

An Integrated Application to Support the Implementation of Measurements and Data Analysis

Robert Czabanowski

Department of Off-Road Machine and Vehicle Engineering, Wrocław University of Science and Technology, Łukasiewicza 7/9, 50-371 Wrocław, Poland

Robert.Czabanowski@pwr.edu.pl

Keywords: Measurement Application, LabView, Virtual Instrument, DAQ

Abstract. The article presents the design of an application created for the integration of various measuring devices in complex measurement systems. Uniform user interface with the necessary functionalities essential for the implementation of measurements simplifies carrying out large measurement tasks. The modular and open structure of the program allows for adapting to current needs. The applied LabView environment enables easy implementation of additional functions related to on-line processing of measured quantities.

Introduction

In the Laboratory of the Department of Off-Road Machine and Vehicle Engineering of Wrocław University of Science and Technology, as in many others, various control and measurement equipment is used. This diversity applies to producers, classes, supported (and most commonly used) communication interfaces as well as the functions that are most often used. The most frequently applied devices are: measuring amplifiers (often used for cooperation with strain gauge sensors), specialized or multifunctional measurement cards (DAQ), PLC controllers, digital multimeters and oscilloscopes, signal analyzer, and others (Fig. 1).

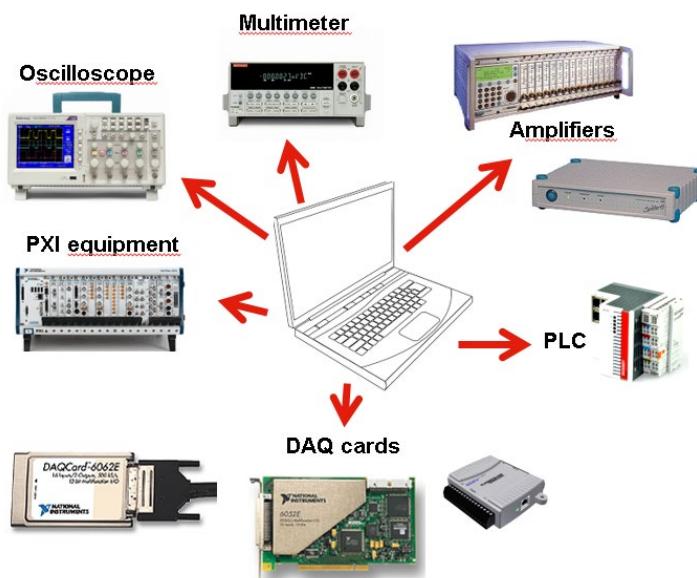


Fig. 1. Laboratory measuring devices with various interfaces (PCI/PXI, USB, FireWire, WiFi, RS232C, GPIB, Centronics, TCP/IP et al.)

Today these devices are mostly constructed so that they can be a part of a computerized measurement system even if it is possible to use the equipment without working with a

computer. To fulfill this task, hardware manufacturers offer dedicated software or at least drivers to enable the creation of custom applications (using e.g.: C, C++, assembler, Python et al.). In the case of building a more complex measurement or control and measurement system, when the system designer is forced to use various devices, there are difficulties in system integration. Applications offered by hardware manufacturers can't deal well with the hardware support of competition. The functionalities offered by these programs are not always sufficient and require the development of special procedures for processing the data recorded in the off-line using other tools (Matlab, Statistica etc.) or also creating macros or extensions (e.g. by generating scripts or subroutines using VBA).

Requirements for a measurement software

For the needs of the implementation of a wide range of measurement tasks, a measurement application should meet the following requirements:

1. Possibility the use of various measuring devices using a uniform user interface.
2. Configuration of the measuring and control devices (measurement interface, synchronization method, type of trigger).
3. Configuration of the individual measurements channels (measurement range, sampling frequency, method of using the measurement signal).
4. Possibility of introducing the owned characteristics of measurement channels or of obtaining them using the teach-in method, in order to obtain the measurement results in the desired units.
5. The ability to perform on-line calculations on the values obtained from several measurement channels so that you can directly get measurements of quantities requiring data from several measurement channels.
6. On-line visualization of measurement results in a chosen way (several typical digital and graphic indicators and various diagrams).
7. The choice of how to use the measurement data, e.g.: archiving, off-line processing.

