Robert SZULIM [1], Maria SZPERKOWSKA [2], Władysław SIELUK [2]
[1] UNIVERSITY OF ZIELONA GORA, INSTITUTE OF ELECTRICAL METROLOGY, 50 PODGÓRNA ST., 65-246 ZIELONA GORA
[2] RELPOL S.A. ZAKŁAD POLON, 11 BROWARNA ST., 65-849 ZIELONA GORA

Multiplatform dedicated control – measurement systems

Abstract

The paper presents some issues with developing dedicated software solutions for control – measurement systems equipped with Graphical User Interface (GUI) and touchscreen dedicated to work under different operating systems like MS Windows and Linux. Some important issues of preparing the system to work under different operating system environment using common software code are described. As an example of a practical solution, the control system of the radiation monitor produced by Relpol S.A. built using the QT multi-platform development environment is presented.

Keywords: multiplatform systems, control – measurement systems, radiation monitoring systems.

1. Introduction

Currently used ready - made control – measurement (C-M) systems are often prepared to allow implementation in multiple operating system environments, e.g. Microsoft Windows or Linux. This is usually caused by the requirements of recipients of this type of equipment and systems. Software solutions often have unique, dedicated features adapted to the requirements of functionality and parameters of the device or system. This makes building software applications for this type of solutions difficult because most of the available designer tools are not prepared to achieve such a functionality.

The requirements for such applications are usually related with the following functionalities:

- Communication with external devices via various interfaces, such as RS-232, Ethernet, Bluetooth, USB or Wi-Fi.
- Multi-language graphical user interface (GUI) capable to use touch control.
- Remote supervisory and control of the device using the web browser and Internet connection.
- Storing history of device operation as files in the non-volatile memory.

A commonly used approach is to use the so-called panel PC computers [1] – Fig. 1. They are equipped with an Intel x86 compatible processor, making it easy to start a new project because there are number of possible operating systems to use on such as Windows or Linux.

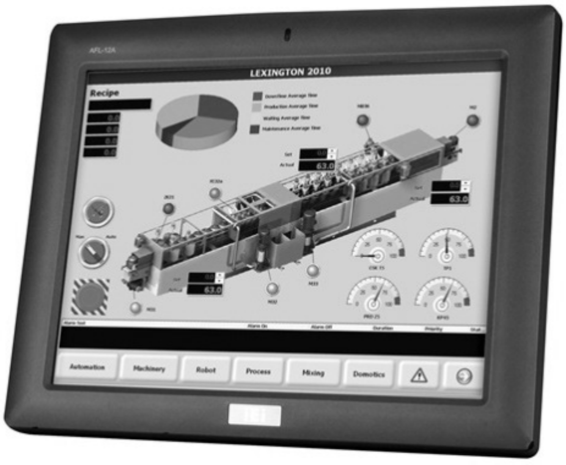


Fig. 1. Panel PC (source http://ieiworld.com)

Computers of this type have several serial interfaces, Ethernet and USB, and can be equipped with an LCD screen with touch control. The construction is similar to the classic PC, so starting the system based on such a computer is not very complicated. Operating system and C-M software can be installed in a CF memory card imitating the classical hard drive. These types of computers can be used where there is a need for the rapid launch of a dedicated control - measurement system using known and popular operating systems, such as Windows or Linux.

2. Multiplatform systems

Developing applications for C–M systems for using in different operating systems causes many problems related primarily to the lack of compatibility and portability of the resulting application code. There is however a number of solutions to help the designer in building applications designed to work under different operating systems. Three possible approaches are briefly described below.

2.1. Compiling common source code

Writing a common source code possible to compile on different operating systems with extensive functionality is difficult. To some extent, this is possible using the ANSI C programming language. However this way it will be difficult to build a program with a graphical user interface, in which the code can be easily compiled for different platforms. Of course, it is possible to use various types of solutions based on conditional compilation, which means that, depending on the target different portions of the source code will be used. In practice, this results in the need of writing different pieces of code performing the same functions several times for various target platforms. Combining these pieces of code together, sharing the values of variables between functions, code updates and other programming functionalities are a complicated matter.

