Implementation of Process Control and Monitoring Software with Dynamic Compilation in .NET Technology

Marek Babiuch [1]***, Radim Farana*** [2]***, David Plandor*** [3]

The paper describes robust software for monitoring and process control which was created to support control and monitoring of the experimental models have been developed within the multiunivesity projects in recent years and successfully used in the teaching fields of automation in laboratories of many universities, which we in the inter-university cooperation verified many times. The advantage of this software is usage of dynamical compilation of libraries of new devices without necessary of software upgrade and recompilation.

Keywords: *.NET, control, component, dynamic compilation, .dll library, measurement, software*

Introduction

Monitoring and processes control is a field that for many years has developed with progress in automatic control theory is very closely dependent on the work with experimental models, in which practically verify the theoretical results of scientific work. Department of Control System and Instrumentation has spent many years developing and implementing experimental physical and virtual models of technological processes that can efficiently verify the methods of analysis and synthesis of control systems with an emphasis on conceptually new approaches.

The models are verified, appropriate methods and algorithms of automatic control are derived, the methodology for the development of new physical and virtual models with strong software support is improving, which holds the current trends in information technology, and thus deepening scientific cooperation with other universities in the branches in Czech Republic and cooperating foreign universities in Europe, but also the effective contribution of this scope, cooperation with industry, where experience with these technologies has led to the realization of many projects [1].

Models for Software Application

Embedded software, will introduce the above mentioned specific experimental model of technological processes. It is a laboratory model for teaching control. It can simulate and demonstrate the basic and advanced processes control. Application includes solutions of controllers (two-position controllers, PI, PID, fuzzy control, etc.). This model can be connected to a computer with multifunction card that has analog inputs and outputs. Our new software application is therefore designed to ensure the display of measured data, and generating control signals. The model is used with software environments Control Web 2000 [2], InTouch, Matlab and WinCTRL. Just last-mentioned application is written in Visual Basic 6 and monitoring processes will be replaced by our new application implemented in. NET as a principle of software components communication. The described model allows to realize one-dimensional and multi-dimensional control tasks. The programmer has the choice of output (measured) variables, which can be measured by temperature sensors located at different distances from the heat source - the bulb, or air flow in the tunnel measured by fan flow-meter. Model is visualized on the left side of Fig.1.

Model of heating and ventilating is formed from a light bulb powered from controlled voltage supply (creates heat and light source), which is located in an enclosed tunnel, by which air flows arising from the activities of the fan [3]. Several sensors are located in the tunnel:

- Three temperature sensors - measuring thermistor of bulb temperature, thermistor for temperature measurement in the immediate vicinity of the bulb and the thermistor measuring the temperature at the rear of the tunnel.
- Photodetector - photoresistor measuring the brightness of bulbs.

[1] *Ing. Marek Babiuch, PhD.*, VSB - Technical University of Ostrava, Czech Republic, Department of Control Systems and Instrumentation, marek.babiuch@vsb.cz

[2] *prof. Ing. Radim Farana, CSc.*, VSB - Technical University of Ostrava, Czech Republic, Department of Control Systems and Instrumentation, radim.farana@vsb.cz

[3] *Ing. David Plandor*, VSB - Technical University of Ostrava, Czech Republic, Department of Control Systems and Instrumentation, david.plandor@vsb.cz

(Review and revised version 15. 09. 2011)

Fig. 1: Model of Heating and Ventilating System.

- Thermoanemometer - consists of two thermistors, the first is located in the tunnel and measures the velocity of the air, the second one is referential thermistor, which is not influenced by air flow.
- Volumetric fan flow-meter - measuring fan with connected rpm sensor.

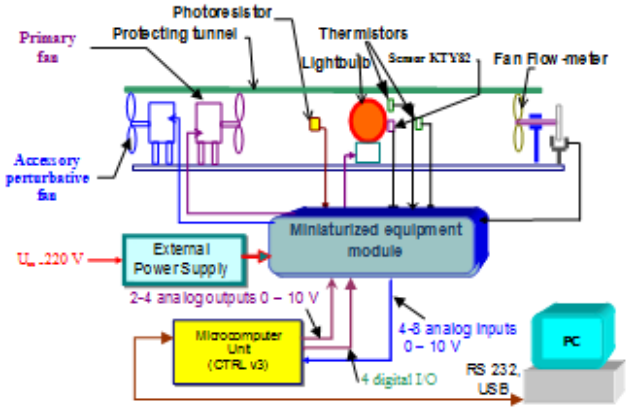


Fig. 2: Block representation of model.