A lot of measurement programs offered on the market meet many of these requirements. The best of the programs used by the author, Catman by Hottinger Baldwin Messtechnik, is designed to support the products of this manufacturer and allows to service only some of the measurement cards (DAQ) of other manufacturers. In order to integrate devices from other manufacturers in measurement stands built in the Laboratory of the Department of Off-Road Machine and Vehicle Engineering a measurement program was created that allows for the operation of various measuring devices from the level of one application run on one computer operating in the MS Windows environment.

The measurement program was created using the National Instruments LabView package [1]. The choice of this environment was decided by:

- High versatility - many manufacturers of measuring equipment provide complete libraries or at least LabView drivers;
- A wide range of ready-made elements to create an intuitive and functional user interface;
- The ability to create applications using graphical programming;
- The author's previous experience in creating applications in LabVIEW.

Programming in the LabView environment is fundamentally different from coding in higher-level languages (C, C++, Python, et al.). Applications (so-called "virtual instruments") are created in two mutually integrated parts: a front panel in which an interactive user interface is defined (equipped with digital and graphic indicators and controls for data entry or process control) and a system diagram that contains a program code represented by a system of

connected function blocks that implement the desired algorithm. Of course, it is possible to use typical structures such as loops and conditional statements, as well as dividing applications into subprograms (they are defined as separate virtual instruments). This allows the construction of applications with specific functionalities: numerical calculations [1, 2], testing of products [1, 3], testing of a virtual PLC [1, 4].

Software developers who have procedures written in the above-mentioned higher-level languages can implement them in LabView. This programming environment allows the use of directly procedures developed using the Matlab package [1, 5].

The concept of integrated measurement application

The implemented application has a modular structure enabling the execution of a measurement task using the intuitive front panel (Fig. 2).

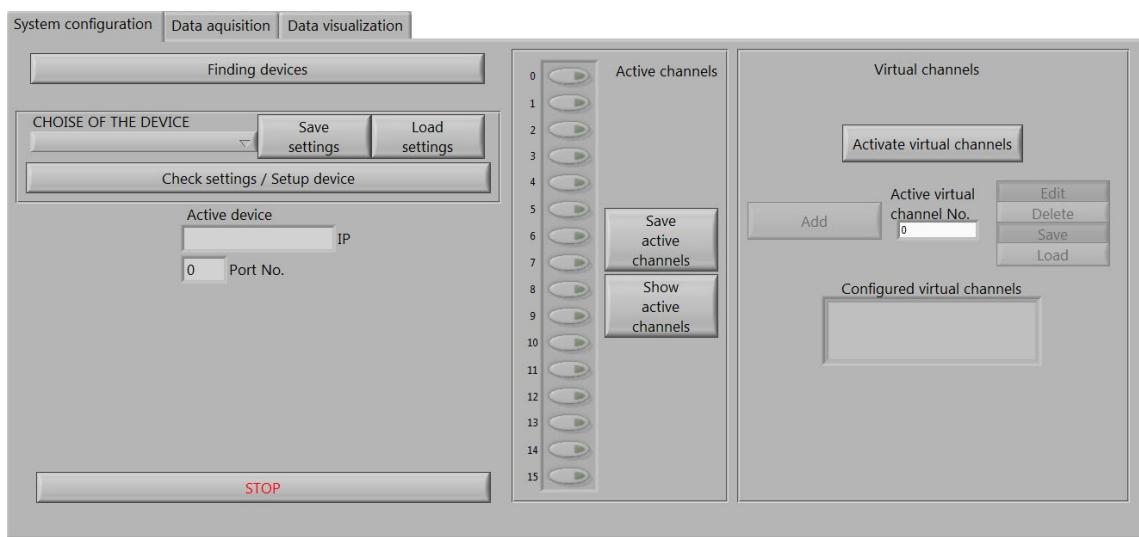


Fig. 2. Panel for devices configuration and set of individual measuring channels

The developed program enables:

1. Initialization communication with the selected measuring device ("Find device" button on main front panel - Fig. 2) - depending on the type of device, it is also possible to select the type of communication interface.
2. Checking the device settings and their correction - each initiated device can be individually configured both by manual correction of individual parameters as well as by loading the configuration saved in an external file using a different program option.
3. When operating multi-channel devices, the program allows setting the parameters of each channel:
 - setting the type of connected sensor,
 - determination of the measuring range and sampling frequency,
 - setup of the dimension of the measured quantity,
 - introduction of sensor characteristics (in this version of the program only by providing gain and offset),
 - specific low-pass filtration parameters (type and bandwidth) - this applies to measuring devices equipped with such filters,
 - setting the sensor power supply parameters from the internal power supply of the measuring device,
 - channel zeroing.
4. Record and read device configuration (to/from mass memory).

5. Activation and deactivation of selected measurement channels.

6. Support for virtual channels:

Virtual channels were created to visualize (and register) on-line quantities that are the result of indirect measurements. Very often, the value of an interesting physical quantity is the result of more or less complex calculations on the values of signals from individual sensors. In addition to controlling and registering interesting physical quantities, such on-line calculations allow you to reduce the work on the development of test reports. The user can define any number of virtual channels using the right part of the front panel (Fig. 2). The limitation of the current version of the program is to support only a few basic mathematical functions (addition, subtraction, multiplication, division and exponentiation - also with the use of introduced constants). Although the LabView package allows the implementation of even complicated numerical calculations [2], in the presented version of the application, to simplify the operation of the equation editor (when defining virtual channels), the implementation of a larger number of mathematical functions has been abandoned. However, this allows for the introduction of more complex characteristics of the sensors used. To avoid rewriting virtual channels many times, the user can save them to a file and read.

7. Acquisition of measurement signals:

A special operator panel (Fig. 3) was developed to configure this stage of the measurement task. The user has the following functions:



Fig. 3. Panel for acquisition of selected values measured during tests

- start and stop measurement,
- specify the duration of measurements,
- choose the diagram scaling method: automatic (adjusts the range of graphically displayed quantities), manual (allows to define the scope of the "Amplitude" axis on the graph),
- zeroing the selected channel,
- set the image refreshment rate of the diagram,
- the ability to activate and deactivate the display of each channel's value on the diagram,
- viewing numerical values of active channels,
- saving results to a file (the function is activated after stopping signal acquisition).

8. Visualization of measurement data:

In the current version of the program, this is a functionality that requires manual loading of a file with a schematic or photo (supported formats: JPG, BMP) and deployments on digital indicators in the appropriate places on the image. By selecting the "Display settings" function, the user can define the format and range of the data displayed in the indicator and assign indicators to the measuring channels. Figure 4 presents the visualization prepared for the needs of the stand for testing the load process with a bucket of a one-bucket loader [6] in the Laboratory of the Department of Off-Road Machine and Vehicle Engineering.

