order to find and correct the mistakes, which could have originated during the modelling phase.

Validization occurs, when the model creator and professionals assess to what degree is the created model satisfactory and suits the original [6], [7].

The process of creating and testing hypotheses about models and verification of the suggestion has its bases in experimental sciences. Similarly, experts of informatics use modelling to analyse complicated problems of the real world so that it would be possible to predict what may happen in some sequence of events or actions. The author claims that if the time changes continuously and fluently the model is continuous. If the time changes in progressive steps the model is analogous to a film. The sequences of images are moving so fast that the spectator perceives the progress. Such as tactful sequence of film images represents a continuous movement of actors, computer models of continuous phenomena are developed in a same way.

III. STEPS OF PROCESS MODELLING

The modelling process is cyclical and is closely related to scientific methods and life cycle of development of great software projects. This process is cyclical because in arbitrary steps the observer may return to a former phase, make revisions and continue from this point in the process.

The first step the problem is started to be analyzed. It is needed to examine the situation adequately in order to identify the problem exactly and to understand the basic questions of the problem. The aims are determined and the problem is classified (whether it is deterministically or stochastic). Only after a precise identification of the problem it can be transferred into mathematic symbols and then move to resolution of the model.

The second step is model definition. In this phase a model is proposed which is a sort of abstraction of an already modelled system. Some of the tasks of this proposition step are: data capture, creation of simplifying assumptions and their documentation, assessment of variables and used units, identification of the relations between the variables and each submodels, assessment of equations and functions.

The third step is the resolution of the model. This step represents the actual implementation of the model. It is necessary not to skip the previous steps and not to start with the implementation without understanding the problem minutely and proposing the actual model. If the model is too complicated, it is necessary to return to the second step and add simplifying assumptions or to the first step to rephrase the problem.

The fourth step is about the verification and the interpretation of the problem solution. If there is the finished solution, the results should be examined carefully in order to be sure that the solution resolves the original problem and is also usable. The process of verification determines whether the solution works correctly; meanwhile, the process of validation affirms whether the system meets the requirements. Testing whether the assumptions correspond with real data is important for the verification. If the model shows some imperfections, it is advisable to return to the first or the second step to find out whether a model improvement is required.

The fifth step is the report on the model. The scientific report may be written for the colleagues in the laboratory or will be presented at a scientific conference. That is why this report includes the components which copy the steps of the modelling process itself: the problem analysis, model proposition, model solution, results and conclusion. The sixth step is recommendation [8].

IV. THE USE OF LABVIEW IN THE WORLD

By the use of developmental environment and the programming language LabVIEW scientists can easily combine mathematic equations and algorithms with the measurements of real signals in real time in use of the latest hardware acceleration technologies (multi-core, FPGA, DSP) and COTS – solutions on the bases of the current market [9].

One of the biggest steel productions in the USA, Nucor Corp, thanks to programming in the environment LabVIEW and by the use of PAC in comparison to programming with Programmable Logic Controller (PLC) and diagrams Ladder, reached a decuple of efficiency increase and a drastic cost reduction for factory automation [10].

One of the most interesting reports is that NI LabVIEW with CompactRIO drive the biggest hybrid locomotive in the world powered by a fuel cell. CompactRIO, on which LabVIEW Real-Time and LabVIEW FPGA work, is controlled by operation of the fuel cell. The user monitors the control system by the touch panel installed in the cab of the locomotive. Control application is made of a VIs with modular control algorithms which are in communication with other elements, and of a system I/O with the gate array FPGA which uses the construction of internal knots. So then it is possible to find every I/O according to the name assigned by the application LabVIEW. Each knot has its assigned features including the limits for caution, gauges (conversion from tension on a particular unit) and events, e.g. writing data to disk. On the bases of PAC, the mentality of Programmable Logic Controller (PLC) was implemented into the system [11].