Microcomputer unit CTRL is used for external signal connections from examined physical object connected to PC. We can use up to 16 analog inputs and 4 analog outputs. Using microcontroller unit we can set digital outputs, generate impulses, through the parameters change their width, number and width of the space, set the analog input, establish the status of digital inputs and manage operations with EEPROM.

HVAC system (Heating, Ventilating and Cooling) from National Instruments company is similar established system. The HVAC system consists of some sort of heating system such as boilers, furnaces or electric heat; a cooling system such as air conditioning or chillers; and ventilation components. In HVAC module we can measure and control temperature, relative humidity, pressure, air Flow, speed. NI Company solves software control using by applications implemented at LabView [4].

Analysis of New Software Application

A presentation program WinCoMeT (Controlling and Measuring Tool for Windows) is the base of an application. The previous version WinCTRL is closely linked to the use of the CTRL unit, while WinCoMeT is using virtually any data acquisition card which supports Microsoft Windows.

All communication devices are connected via components. A special program code for communicating with the CTRL unit is created and the program provides its methods to WinCoMeT. A strict naming convention for the communication components is set, this needs to be considered when the additional communication components for other devices are created. This system reliably ensures the possibility of a dynamic development of an application without having to interfere with the main program WinCoMeT. From Fig.4 it is clear that the new model (right) is far more flexible, individual parts can exist independently and they can be connected with other parts on the basis of predetermined rules. In case of a change it is sufficient to replace the modified part only.

So first we created a component of CTRL unit, as it is currently used for communication with the hot-air model, then we focused on the multifunction card from National Instruments used by the Department for the variability and

robustness of the software. These cards allow extra device simulation without the necessity of having the multifunction card physically connected.

All program data are stored in the SQL database. Presentation of the measured data is performed using the selected components. Export of data is implemented in standard formats. Charts can be saved directly into a graphic file, or saved into the clipboard for direct insertion into other applications, or there is an option to directly export data to Microsoft Excel, including drawing a chart. The measurements can be stored in a specific format for the program WinCoMeT with re-opening, viewing and exporting options. All parameters for the generated signals and controllers can be edited in the program, including definitions and calculation formulas. A user with basic programming skills can develop the application himself, as well as add new controllers, modify or improve the existing ones. The newly created scripts and formulas are dynamically compiled and used by the standard calling method.

Internally, the program is designed with objects that interact with each other (Fig. 3). It always consists of forms, classes and the only component of the device. The main form creates instances of objects when the application starts. Firstly, it creates an instance of the class for the device, where the currently selected (or the last saved in parameters) *. dll component of the device is compiled. A class for the device then provides communication between a component and other application objects and provides methods to read, write, initialize and configure the components. Furthermore, an instance of an object for regulators and signals is created, where methods are compiled during initial-

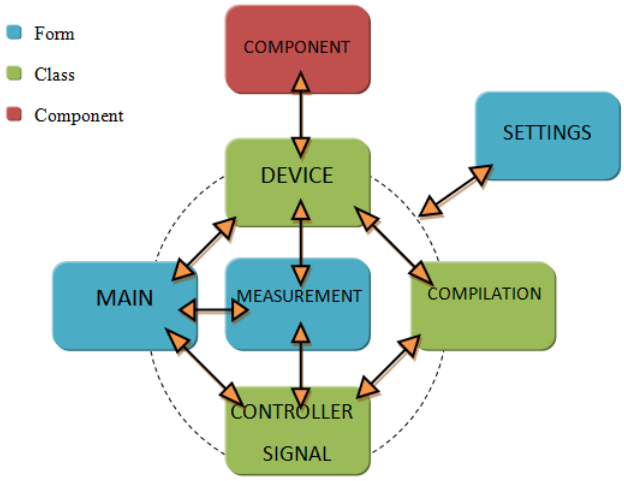


Fig. 3: Component model of implemented .NET software.

ization for all controllers and signals defined in the program. Controllers and signals are defined in detail - they are determined by their parameters, including data types, the limits of their values and a script is defined to calculate their output values. This definition is the reason to necessarily run the program from the text database values and create a class and methods that will be used for measurement and control in the program. A separate class is used to compile the classes of devices, controllers and signals.