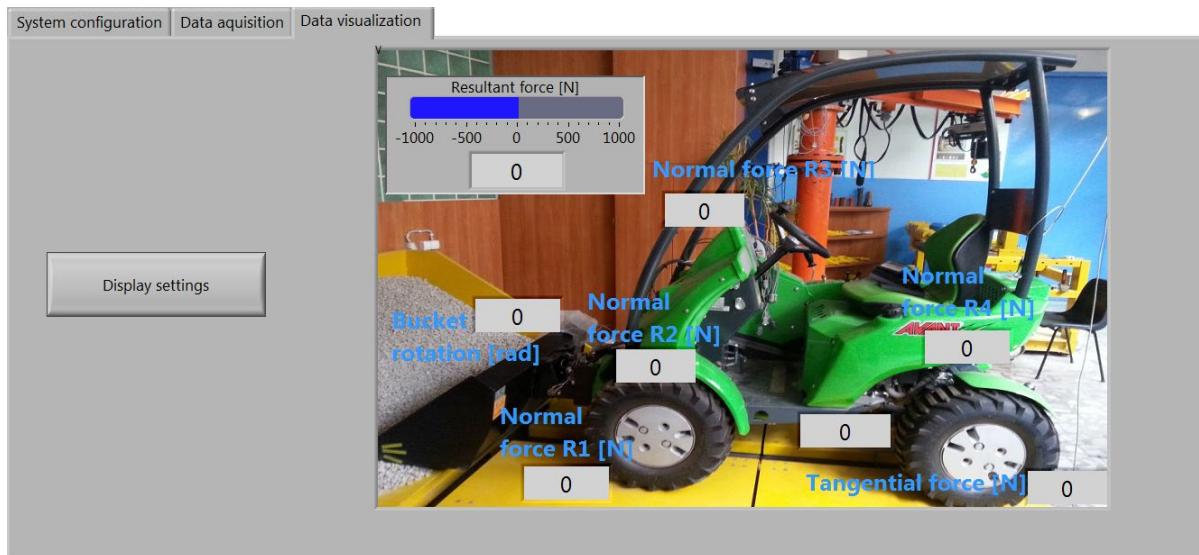


Fig. 4. Panel with an example graphic visualization for acquisition of selected measured quantities (channels)

The functions of the program described above can be implemented thanks to the activated subpanels that appear during the implementation of some program functions (Fig. 5).

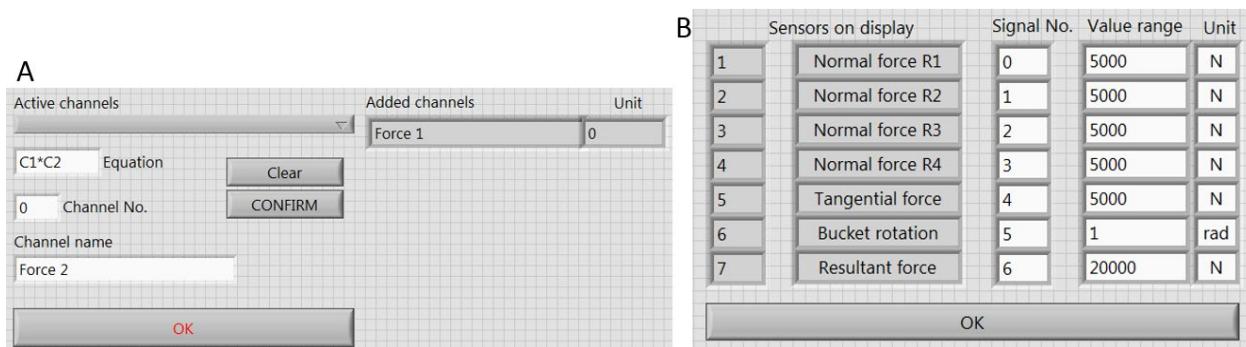


Fig. 5. Selected subpanels for detailed configuration of program functions (A - subpanel for define of virtual channels, B - subpanel for configuration of display settings)

Software functionality tests on a laboratory stand

One of the important functional tests of the developed measurement program was carried out on a laboratory stand equipped with a series of sensors designed to identify the loads of the working tool (bucket), its position and wheel loads (Fig. 6). The loads are measured by strain gauges sensors (cooperating with the QuantumX amplifier), the position of the bucket is identified by potentiometric transducers (during normal operation, the potentiometers also cooperate with the QuantumX amplifier, here a simple DAQ card, NI USB-6009, was used for the test).

The use of virtual channels turned out to be particularly useful for cooperation with a measuring system equipped with a multi-dimensional sensor of forces and moments. The load sensor shown in Figure 6 makes it possible to identify 6 load components, 3 forces (F_x , F_y , F_z) and 3 moments (M_x , M_y , M_z). The developed matrix characteristic of the load transducer [6] (equations 1, 2) allows to identify the components of load, take (filter out) inter-channel crosstalk (between measurement channels), and also eg: calculate the resultant load. The use of virtual channels allows the introduction of all the 36 coefficients of the characteristics, calculating a desired quantities, the visualization and archiving. In Figure 4, the resultant force was calculated using virtual channels.