Perhaps it was not suspected that the European Organization for Nuclear Research CERN uses the software IN LabVIEW and hardware NI PXI to control the biggest accelerator in the world. In the particle accelerator the volumes of particles or nuclei collide or it is able to make these volumes collide with particles in target. These collisions cause that a great amount of energy is released. This amount is big enough for creation of an environment of high energy which existed in the early times of the universe. The new accelerator provides the protons a level higher energy than it yet was possible [12].

V. TEMPERATURE MEASUREMENT OF THE PROCESSOR AND ITS INSCRIPTION INTO THE FILE

If we want to create a block diagram in the environment LabVIEW which will show the entry about the temperature of the processor and it will be possible to use it, we have to find an appropriate way which will allow it and will be usable for a great number of computer equipments. There are many ways which can be used. But we have to put in consideration that not all of them can be fully functional on all types of computers. As an example can be the representation of the processor temperature in BIOS. It is an alternative which may be used but here is a problem that not all computers have its up-to-date temperature representation, especially notebooks. That is why it should be better to use auxiliary computer program (utility) which will represent the temperature securely and it can be used while creating a block diagram in LabVIEW. There are many programs to choose from.

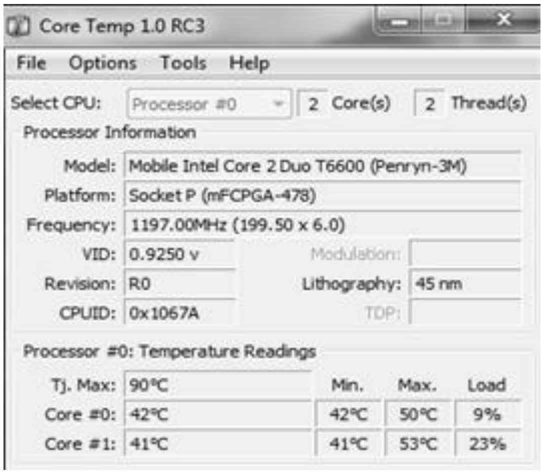

Fig. 1 Environment of Core Temp utility

After careful consideration we decided for program called Core Temp which is able to open up information about temperature in the majority of the processors AMD and Intel.

In LabVIEW for better work with data through Core Temp, besides the program installation, other components are needed to be installed. One of them is to the program appertained DLL library which provides a mechanism of code and data sharing and, in this case, it provides temperature acquisition for our work in LabVIEW.

From the web page <http://www.alcpu.com/CoreTemp/> where is cited that the software can be downloaded for free and which we use for non-commercial purposes, we have downloaded the program Core Temp and also the file DLL called GetCoreTempInfoNET. This DLL library was programmed in language C# and its functionality is provided by using the platform Microsoft .NET Framework version 2.0 and higher. This means that this program will run at the presence of these components including the already running utility Core Temp.

After providing the components we got to the phase of creating the block diagram in which our program will be created. The file GetCoreTempInfo saved in the memory was used at the beginning. In the newly made empty project, we have put the component Constructor Node into block diagram. By the use of this component we have put the constructor CoreTempInfo () from the CoreTempInfo object. This constructor represents a kind of class which containing the code for various methods and functions which provide data and information acquisition that came from the program Core Temp. To its outputs we connected the component Invoke Node which in this case represents a call for a method to acquire data. From the library´s methods GetData has been chosen. The last step was to put another component from this group namely Property Node which has been connected to the outputs of the previous component. Here the GetTemp feature has been chosen. It provided return of the particular value, in this case, the processor temperature in degree Celsius (° C).

In Figure 2 is the final integration of the components described above. The component at the end, into which the links of CoreTempInfo components conduct, is called Close Reference and it serves as a link closure for a subsistent object and thus deletes entry from the memory.