2.2. Solutions using interpreters and intermediate code

Solutions using code interpreters may use technology like Oracle Java or Microsoft .Net. In both cases, the source code is compiled into the so-called intermediate code, distributed to the target system. To run the program, the presence of the so-called virtual machine in the operating system that interprets the code is required. For many operating systems, appropriate versions of Java virtual machines are developed. They are also available for most of the Windows and Linux based systems. It is possible to use the same code to run programs with a graphical interface. However, there are some problems with supporting for serial interfaces, which is a subject of complains of many developers.

Another solution is to use the Microsoft .Net Framework. It is a dynamically developed technology for building software solutions with the ability to use multiple programming languages like C++, C# or Visual Basic. There is significant limitation in the fact that Microsoft has not developed fully compatible runtime library for Linux systems. The .Net Core available as Open – Source project is still not ready to be the equivalence of the .Net Framework library.

As an alternative to .Net Open-Source .Mono platform could be used. This platform allows the compilation of programs for Windows and Linux using .Net code. Unfortunately, .Mono platform still has a number of limitations that hinder writing complex systems.

2.3. Compiling the code using meta objects

It is possible to write complex programs that can be compiled into native applications for General Purpose Operating Systems (GPOS) like Windows or Linux using common code of C++ programming language. This task could be carried out by QT tool [2] and allows building application of C-M systems using GUI, communication procedures with serial interfaces, protocol stack, concurrency programming, printing and multimedia features. Compilation in this case takes place in two stages. The programmer writes a code in C ++ language as well as functions and classes that are provided by the supplier of QT utility. It is a library of common classes and functions to use for different compilation platforms. During compilation, meta C++ objects of the library classes and functions are translated into native C++ code designed to compile for a specific operating system using a compiler specific for this platform, as GCC [3] or Visual Studio [4]. The program code should not use the specific native Windows or Linux API functions because the ability to compile for other systems will be lost. It is also not possible to refer to the specific functions of the system drivers. The generated native executable code is able to run without any additional interpreters, virtual machines or indirect emulators, and is fast in action.

3. Stationary radiation monitor

Stationary radiation monitor – SMP (Polish brand name) produced by Relpol SA is a device designed to detect exceeding the maximum permissible level of gamma and neutron radiation. These devices can be used, among others, in such locations as airports [5], border crossings, warehouses, ironworks etc. Exceeding the acceptable level of radiation is indicated in several ways, depending on customer needs and configuration. This may be a sound signal, light, notification of other systems and the presentation of the status on the operation panel.

The SMP device cooperates with a set of radiation detectors, which are controlled via RS-485 interface. Detectors are installed close to the zone where every object must be checked against radiological contamination. The system is controlled by the dedicated C-M system software – Fig. 2.

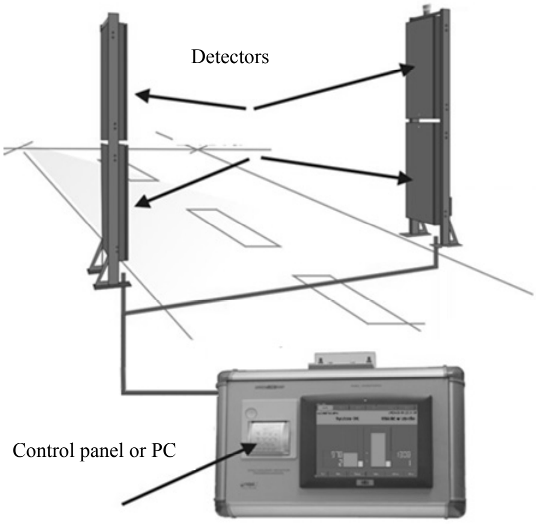


Fig. 2. Stationary Radiation Monitor

3.1. Control – measurement system software

The device is usually controlled by the C - M system in the form of the software running on the panel PC under the control of Linux system. Windows - based solution may also be used on classical PC computers depending on the needs of the customers of the system.

The software was written in C ++ code using QT tools. Using classes library provided by the QT tool manufacturer, a real-time system, with a well-defined intervals to communicate with detectors via RS485 was launched - Fig. 3.

The number of detectors may vary depending on the configuration. There may be up to 32 gamma and neutron detectors. The system gathers information about the state of detectors like counts and status (alarm, failure). The state of the system is presented on the control panel screen, respective signals are sent to control outputs while the information are recorded to the history files.

At the same time system must be able to communicate with the supervising SMP-Studio system. It is also possible to present the state and control of the work of the device using the Web browser over Internet. This requires WWW server to cooperate with the SMP control – measurement system.