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 & A_5 & A_6 \\ B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \\ C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\ D_1 & D_2 & D_3 & D_4 & D_5 & D_6 \\ E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \\ F_1 & F_2 & F_3 & F_4 & F_5 & F_6 \end{bmatrix} \begin{bmatrix} U_{Fx} \\ U_{Fy} \\ U_{Fz} \\ U_{Mx} \\ U_{My} \\ U_{Mz} \end{bmatrix} \quad (1)$$

$$\left\{ \begin{array}{l} F_x = A_1 \cdot U_{Fx} + A_2 \cdot U_{Fy} + A_3 \cdot U_{Fz} + A_4 \cdot U_{Mx} + A_5 \cdot U_{My} + A_6 \cdot U_{Mz} \\ F_y = B_1 \cdot U_{Fx} + B_2 \cdot U_{Fy} + B_3 \cdot U_{Fz} + B_4 \cdot U_{Mx} + B_5 \cdot U_{My} + B_6 \cdot U_{Mz} \\ F_z = C_1 \cdot U_{Fx} + C_2 \cdot U_{Fy} + C_3 \cdot U_{Fz} + C_4 \cdot U_{Mx} + C_5 \cdot U_{My} + C_6 \cdot U_{Mz} \\ M_x = D_1 \cdot U_{Fx} + D_2 \cdot U_{Fy} + D_3 \cdot U_{Fz} + D_4 \cdot U_{Mx} + D_5 \cdot U_{My} + D_6 \cdot U_{Mz} \\ M_y = E_1 \cdot U_{Fx} + E_2 \cdot U_{Fy} + E_3 \cdot U_{Fz} + E_4 \cdot U_{Mx} + E_5 \cdot U_{My} + E_6 \cdot U_{Mz} \\ M_z = F_1 \cdot U_{Fx} + F_2 \cdot U_{Fy} + F_3 \cdot U_{Fz} + F_4 \cdot U_{Mx} + F_5 \cdot U_{My} + F_6 \cdot U_{Mz} \end{array} \right. \quad (2)$$

Conclusions after preliminary tests of the developed measurement application

The conducted laboratory tests carried out allowed to check the developed program and correct minor application errors. They confirmed the possibilities of building a graphical measurement environment to support complex measurement systems equipped with various measuring devices. In the described version, the program allows support for Hottinger Baldwin Messtechnik amplifiers (MGCPlus, Spider8, QuantumX), several multifunction measuring cards (DAQ) from National Instruments and Tektronix digital multimeters and oscilloscopes. Unfortunately, the anticipated constraints resulting from the capabilities of measuring devices were also confirmed, especially when using communication interfaces with significantly different data transmission rates. During testing at high sampling frequencies, a larger number of measurement signals proved to be a significant limitation for the computer used during measurements as a control unit. The tests carried out in the Windows environment have shown that in order to increase the requirements for the number of measured quantities, the amount of data that must be processed (all online calculations: taking into account the characteristics and virtual channels) visualized and stored in the system buffer for possible archiving. All this requires a computer with a powerful processor and lots of memory. The authors also noted the need to extend the program's capabilities to cache data in the computer's mass memory (e.g. a high-speed hard drive) so that the buffer size in the memory would not limit the amount of data to be stored during longer tests.

