ACTA TECHNICA CORVINIENSIS
– Bulletin of Engineering
Tome IX [2016] , Fascicule 1 [January – March]
ISSN: 2067 – 3809

[1.]Dávid ANTAL, [2.]Gábor PETO, [3.]Attila SIMON

DEVELOPMENT OF COST EFFICIENT, OPEN SOURCE BASED BULIDNG MECHATRONICS SYSTEMS

[1-3.] Department of Electrical Engineering and Mechatronics, Faculty of Engineering,
University of Debrecen, Otemeto street 2-4, Debrecen, HUNGARY

Abstract: The European Union is constantly striving to reduce energy use of buildings, therefore constant regulations put into force in the energy sector. In response to these changes the population has to adapt, which means renovating or automating the heating system of the building sites. Industrial control systems required for the automation are inaccessible to the public considering their high price, in contrast, open and closed source field controllers offer a good and cheap, but limited capability alternative. This article presents a new method to achieve cost effective building automation alternative for small and medium sized buildings. Comparison of closed source and open source based building automation system is introduced.

Keywords: Energy consumption, Open source, Closed source, Building automation

INTRODUCTION

In the European Union the annual energy consumption divided into three main sectors: buildings, industrial and services. The buildings' energy consumption accounts for the nearly 40 per cent of total energy usage. According to the EPBD directives, the energy usage have to reduce continuously, and by the mid-range term after 2020 only zero energy buildings can be build.

The building energy usage's main part the heating, ventilating and air conditioning system (later HVAC), and the sector holds the largest untapped potential to save energy. To optimize energy usage, there are two opportunities:

1. Modifying the building's physical parameters by renewing heat transfer surfaces
2. HVAC system optimization by field controllers

Modifying any building site's physical parameters are costly compared to HVAC system optimization. However, some robust HVAC controllers (e.g. PLC-s) on the market can nearly cost the same as a full building renewal with the disadvantage of their inflexibility. This means that each product can only use its own manufacturer's accessories and software that is required for programming and to achieve sufficient operating conditions.

In the following Open and Closed source system properties are presented, including its physical realization, and finally compare it.

OPEN-SOURCE SOFTWARE AND HARDWARE

"Open-Source Software (OSS) refers to software systems which are free to use and whose source code is fully accessible to anyone who is interested. Most OSS systems start out with a developer who wants to solve his or her own particular problem and makes the solution (system) available to others for free. Because it is free, it often attracts many users who have a similar problem, and because of the free access of source code, some interested users become co-developers by extending or improving the initial system. Together with the original developer, users and co-developers create a collaborative OSS community around the system. Without such OSS communities, OSS projects are not likely to be successful. Most OSS systems are not necessarily carefully designed in advance. They evolve in response to the needs of users in the OSS community, and the evolution is carried out by contributing (co-)developers of the same community. Although the evolution of an OSS system is not well planned, "giving users of a product access to its source code and the right to create derivative works allows them to help themselves, and encourages natural product evolution as well as preplanned product design."[1].

Through the years, the open source projects became more effective, reliable and the community created standards like GNU GPL (GNU General Public License) that warrants the open source application rights. As the open source software begins to conquer the world, so does its own hardware such as the Arduino platforms. Numerous commercial hardware are already available.

These platforms are used mostly by hobby electronics and universities for education purposes, because they are easy to use compared to others like ATMEL or Microchip. Furthermore, some companies saw a great opportunity and they are already using these products with their own hardware. This way they don't need highly trained developers and programmers.

CLOSED-SOURCE SOFTWARE AND HARDWARE

These systems are set up by companies specializing in closed-source software and hardware development. This process is difficult, time and money consuming, so the clients have to put their hands in their wallets for these products. In return, clients will possess a gadget worth to pay more than open-source systems. It is possible to choose from wide variety of additional accessories and modules, thereby avoiding to pay for unnecessary modules and accessories. The owner also receives customer support with the development team's professional experience; hence the customers never have to repair the system at their own risk. Not only the systems working exactly as they need to, but also more resistant to exterior attacks and unauthorized accesses. The only disadvantage is that the source code never gets to the customer, so they do not know how it works. Only the company's employees can modify it if so requested by the client.

ARDUINO BASED DESIGN CONCEPT

The concept was to create a multi-functional device specially designed for non-industrial environment.

The main purposes are the following:

1) cheap energy control for buildings
2) measurement data acquisition
3) user friendly
4) plug & play
5) compatibility with data analyzing programs

The aim is data acquisition and control system development that is available for even an average household, and provide a user-friendly interface to handle measurement data.

