Further decrease of PC-to-PC SCADA implementation costs using credit-card sized computers

Avdo Hercegovac
Infinity Mesh doo
Sarajevo, Bosnia and Herzegovina
avdo.hercegovac@gmail.com

Elmir Babovic
Faculty of Information Technologies
University Dzemal Bijedic
Mostar, Bosnia and Herzegovina
elmir.babovic@edu.fit.ba

Abstract—**In this research, authors proposed extension of the PC-to-PC SCADA concept, by using credit card sized computers. PC-to-PC SCADA concept decreases total SCADA implementation cost by replacing expensive sophisticated PLC's on RTU side, and replacing them with inexpensive general - purpose PC configuration, thus decreasing total implementation costs as well as enabling the system to include sophisticated control software with elements or artificial intelligence on remote side. This research further improves this concept by proposing to replace RTU side with credit-card sized computers with custom hardware interface in order to controls the object/target device.**

Keywords-PC-to-PC SCADA, artificial intelligence in remote control, credit-card sized PC

I. Introduction

This research represents further improvement of the PC-to-PC SCADA concept first proposed in 2009. [1] The concept was based on the idea to place standard PC configuration on both Master Terminal Unit (MTU) and Remote Terminal Unit (RTU) side of Supervisory Control And Data Acquisition (SCADA) system, thus to decrease total SCADA development, implementation and maintenance costs. The possibility to include elements of artificial intelligence on RTU side which was not possible using Programmable Logic Controller (PLC's) represents a further advantage. Nowadays, expansion of credit-card sized computers allowed this concept to go a step further in decreasing implementation costs. Unlike typical SCADA control system which consists of one or more remote terminal units (RTU or PLCs) connected to a variety of sensors and actuators, and relaying information to a master station [2], this concept attempts to further decrease SCADA concept with improvement of Artificial Intelligence (AI) capabilities. Recent trends attempt to include AI and computational intelligence (CI) into IT support for industrial processes. [3]

Considering SCADA as much wider concept used as one of main industrial remote control concept already for decades, this research does not aim to decrease its significance. This research rather propose and expand special type of SCADA, initially named PC-to-PC SCADA in certain number of SCADA systems in which additional software functionalities are required. Such cases are special systems requiring more sophisticated software support, high level of independence and autonomy of RTU as well as the elements of Artificial Intelligence on RTU side.

II. Proposed concept

A. *Credit-card sized computers architecture*

Modern credit-card sized computers are widely accessible and relatively low priced. From the architectural point of view, they are based on Harvard architecture rather than Von Neumann. This means that the data area is divided from the instructions area, therefore allowing certain memory manipulation which standard PC configuration does not allow. Furthermore, ARM Cortex family of MCUs is widely used and nowadays presents a certain standard of general-purpose MCUs.

B. *Processors*

In this research, Raspberry PI3 is used. Raspberry PI3 is based on ARM Cortex A class of MCUs which are convenient for use in high-end technical solutions. Raspberry PI is a general-purpose development platform. Apart from standard MCU capabilities, Raspberry PI is convenient due to a number of GPIO (General Purpose Input/Output) pins as well as because of serial communication capabilities. These capabilities allow this platform to be efficiently used as RTU node in SCADA system. The PC-to-PC concept is based on the fact that expensive and sophisticated RTU which is usually based on PLC can be replaced with standard general-purpose PC configuration. This research and experiment are conducted on further decrease of RTU implementation price as well as increase of control capabilities. Additional control capabilities are ensured by mentioned general-purpose input/output (GPIO) and serial communication capabilities which additionally facilitate real-life control of an object on a remote site. First advantage of using Raspberry PI and other similar platforms, comparing to standard PLC based RTUs such as those standardized by IEC 61131-3 software architectures is the price. Total cost of implementation PLC based RTU is significantly higher both in initial cost as well as in maintenance cost. Furthermore, credit-card-sized PCs programming in standard languages such as C++ or Phyton allows variety of different solutions. However, main advantage of using Raspberry PI or other similar PC as RTU is the possibility of including elements of Artificial Intelligence on RTU side, more specifically Machine Learning. This advantage allows creating fully autonomous control system able to learn the controlled processes and propose control actions based on previous experience.