If instances are successfully created for devices and controllers, it is possible to proceed to create an instance of a form for measurement and control. There is only one form for both measurement and control, while for measurement only unnecessary fields are hidden and different modes are properly taken into account in the programming code. A form for setting up provides the opportunity to change the parameters of the objects.

Following tools were chosen for the software realization:

- Visual Studio 2008 a .NET Framework 3.5 - Robust platform .NET is the key products for developing applications for Windows-based managed runtime environment with an extensive set of base classes called .NET Framework since 2002.

- Microsoft Chart Control - MS Chart Control for .NET Framework provides huge possibility of data visualisation. It is fully controlled component for WinForms and also ASP.NET applications.

- Microsoft SQL Compact database - Allows encapsulation to the target application. We donâĂŹt need to install the extra database environment after installing WinCoMeT. Database works as an in-process component; it is part of the process of parent application.

Component Creation and Dynamic Compilation

Creation of components

Applications created by .NET technology are composed of components. All objects have properties, methods and events that are the basis for object-oriented programming (OOP). It is very useful when designing and implementing a project to separate the functional part of the program from the presentation part. This means that the program is divided into several parts, which communicate with each other like they were in one program, but they are separate and can be used independently by different client applications [5]. The presentation application calls the service method, sends the input data, processes and displays the output data. This makes it possible to edit each individual application, which works, while maintaining the convention (which must be defined at the beginning of the project, preferably in the modeling phase) with the same typified data format. The presentation part of the application can then communicate in a unified way with different components, which are closely tied to a specific hardware.

A component is a special type of execution file created in the development environment of .NET technology. After compiling the component it is usually connected to an application that will use it by adding a link (reference). These components very often run on a Web server and provide data and services to the web services. The desktop application works similarly, but not on such a scale [6].

The component's creation itself is not complicated more than the creation of a regular class, to which the necessary properties, methods and events are added and linked to client applications. The . NET component is actually a class in a pre-compiled package with the *. dll file (Dynamically-Linked Library). When running it the .NET component is called and loaded into a memory, which is ready for use by client applications. Such components are usually developed and tested as separate projects and do not necessarily have to be a part of a project (may not be called from external client applications). The components can be added to the client application after compiling the library *.dll [7].

All Microsoft Windows platform applications run in a computer's memory and use its resources like disk space, network services and graphic interface. Modern Windows systems are already multi-tasking operating systems; this means that more applications share memory and its other options at once. Windows system manages the operation of applications, dynamically allocates memory to them through the processes. The memory is occupied by an application and its data only for a period of time, the time that is allocated to individual process threads. The threads are sub-process computing tasks. Memory is allocated dynamically according to application needs and priorities. The scheduler of Windows System allocates memory according to application requirements and monitors whether the time of the thread has expired. Then automatically transfers the released memory to other threads.

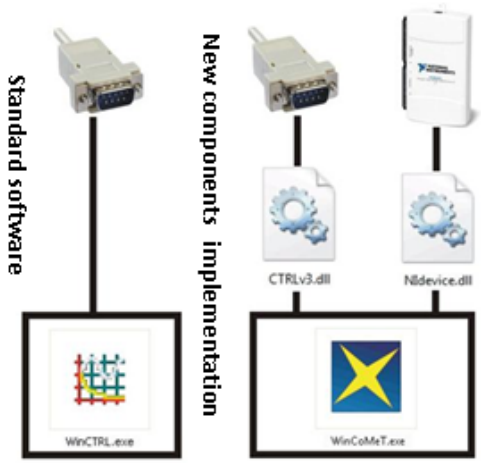


Fig. 4: Comparison of standard and component software.

Many applications such as Microsoft Word or Excel support a wide range of functions. Rather than having the application in the form of one very large file, it was suitable to divide it into sub-functional units which can then be used in both programs. The main program files have the extension *.exe, while separate functional units are in files with extension *.dll. If the main program needs to use the functionality contained in the *. dll library, it will add it to the main application process. Windows System works with the main program and executed code of *.dll library in a single process. A call for *.dll component, which is a part of the other application process is called in-process. There are situations where components are not called as in-process. This happens, for example, in a situation when we want

the program to print a document. An application requests a driver from the Windows and it is loaded into memory. If the same driver is needed by another application, Windows loads the driver as a non-process service, in which case the driver may be used by more than one application at once. Consequently, the Windows System cannot read more than one copy of the same driver into memory.