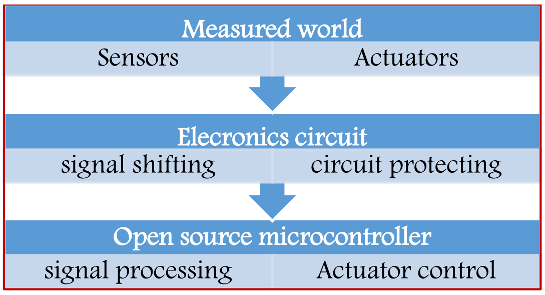

Figure 1. System build-up layers

The first layer represents the connection to the outside world with precision sensors and actuators. Sensors convert the non-electric parameters – temperature, humidity, and global radiation etc. - of the environment to electric signals. These signals can be analog or digital, depending on the application. The sensor selection is prominently influences the system's behavior, because a needlessly accurate sensor increase the cost, or an inaccurate measurement can cause unexpected events or failures.

The second level is the electronic shield that receives the signals from the sensors to shift signals for the operating slope, and amplifies the control signals for the actuators. The shield has another function to protect the control unit from harmful voltage spikes, electrostatic discharges (ESR) and also provides regulated voltage source, for all of the connected devices.

The third part is the open-source hardware platform, named Arduino that is a physical computing platform based on a simple AVR microcontroller board, and a development environment for writing software for the board. Using the platform makes the development easier and time effective, because its programming language named Wiring designed for rapid circuit prototyping. Arduino boards have a common connection pinout configuration that reached a wide range of the electrical community. So that pin configuration regards a standard, and usable of that makes the shield more flexible.

The development started as a fully custom-made PCB co-operating with Arduino compatible with any type of sensors. But this direction has soon turned out to be a dead-end, because designing and programming became too time-consuming and difficult.

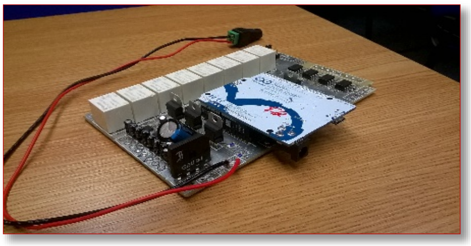

Figure 2. UNI_IO circuit, with Arduino Leonardo

The project's current aim is to apply a widely available development kit into an extension board compatible with the most commonly used sensors and actuators. The first version of the PCB (named as: UNI_IO) is shown at Figure 2.

Table 1. UNI_IO configuration

Name	Input	Output
Analog	8 ESD protected	8 PWM to Analog
Digital	8 isolated	8 relay
Serial	EIA 485, 1 Wire	
I/O	2 pull-up 5V I/O channel	

The PCB has 2 different connection ports, one for Arduino compatible, and the other to connect various controllers that meet the 3.3V-5V operating voltage requirement. Table 1 shows the specification of the PCB.

The digital inputs are isolated by an opto-coupler, so that protected from overvoltage. For the inputs, several digital signal sensors can connect, with 30V maximal input voltage. The digital outputs are relay output, which can switch power to drive devices.

The analog inputs are protected from ESD, and the voltage divided by 3 to avoid overvoltage failures. Analog inputs are an operational amplifier output with 2 times gain to create analog output from PWM (Pulse width modulated) signal for 0-10V transmitting.

The EIA485 converter placed on the PCB for the long distance communication (up to 1000-1200m), for example Modbus communication. Two pulled up input/output placed too, for mostly the 1-Wire communication.

And finally the PCB provide external and internal voltage levels: 12V, 5V, 3.3V to operate the board itself, and the sensors.

The Arduino Leonardo cooperating the UNI_IO panel controller is ready to use, because in addition to the circuit, a control software is also made, which is built around the Modbus communication protocol. The idea was to develop a suitable program capable of handling household system signs such as water meters, gas meters and also HVAC consumption components like valves, motors and boilers. The success of the program shows that structure only has to be installed once on the controller, then any modifications are possible remotely via Modbus or even with a nearby laptop using a simple USB connection in real time.

The cycle period within full use is between 5-700 millisecond sparing additional resources if the aim is to measure once every second. In the previous sections only one half of the concept have been presented (field unit), but it requires an additional control unit which processes the data, stores and implements the control algorithms, such as PID control. For this purpose we can use Raspberry Pi, which is an open source-based AMR embedded system on which to run the control software.