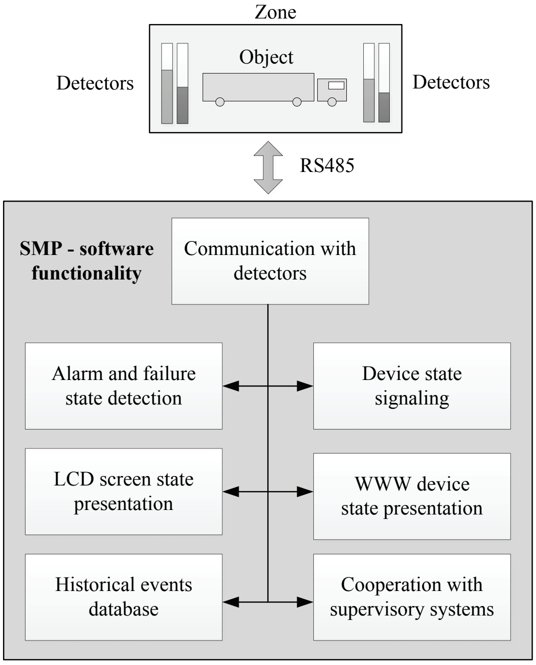
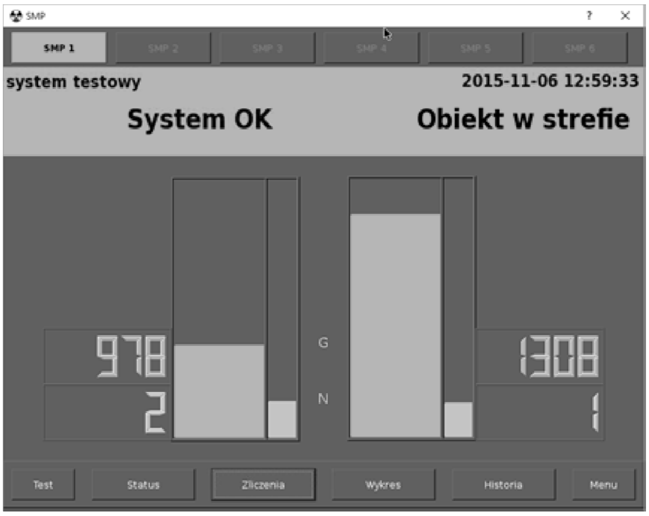

Zone
Object
Detectors
Detectors
RS485
SMP - software functionality
Communication with detectors
Alarm and failure state detection
Device state signaling
LCD screen state presentation
WWW device state presentation
Historical events database
Cooperation with supervisory systems

Fig. 3. SMP device construction

All functionalities must work simultaneously. This forces the system to work concurrently, which implies the use of the multithreaded programming approach with all the software overhead that goes with it, like synchronizing the access to shared resources.

3.2. Graphical user interface

One of the requirements for modern C-M systems is the presentation of data using a GUI to enable easy and quick access to device status and change its settings. The software can also be run and controlled with the use of touch screen. In the case of the panel PC as a startup platform, there is no classical keyboard and mouse installed in the system. The entire application control is done using the touch screen to simulate a the mouse. The control application must be prepared according to certain rules related to position and size of active elements on the screen. Some of the graphic elements must be designed in a different way than in traditional applications with a GUI - Fig. 4.


system testowy 2015-11-06 12:59:33

System OK Obiekt w strefie

Fig. 4. SMP system graphical user interface

3.3. Work under Linux and Windows systems

The C-M system built using QT tool allows preparing two versions of executable code without changing the source code. To accomplish this, the GCC [4] compiler available for both operating systems was used. Executable files of the application have a size of about 1.5 MB. To start the application, a set of QT library files and tools for each target operating system is required. The library files are designed to integrate the operating system services, such as communication interfaces, multimedia, printing, graphic interface, etc. The size of all the library files depends on the number of the used software functionality from a few to tens of megabytes. In the case of SMP it is about 50 MB. The developed software and files library can be run with the same functionality under Linux and Windows in different versions and distributions. To work under Linux system, the graphics module libraries X11 (X Window) are required. It is possible to use different windows managers like XFCE, KDE, Gnome, FVWM and others. Multimedia features, such as the sound emission and printing functions, are also supported. The complete system runs under Linux Slackware Intel x86 compatible using 256 MB RAM memory and 4GB CF disk. Launching System for Windows requires much more RAM and much larger hard drive or CF card.