Fig. 2 Integration of components to acquire temperature

The used DLL library provided that the data can be acquired from the Core Temp program and we can work with them in our block diagram. Since the program presents the temperature of the processor cores in certain intervals of time, it was necessary to put two fields Array (named Core 1 and Core 2). For each core they will display the values separately. They were connected to constructor GetTemp. At the same time, a numeric display was placed in the front panel of these fields. Different types were chosen: Thermometer for one and Meter for the other.

Since the constant given in the gate/field starts from the value 0, the first core will represent this value and for the second core the value 1 must be set. In the block diagram, it was necessary to put all the components and the objects into White Loop in order to have a working loading. In this loop, there is a Loop Condition (part of the White Loop) to which a control component Stop was created. This loop allows the program to run until it is stopped. Three graphs have been chosen to represent the processor temperature: on the block diagram there are two processor cores, each will have a graph. There is a third graph which will represent the measured values at the same time. Waveform chart has been chosen which will suffice for the representation of these values. For representation on values in the graph it is necessary to connect another component between the graph and the gate/field Array. This component, Index Array, will set the loading of temperature of each core by the use of connectable constant (0, 1). For the graph, which will represent the measured values of the cores simultaneously, a component had to be placed between the graph and the fields which will connect together the two fields of the cores. The component is called Build Array and, on its inputs, the links from the fields representing the core temperature were brought. For the function of temperature loading time interval, the component Time Delay was placed in the block diagram and it was set to three seconds.

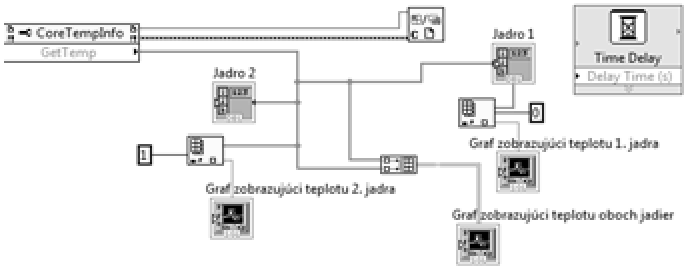

Fig. 3 Integration of components for temperature representation into fields and graphs

VI. DISCUSSION

The created block diagram which by the use of its front panel registers and represents the measured values of processor core temperature provides the user with a well-arranged and, by the graphical representations, aesthetically approachable view on the measured entry. According to Figure 4 (upper figure), placed graphs allow to monitor the progress of temperature values of each core in a time horizon of 100 seconds at three-second interval. It also allows examining and evaluating temperature changes in comparison of previous entries in the same time horizon.

Besides the described graphs, the front panel provides another view on measured entries. As mentioned above, other outputs of this program were created for entries from the first processor core.

The entry Average of the last three measurements from Figure 4 (bottom figure) always registers the average of the last three values in every new value loading and allows monitoring the trend of the last measurements in numerical terms with exactly two decimal places. In the case of integer, zeroes are not represented after the decimal.

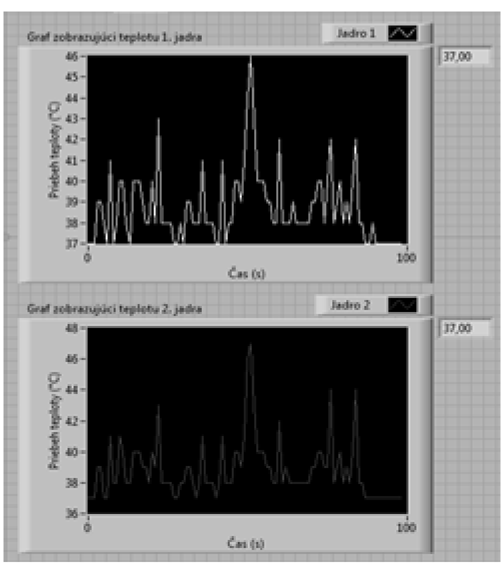

Fig. 4 Numerical temperature representations of cores and the average

Simultaneously, entries of the current temperature are saved on a disc by means of functions for recording in LabVIEW. It is a record into a text file in an ASCII format. A creation of new text file will take place automatically at the beginning of program. The user has to only enter the name of the file and the place on the disc where it should be saved.