Countless open-source controller and SCADA (supervisory control and data acquisition) software available on the market. For instance, the software named ScadaBR is capable of supervising an average household. The advantage of the SCADA based supervisor is the user friendly graphic interface helping to monitor and control the internal house environment easily and remotely.

CLOSED SOURCE DESIGN CONCEPT

The name of the product series is Energy Mentor that rests entirely on closed-source foundations in contrast to the Uni_IO system and hardware. The main difference between the two embedded systems is that the EM has an integrated stand-alone central computing unit. The aim of its design was to create a device with the most versatile utility in the field of measurement and control purposes. As a result it can be applied to both home and industrial environments.

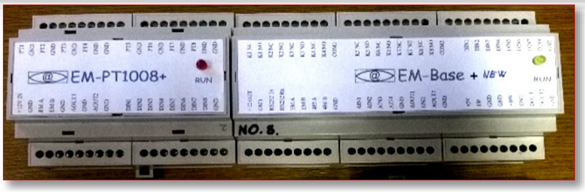

Figure 3. Energy Mentor series

Main advantages:

- Additional expansion modules on demand
- Low-power consumption, economic
- DIN-rail mountable, robust
- suitable to form network with multiple CPU-s and modules
- Programmable in RTOS
- Stand-alone system

The heart of the hardware is an AVR Xmega microcontroller with +3,3Vdc supply voltage according to industrial trends. The microcontroller has large amount of program-memory along with I/O ports compared to open source systems. This way it supports far more devices and suitable applications.

Currently the field controller has two support field IO modules, one with four and the other with eight channels. These two modules are rather useful in industrial and special environments.

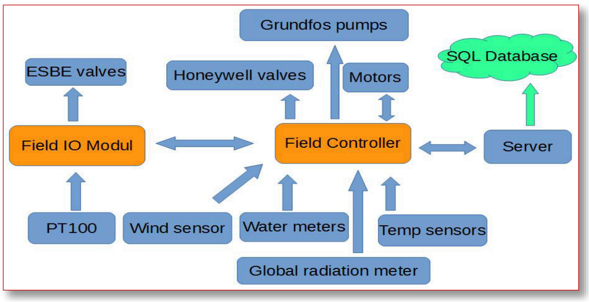

Figure 4. Test system construction

Such as testing of solar panels or engines, with long-life, high temperature range and accuracy demand fulfilled by PT100 heat sensors.

Table 2. Field Controller configuration

Name	Input	Output
Analog	4 ESD protected	2 PWM to Analog
Digital	8 isolated	2 TTL, 8 relay
Serial	EIA 485, EIA-RS232C, I2C, 1 Wire, Ethernet TCP/IP	
I/O	2 pull-up 5V I/O channel	

As shown in the configuration table (Table 2) a wide range of protocols and data transmission systems are available. This makes it easy to deploy an Ethernet TCP/IP or Ethernet / RS485 Modbus communication network. By using the Modbus protocol, the devices can be provided with a unique identifier, as well as Master or Slave functions. The Master device controls and manages the slaves together.

The development of the device is currently in the testing process of the second generation. The devices withstood extended testing between both laboratory and production environment conditions. Within the stress-test a hybrid photovoltaic solar-thermal system consisting of pumps, linear motors and valves had been controlled and measured. Since 2014, the system is working properly.

The firmware development started in Basic language, but is already in the process of rewriting into C language as well. In addition, a Real Time Operating System (RTOS) implementation is in the future plans to improve schedule processing flexibility.

CONCLUSION

As you can see in Table 3, the advantages of using Open Source systems cheapness, faster development production, products require less energy to invest in however, a less stable and efficient system is obtained. The closed source products in contrast provide more favorable physical designs, better communication capabilities and performance which is associated with robustness and reliability. Due to higher performance C.S. systems are able to accomplish the specific and unique tasks, such as: a central controlling unit for renewable energy systems, or HVAC systems.