Type of components

In-process components are .NET components in *.dll format, which work within the host application, share memory and processor requirements for the host application. When start up the component is loaded from a disk and added to the host application process. Since there are no external procedures called for mediation of communication between a component and the host application, the setting and determining property values, the call of methods and reactions to events caused by the component call is fast. Out-of-process components are alternative architecture which includes server components that run separately and independently of the process applications that use these components. Server components generally have the extension *.exe. When Windows reads the out-of-process component, a separate process is created for it that is controlled by the requirements of the component independently of the client's application. Windows provides communication between the server and a client application via messages. Compared with the older development environment it is especially advantageous that:

- In-process components run faster because no remote procedures are called. Client applications have direct access to a component's means without having to use the communication dialogue.
- Out-of-process components can prove useful in some cases better than a in-process component, as if the out-of-process application crashes, it will not also cause a crash of the client application. For in-process applications the crash of a component means usually also a crash of the client application.

Dynamic compilation

The main goal of WinCoMeT application is possibility of usage various hardware devices. At previous chapter we described principle of the components, however, we refered on these components directly at source code. Components for which there was a link (reference) at the compilation time of the applications can be used without any restrictions. We can create instance of the class, call methods etc. These components and dll files are located in application folder with other parts of application. We had to solve the possibility of loading new components, which does not exist at compilation time of the application so we cannot create a link before this compilation. We have the opportunity to use dynamic compilation. This is way of using the class of the System.Reflection namespace which we must use at our source code.

using System.Reflection;

Then we can use the Assembly class that provides methods for loading components into an application and the possibility of full utilization. For example if we have a component in the form of *. dll file and want to use it in source code, we do it using the Assembly class in the following way.

string str = Application.StartupPath + "\\Device \\" + device + ".dll";

Assembly asm = Assembly.LoadFile(str);

We approach to component by asm variable, for example, so that we assign the class which we now into variable, and immediately create an instance.

New devices and controllers

The new device can be added into the program in general settings by clicking the new device dialog to select *. dll file from file explorer. The component is verified after selection and if it suits, it is added to the application and input and output of device are defined, and classes and methods are compiled. Then you can already use the device. Your own controls and signals are also possible to define in the application, as well as edit existing ones. For all parameters can be defined the data type parameter, the default value, type of restraint and self-limitation.

Components or dll libraries provide communication with hardware devices and WinCoMeT applications. These libraries are also created at Visual Studio 2008. Convention, which has to kept, was established for a new equipment creation. Therefore, the rules were established for the terminology of classes and methods which are called from WinCoMeT application. Values of standard variables are loaded after dynamic compilation. Required variables are integer count of inputs and outputs. Then, the class which has the same name as the component is found. It created an instance of this class. Class has a constructor with a compulsory parameter of type dataset, which is used for parameters that are stored in a database in WinCoMeT. The advantage of this solution is immediately ready for use, including component settings that were made during the previous measurement. Then it is possible to call methods that are also standard. They are methods for reading values, for writing values and methods for settings. Device

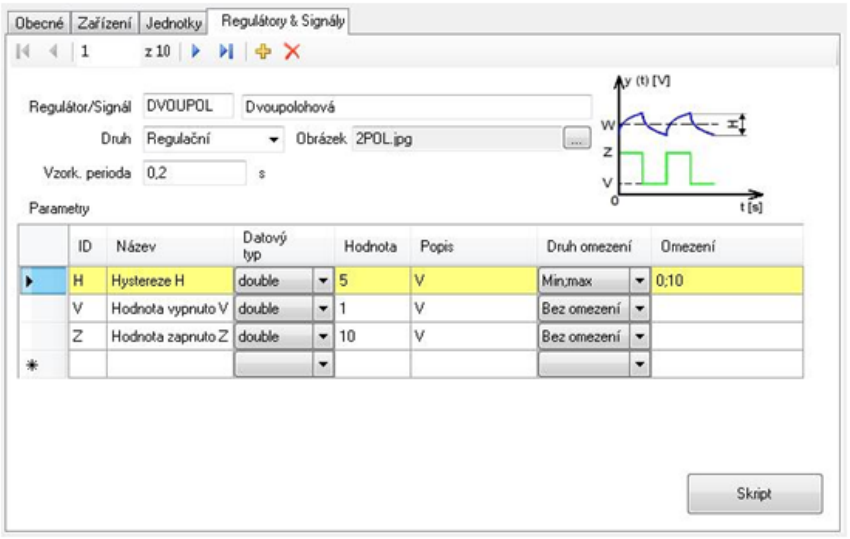


Fig. 5: Definition of new signals and controllers.

parameters are set on the component site. Component therefore has its own form for setting parameters set. These parameters are then sent in the form of a dataset back into WinCoMeT where they are stored in a device database table.