3.4. Cooperation with SMP-Studio

SMP devices can operate standalone but in many cases they are installed in distributed zones of the same object (ex. airports). Often in such locations there is a need to collect documentation contamination alarm occurrence in the form of appropriate reports. In many cases it is also required to document the event in the form of the video record. The SMP-Studio system was prepared for such purposes – Fig. 5. It can work with SMP devices and IP TV cameras. If an alarm is triggered by SMP, video material for the event is stored. It is possible to generate the report, along with a photo of the object that caused the alarm. This especially applies to objects like airports. The system has been installed in several large passenger airports in Poland, among the others: Warsaw, Rzeszow, Modlin. The system allows keeping track of movement within the observed zone in the form of a video preview, which can be observed by the staff of the object. It is possible to run multiple workstations in distributed locations.

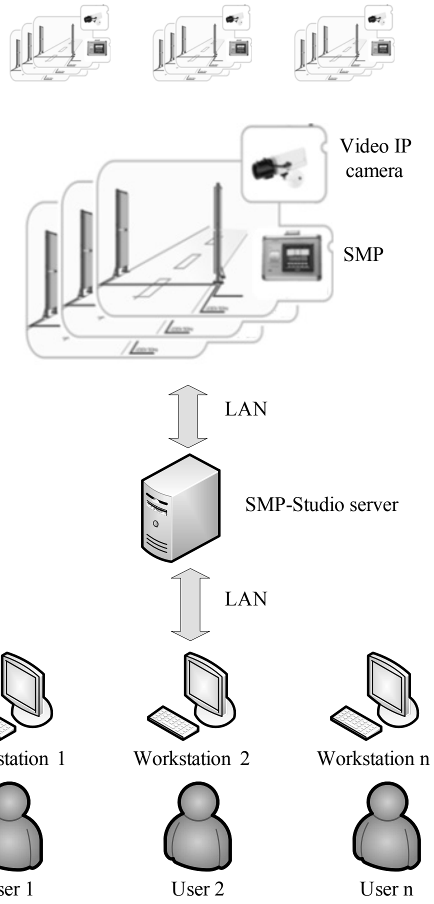


Fig. 5. SMP system work

4. Summary

By using QT tools for building the software it was possible to prepare the system ready to run under Windows and Linux operating system without elaborating a separate source code. The QT tool and the GCC compiler are available for use on the principles of Open – Source license. This solution is cost effective because Open – Source tools are distributed free of charge. The system can be used in installations based on Linux or Windows. This allows the manufacturer to adjust the system to the specific requirements of customers. The latter can use their own PC to run the system, which reduces implementation costs.

5. References

[1] http://ieiworld.com, IEI Integration Corp. Home
[2] http://qt-project.org, Project QT, 2016
[3] https://www.visualstudio.com/, Microsoft Visual Studio, 2016
[4] https://gcc.gnu.org/, GCC, the GNU Compiler Collection, 2016
[5] Michta E., Szulim R.: Airports radiation protection, Measurement, Automation, Monitoring, 2011, Vol. 57, no 11, p. 1304—1306 (in Polish).

Received: 02.04.2016 *Paper reviewed* *Accepted: 01.06.2016*

Robert SZULIM, PhD, eng.

Graduated from the Faculty of Electrical Engineering College in Zielona Gora in 1995. After graduation, he started working at the Institute of Metrology, Electronics and Computer Science at the Higher School of Engineering in Zielona Gora (now the University of Zielona Góra). He deals with issues related to computer networks, measurement and control systems, data warehousing and application of artificial intelligence.

e-mail: R.Szulim@imei.uz.zgora.pl

Maria SZPERKOWSKA, MSc, eng.

Higher education, graduated from the Faculty of Electrical Engineering Poznan University of Technology in Poznan in 1975. She is the General Manager of Polon division of Relpol SA company. The company is engaged in designing, manufacturing and selling electromagnetic relays, digital security systems, automation, measurement, control, communication and registration for electrical MV substations systems and radiation monitoring.

e-mail: Szperkowska.M@relpol.com.pl

Władysław SIELUK

Head of the Design Department of Polon division of the Relpol SA company. Coordinates and supervises the work of designers and developers who develop new solutions for equipment manufactured in the company, in the field of radiation portal monitors and digital security systems used in medium voltage power switchgears.

e-mail: Sieluk.W@relpol.com.pl