Table 3. Comparison between systems

Name	Input	Output
Open-Source	Prototype, free source code, easy programming	unreliable, ineffective, non-unique solutions
Closed-Source	robust, reliable, high performance, huge I/O, wide range of communication	complex programming knowledge, time and money consuming design

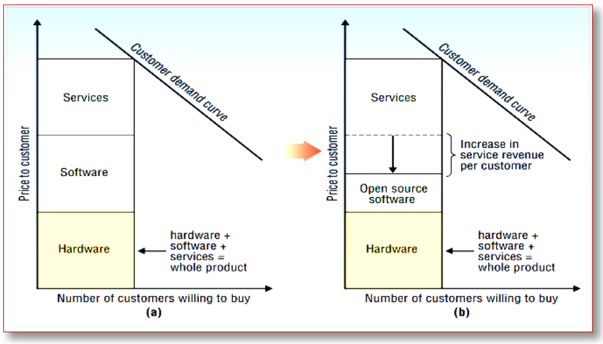
Customer demand curve
Services
Software
Hardware
Price to customer
hardware + software + services = whole product
Number of customers willing to buy
(a)
Customer demand curve
Services
Increase in service revenue per customer
Open source software
Hardware
Price to customer
hardware + software + services = whole product
Number of customers willing to buy
(b)

Figure 5. Price comparison, RIEHLE, Dirk. The economic motivation of open source software: Stakeholder perspectives, page 26.

But overall, we can say that open source systems can be a good alternatives for closed source systems till a certain level of need. For example if more I/O ports or faster operating speed is required closed source is a better choice. In turn open source devices are cheaper solutions that can be seen in Figure 5. This great advantage can be decisive for the appearance on the market.

Acknowledgment

The work is supported and financed by Electrical Engineering and Mechatronics Department, Faulty of Engineering, University of Debrecen.

References

[1.] NAKAKOJI, Kumiyo, et al. Evolution patterns of open-source software systems and communities. In: Proceedings of the international workshop on Principles of software evolution. ACM, 2002. p. 76-85.

[2.] THOMPSON, Clive. Build it. Share it. Profit. Can open source hardware work?, Work, 2011, 10: 08.

[3.] RIEHLE, Dirk. The economic motivation of open source software: Stakeholder perspectives. Computer, 2007, 40.4: 25-32.

[4.] MODBUS, I. D. A. Modbus application protocol specification v1. 1a. North Grafton, Massachusetts (www. modbus. org/specs. php), 2004.

[5.] PEARCE, Joshua M. Open-source lab: How to build your own hardware and reduce research costs. Newnes, 2013.

[6.] Szasz Cs, Husi G: The intelligent building definition: a central-european approach In: IEEE System Integration (SII), 2014 IEEE/SICE International Symposium. Tokyo, Japan, 2014.12.13-2014.12.15. [s. l.]: IEEE, 2014. pp. 216-221. (ISBN:978-1-4799-6942-5)

[7.] Husi G: The latest research results of the Intelligent buildings group in the DEnzero project In: IEEE System Integration (SII), 2014 IEEE/SICE International Symposium. Tokyo, Japan, 2014.12.13-2014.12.15. [s. l.]: IEEE, 2014. pp. 240-244. (ISBN:978-1-4799-6942-5)

[8.] Vesselenyi T; Bungau C; Husi G: Research equipment and computation techniques for Intelligent Building applications In: IEEE System Integration (SII), 2014 IEEE/SICE International Symposium. Tokyo, Japan, 2014.12.13-2014.12.15. [s. l.]: IEEE, 2014. pp. 228-233. (ISBN:978-1-4799-6942-5)

[9.] Géza Husi; Nóra Tóth; Bungau Constantin; Vesselényi Tiberiu: Intelbuild to be Continued - Intelligent Building Infrastructure Solutions for the New Denzero Intelbuild Future Projects at University of Oradea Nagyvárad-Félixfürdő, 2013.06.01. - 2013.05.30. (2013)

[10.] Géza Husi; Csaba Szász; Hideki Hasimoto Application of reconfigurable hardware technology in the development and implementation of building automation systems Environmental Engineering and Management Journal (ISSN: 1582-9596) (eISSN: 1843-3707) 13: (11) pp. 2899-2905. (2014)

[11.] Emese Bánóczy, Péter Tamás Szemes, Péter Korondi: Simulation of building renovation's return in Energy Plus, Environmental Engineering and Management Journal 13: (11) pp. 2743-2748.

[12.] SCADABR, Manual. Disponível em: http://ufpr. dl. sourceforge. net/project/scadabr/Software/Docs/Manual%20ScadaBR.pdf. Acesso em, 2014, 2

[13.] RIEHLE, Dirk. The economic motivation of open source software: Stakeholder perspectives. Computer, 2007, 40.4: 25-32.

[14.] BANZI, Massimo. Getting started with Arduino. "O'Reilly Media, Inc.", 2011.