C. *Artificial intelligence on RTU side*

One of the main advantages of PC-to-PC SCADA systems is the possibility to include Artificial Intelligence, more precisely Machine Learning which was either not possible or at least extremely costly when using classical PLC based RTU nodes. For this research and proof of concept, elements of Machine Learning are implemented in the form of control application installed on Raspberry PI which is placed on RTU remote side. Measurements are not relevant for the experiment but rather used as example and proof of hypothesis that this concept can be implemented as well as proof of semi-intelligent functionalities.

The possibility of implementation of this concept also improves the general SCADA concept by delivering the possibility to have a fully controlled RTU side as well as a completely independent RTU side in cases of a communication break. During the communication disruption, RTU side is able to not only switch to automatic control of the real-life device, but also to continue with the Machine Learning algorithms in order to eventually learn and adjust to automatically optimize remote control of the device and thus avoid potential risks of major malfunctions or disastrous situations.

On the RTU side, a heater was installed as a controlling device, which will be turned on and off according to received measurements. All measurements, such as temperature and humidity, are saved on the RTU side. Apart from this, the individual performances of devices controlled from the RTU are recorded, with the ideal performance already defined. The proposed method is to set several time intervals and measure external influences such as temperature or humidity during those times. Based on these performance measurements, it will be able to adjust the heating device accordingly, in order to get ideal conditions during temperature and humidity fluctuations.

The entire model will be optimized through consistent updates of the neural network and the calculations of the algorithm which controls the heating device. In this way, external intervention from third parties is very significantly reduced, and the model becomes self-adjusting.

Another example where Machine learning can be utilized is in measuring how the presence of people in the room affect temperature or humidity. References were added to the article where a detailed description as well as proof of concept can be found, regarding the use of sensor data to detect the presence of unauthorized persons in the room, in order to prevent tampering or restrict access to the device. [4]

III. Experimental Model

A. *Hardware*

In order to prove that SCADA works on devices which are the size of a credit card, a solution for that purpose was created. For experimental purposes a credit-card-sized PC (Raspberry pi 3) on the RTU side was used [Fig.1.]. Sensors for temperature and humidity were added, as well as led indicator which will give a warning if the humidity is higher than predefined.

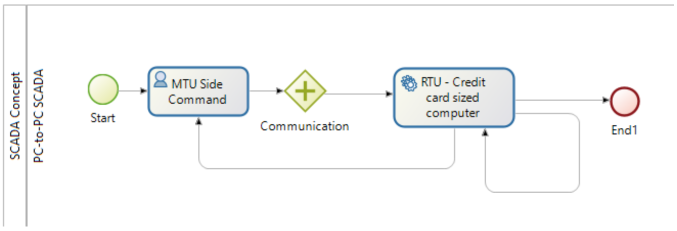

Fig. 1. Simplified workflow model illustrating the control loop

Furthermore, a relay connected to a higher power consumer was added, in order to have real time control and switch the specific device On or Off. For the MTU side a web application on the Amazon cloud was created and hosted [fig.2].

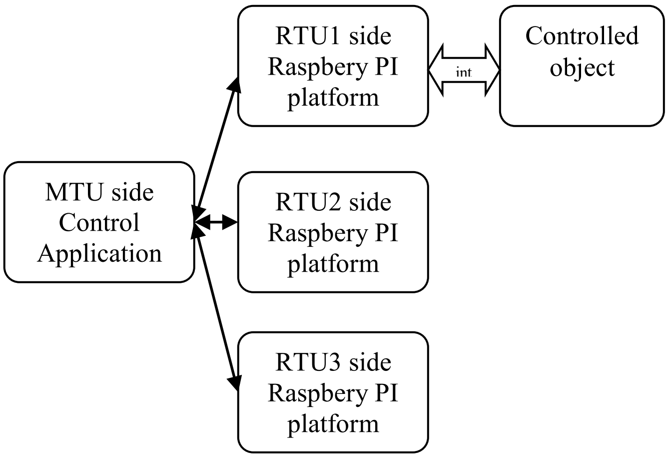

Fig. 2. Block diagram of the PC-to-PC SCADA concept

In order to demonstrate ease of configuration and experimentation on this concept, Figure 3 illustrates a simple configuration based on hobby-level equipment. Equipment used for this experiment is non-industrial. However, after experimental results show satisfactory output, implementation should be upgraded to industrial level equipment and should be based on industrial standards.