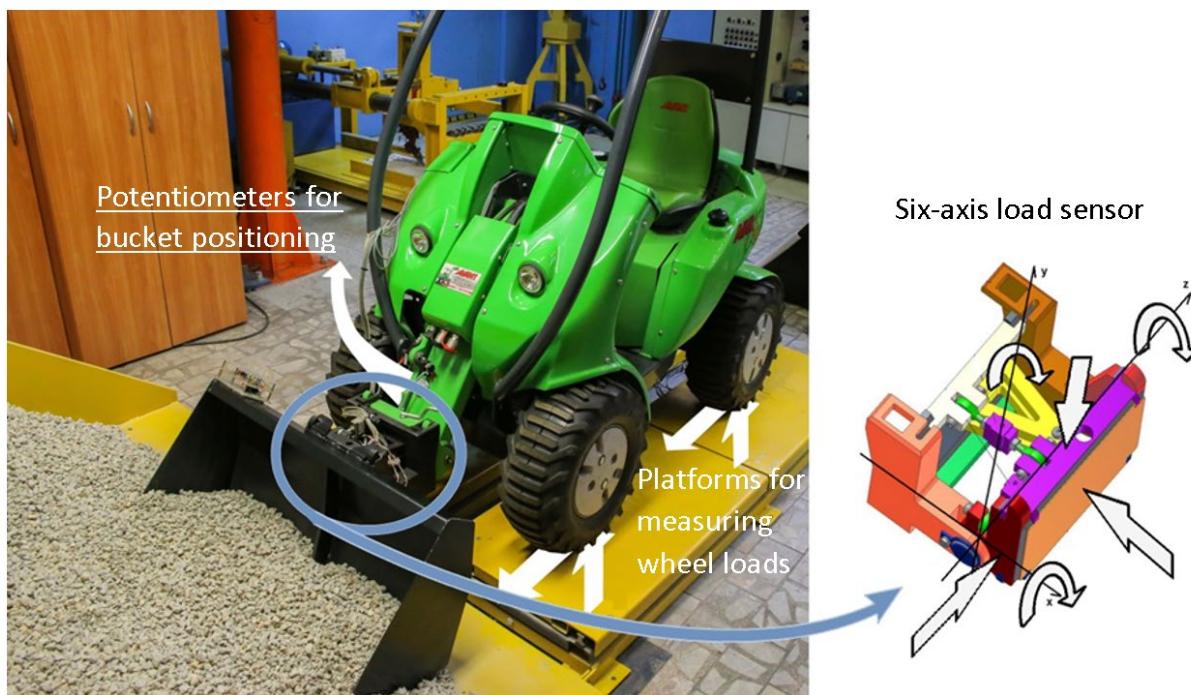


Fig. 6. A new-generation test stand with a measuring system to identify the loads of the work tool (bucket) and wheels of the undercarriage and the position of bucket [6]

Summary

The presented computer measuring program created in the LabVIEW environment enables the implementation of complex research programs with the visualization and registration of all the measured quantities. The open structure of the system allows for a relatively simple expansion with further elements: e.g. for automatic control of the executive elements, support for other measuring devices. Particular attention requires the expansion of the equation editor to define virtual channels and the use of mass storage as a buffer with more data.

References

- [1] National Instruments (2017). LabVIEW, <http://www.ni.com/LabVIEW>.
- [2] A. Suliman, Applications of Virtual Modules in Numerical Analysis, Open Access Library Journal, 3, (2016) 1-8.
- [3] H. Mageed and A. El-Rifaie, "Electrical Metrology Applications of LabVIEW Software," Journal of Software Engineering and Applications, Vol. 6 No. 3, 2013, pp. 113-120.
<https://doi.org/10.4236/jsea.2013.63015>
- [4] M. Alia, T. Younes and M. Zalata, Development of Equivalent Virtual Instruments to PLC Functions and Networks Journal of Software Engineering and Applications, Vol. 4 No. 3, 2011, pp. 172-180. <https://doi.org/10.4236/jsea.2011.43019>
- [5] R. Bitter, T. Mohiuddin and M. Nawrocki, LabVIEW Advance Programming Techniques, Second Edition, CRC Press, Boca Raton, 2006. <https://doi.org/10.1201/9780849333255>
- [6] R. Czabanowski, Ł. Leśniak, A. Łabuda, The computer measuring system of the stand for testing the load process with a bucket of a one-bucket loader, AUTOBUSY, No. 12, 2016 (in Polish).