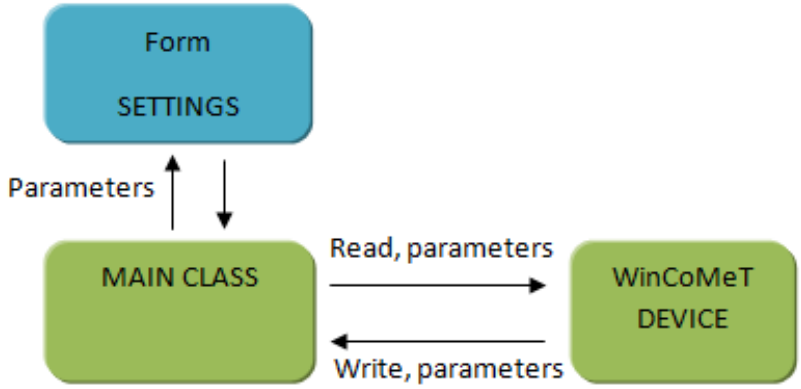


Fig. 6: Communication of application with component.

Stored parameters are sending directly to the component with next creation of instance, therefore it is not necessary to set up the component again during next application execution. At fig. 6 is described the communication system between device class and main component class, which communicates with a custom form for settings and provide transfer of the parameters.

Software Application

The program is designed as an MDI application (Multiple Document Interface), and then there is one major form FrmMain.cs. All other forms are subordinated; they are called from the main form. It can be access to objects of superior form from all child forms and also through other subordinate forms.

Required libraries load during main form appearing, there is a dynamic compilation of the currently selected (or default) device, then the dynamic compilation of methods for calculating the output value of the control signals are generated and the possibility of a new version of application on the Internet is verified.

General program settings

For the program settings are two forms. The first one is for general settings, equipment selection, definition of units, definition of controllers and generated signals. The second form contains a set of inputs and outputs of the currently selected (or default set) devices. In form of general settings are 4 bookmarks:

- General - graph description and measured values display format.
- Device - device selection, parameters of device configuration, settings the limit of control value. Here is possible to limit control value at the top and bottom by characteristics of the hardware device.
- Units - definition of units for assignment to inputs and outputs.
- Controllers / signals - parameters of controllers and signal definition.

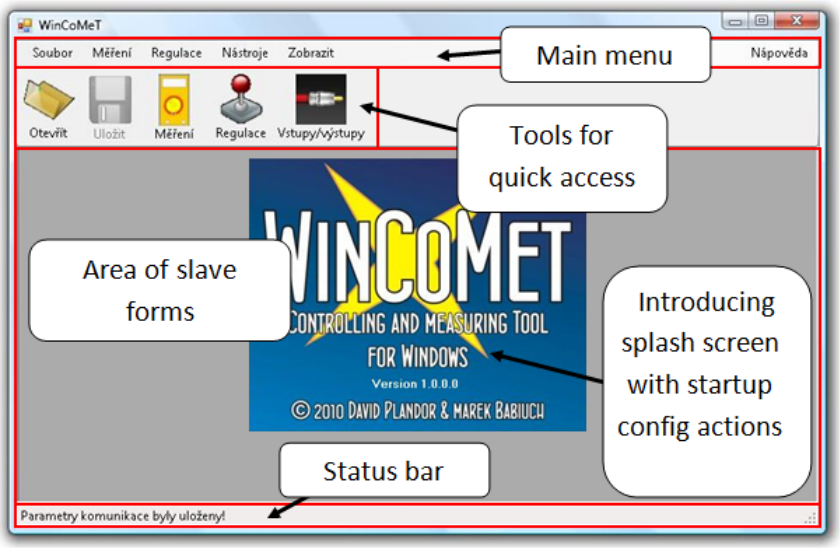


Fig. 7: Face of WinCoMet application.