Temperature values of the processor core are recorded in a three-second interval, in a same way as they display in numerical graphs and representations. Data with current time of measurement are recorded from the program initiation. This means that it is possible to return to acquired data anytime and find out what was the temperature course in various time intervals.

VII. CONCLUSION

Reached results can be used in wide practice. It is possible to use the program for performance regulation of cooling systems through linking other components and functions in block diagram to the designed program and thus provide a more efficient hardware cooling. Other ways of program use may be e.g. in a laboratory for management and storage of records about processor temperature. Thus we can have at disposal the entries about the temperature course during the monitored period for observation and analysis.

The programming environment LabVIEW has great possibilities in scientific, technical and other groups nowadays. According to our opinion, this tool will be more and more used for various computer simulations and will be an important component in the world of process modelling and simulation in the future.

ACKNOWLEDGMENT

This publication is supported thanks to the Fund for supporting the Centres of Research and Development with internationally comparable quality of operations, Faculty of Natural Sciences, CPU Nitra, Slovakia.

VIII. REFERENCES

[1] P. Peringer, *Modelování a simulace,* Fakulta informačních technologií: Vysoké učení technické v Brně, 2006, pp. 206.

[2] J. Fejfar, J. Šťastný, and M. Cepl, “Time series classification using k-Nearest neighbours, Multilayer Perceptronand Learning Vector Quantization algorithms,” *Acta univ. agric. et silvic. Mendel. Brun.*, 2012, LX, No. 2, pp. 69–72.

[3] J.A. Sokolowski, C. M. Banks, *Modeling and Simulation for Analyzing Global Events*, Wiley: 1 edition (July 14, 2009), pp. 205.

[4] D. Klocoková, “Integration of heuristics elements in the web-based learning environment: Experimental

evaluation and usage analysis",. Paper published in *Procedia - Social and Behavioral Sciences*, . ISSN 1877-0428, 2011, vol. 15, pp. 1010-1014.

[5] D. Klocoková, M. Munk, "Usage analysis in the web-based distance learning environment in a foreign language education: Case study," Paper published in *Procedia - Social and Behavioral Sciences*. ISSN 1877-0428, 2011, vol. 15, pp. 993-997.

[6] R. G. Sargent, "Verification and Validation of Simulation Models," (Presented Conference Paper style) presented at the 2003 Winter Simulation Conference, pp. 37-48.

[7] K. Jensen, "Coloured Petri Nets - Basic Concepts," *Analysis Methods and Practical Use*: Volume 2, 1997, Springer-Verlag, pp. 236.

[8] A. B. Shiflet, G. W. Shiflet, "Introduction to Computational Science: Modeling and Simulation for the Sciences," *Princeton University Press* (May 22, 2006), pp. 576.

[9] National Instruments. (2013, September). National Instruments představuje novou úroveň produktivity s LabVIEW 2011. [Online]. Available:

<http://digital.ni.com/worldwide/bwcontent.nsf/web/all/FDAC>.

[10] D. Branth. (2013, August). Snižování spotřeby energie a zvyšování efektivity procesů pomocí programovatelných kontrolerů. [Online]. Available:

<http://sine.ni.com/cs/app/doc/p/id/cs-13732>.

[11] T. Erickson. (2013, August). NI LabVIEW a CompactRIO řídí největší hybridní lokomotivu na světě poháněnou palivovými články. [Online]. Available:

<http://sine.ni.com/cs/app/doc/p/id/cs-13729>.

[12] Losito, R. (2013, August). CERN využívá software NI LabVIEW a hardware NI PXI k řízení největšího urychlovače částic na světě. [Online]. Available:

<http://sine.ni.com/cs/app/doc/p/id/cs-13683>.