Inputs and outputs settings

If the device is selected, we can set the parameters of their inputs and outputs. For all inputs and outputs can be set the following parameters:

- Active / Inactive status - it means option, if the input or output will be used. If not marked as active, it does not appear in other forms.
- Constant- numerous constant for measured value.
- Value - expressing the size of the resulting measured or calculated values.
- Sampling period - sampling period in seconds - the minimum and default value is 0,1 s.
- Conversion - we can enter a polynomial (function of x variable) under which the resulting value is recalculated. It is possible with the known characteristics of the sensor to convert the measured value to the final value of another dimension (e.g. from voltage of a thermistor to temperature). If we do not have the characteristics of sensors available from the manufacturer, it can be determined by experimental identification.
- Description - description of input / output.

Measurement and control

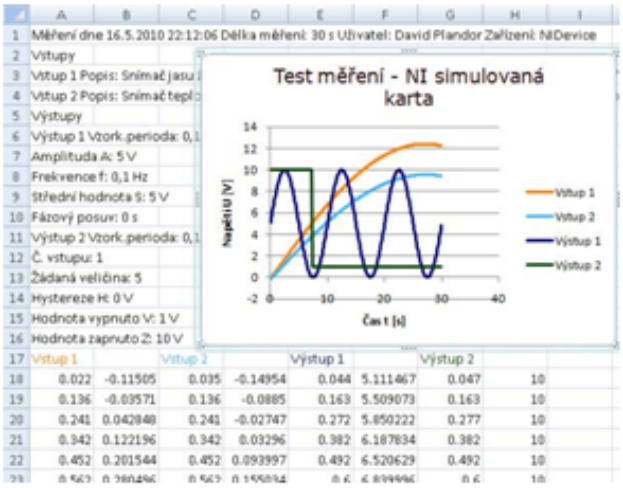
Maximized form for measurement and control loads after click on the icon control start. This is a child form of MDI application, therefore remains main menu, status bar and quick launch for another use.

DataGridView as a standard tool Visual Studio serves for parameters of inputs and outputs and also for parameters of controllers and signals visualisation. All DataGridView objects are connected with database tables and their content is loaded during main form initialization.. All parameters can be changed during the measurement, we can change for example desired value controller gain. Press Start to start the Control process begins with start button clicking. Measurement or control takes optional time in seconds, but it can be stopped by clicking the same button and the description is dynamically changing. Values are filled at the data bookmark after control process. View object is created at SQL database for inputs and outputs, it is one list for all data series (inputs and outputs) and after click on a specific input or output will show two values - the time and measured (functional) value.

Export of measured data and parameters

For further processing the data is exported into Microsoft Excel, including the text description of the measurement, the date and time of measurement, the user who carried out the measurements and all the of control or measurement parameters. MS Excel also draws a graph that takes the export data as the description of the color chart or data series from the program WinCoMeT.

Users can use the export to *. csv format, which contains the same information, except for plotting the graph. Charts can also be exported directly into graphical formats *.bmp, *. jpg, *.png, *.gif, *.emf and *.tif, or the Windows clipboard, where you can paste it into another application (Fig.8).



	A	B	C	D	E	F	G	H	I
1	Měření dne 16.5.2010 22:12:06 Délka měření: 30 s Uživatel: David Plandor Zařízení: NiDevice								
2	Vstupy								
3	Vstup 1 Popis: Snímač jasu								
4	Vstup 2 Popis: Snímač tepl								
5	Výstupy								
6	Výstup 1 Vzork.perioda: 0,1								
7	Amplituda A: 5 V								
8	Frekvence f: 0,1 Hz								
9	Střední hodnota S: 5 V								
10	Fázový posuv: 0 s								
11	Výstup 2 Vzork.perioda: 0,1								
12	Č. vstupu: 1								
13	Žádaná veličina: 5								
14	Hystereze H: 0 V								
15	Hodnota vypnuto V: 1 V								
16	Hodnota zapnuto Z: 10 V								
17	Vstup 1		Vstup 2		Výstup 1		Výstup 2		
18	0.022	-0.11505	0.035	-0.14954	0.044	5.111467	0.047	10	
19	0.136	-0.03571	0.136	-0.0885	0.163	5.509073	0.163	10	
20	0.241	0.042848	0.241	-0.02747	0.272	5.850222	0.277	10	
21	0.342	0.122196	0.342	0.03296	0.382	6.187834	0.382	10	
22	0.452	0.201544	0.452	0.093997	0.492	6.520629	0.492	10	

Fig. 8: Export of measured data.

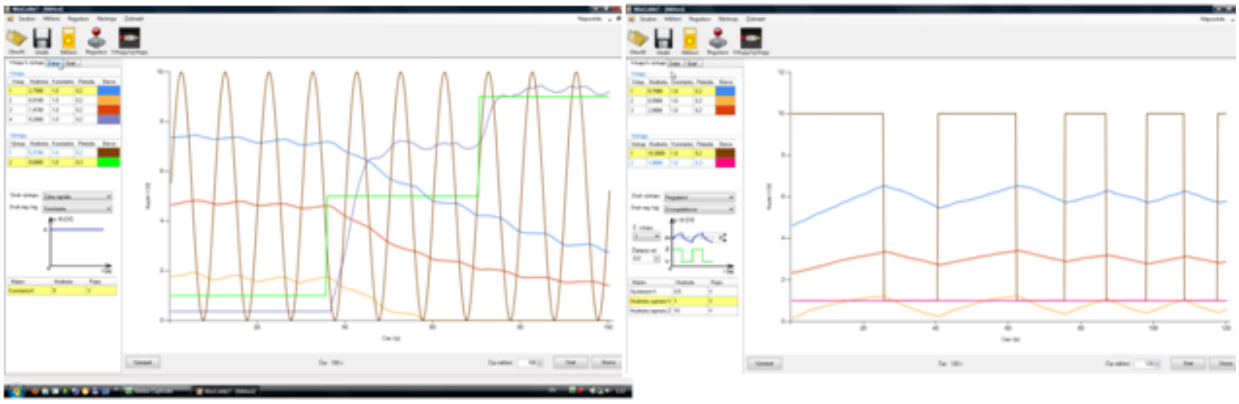


Fig. 9: Examples of real-time control and measurements.

Conclusion

The paper has described the typical area of specific research of the Department of Control Systems and Instrumentation at the Technical University of Ostrava and that is the software implementation for support the monitoring and control of technological processes of various area at mechanical engineering and geology automation. In addition, there is the essence of creation described software using new software tools and technologies used in many of our projects. New software at .NET technology uses dynamic compilation during the program execution, so you can add the new equipment to the application and write new program code for control models.

Acknowledgement

Software has been registered at the Science and Research Information Register of Czech Republic. The presented results have been obtained during the solving of research project SP201030/2010 and following SP201118/2011 supported by the Ministry of Education, Youth and Sport of Czech Republic.

References

[1] L. Landryova and M. Babiuch, *Modeling Objects of Industrial Applications in Handbook of Research on Social Dimensions of Semantic Technologies and Web Services. - Chapter XXXVI pp. 743-759, 1099 pages.* Informatic Science Reference, Hershey, New York, IGI Global, 2009. ISBN 978-1-60566-650-1 hardcover, ISBN 978-1-60566-651-8 ebook.

[2] M. Babiuch and J. Skuta, *Usage of SPI interface in applications with MEMS components.* Acta Montanistica Slovaca. Kosice(FBERG), vol. XIII., 2008. p. 178-182, 2008. ISSN 1335-1788.

[3] L. Smutny and R. Farana, *The Consortial Approach to Advanced Control Laboratory Education in Proceedings of the 4th IASME/WSEAS International Conference on Engineering Education (EE'07).* WSEAS Press, 2007. ISBN 978-960-8457-86-7, ISSN 1790-5117.

[4] National Instruments, *Configuring an HVAC System using National lnstruments, 2011*. Available from www <URL: http://zone.ni.com/ devzone/cda/tut/p/id/3135>.

[5] D. Fojtik and J. Kulhanek, *Concept of User-Specific Ticket Selling Cashbox System*. Proceedings of the 9th ICCC'2008. Sinaia, Romania, May 2008, pp. 171-174, 2008. ISBN 978-973-746-897-0.

[6] M. Babiuch and M. Hnik, *Using Technology of .NET Web Services in the area of Automation.* Transactions of the VSB - Technical University of Ostrava, Mechanical Series, No. 2/2009, volume LV, article No. 1680, p. 1-6, 2009. ISBN 978-80-248-2144-3, ISSN 1804-0993.

[7] M. Groh, *Creating Components in .NET, 2011*. Available from www <URL: http://msdn.microsoft.com/en-us/library/ms973807.aspx>.