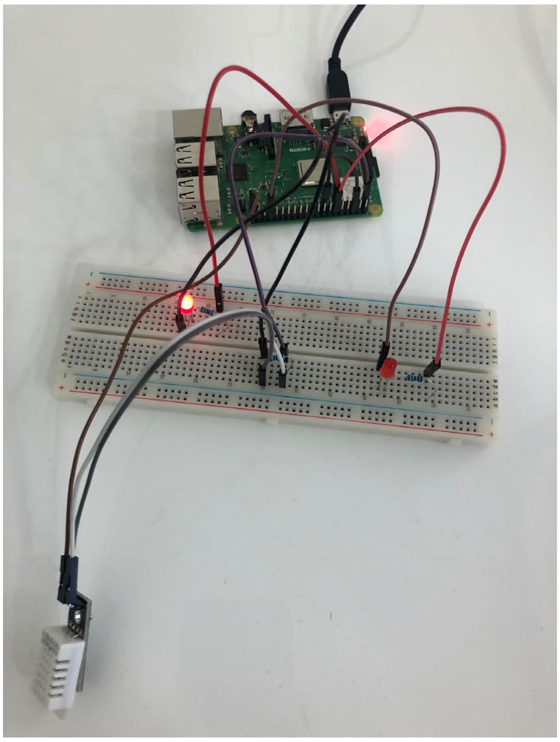

Fig. 3 Experimental model for proof of concept showing only one Raspberry PI controller and only one ambient temperature and humidity sensor

B. *Software*

The software solution was designed to ensure real time communication between the MTU and RTU sides. The communication between them is through the central server.

For the central server, instances of Linux Server were utilized. The server was configured in such a way as to run asp.net application. SignalR was set up on the central server to enable real time communication. SignalR provides API for creating remote procedure calls (RPC), and in this way establishes communication so that the procedures on the RTU side as well the client side are both executed from the central server. Communication can be established using http or https protocol, but in this manner the connection between the server and the client is persistent as opposed to a classic http connection which must be re-established each time, and for every connection.

C. *The Web application*

The application or the MTU was created using Bootstrap framework for the design and JavaScript library SignalR for real time communication with the server [Fig.4.]. Furthermore, the application has the ability to receive or send new data. If there is a change registered by the humidity sensor a notice will be sent to the central server (Hub server), where it will be received by the web application immediately. The received data will be handled by the application logic.

This Data can be further processed or used for executing some specific action or even to show statistics diagrams. In this case the data will be presented as a diagram which will show the ambient temperature and humidity in real time.

Furthermore, two new buttons were added to the application, that are used to turn the heating device on or off. It is important to mention that these two buttons take action priority in terms of execution above the automatically controlled on and off action.

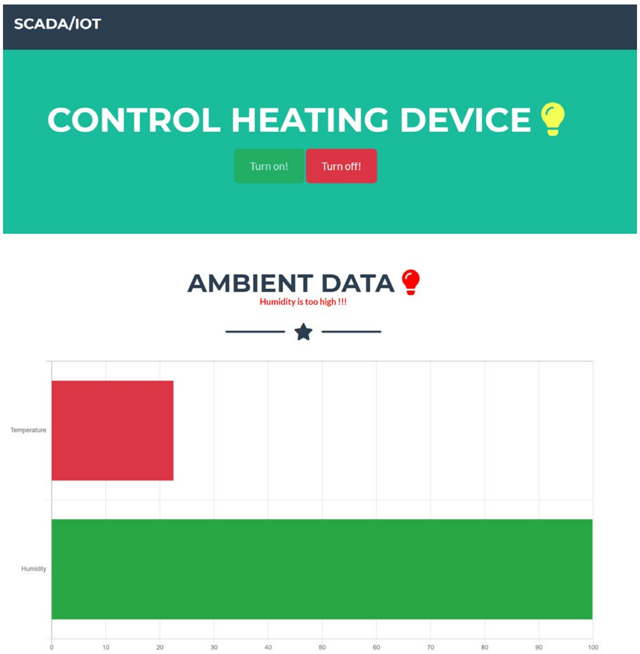

Fig. 4. Application interface of MTU node

D. *The RTU*

The RTU uses the Raspbian operating system with additional libraries. The application was written in ASP.Net Core and deployed to the Raspberry pi. It is designed to connect to the central server and to have an ability to send or receive data in real time. In this case the sensor data is immediately sent to the central server and forwarded to the MTU. The communication is also used for receiving incoming data, which is then used for manipulating the power switch. For demonstration purposes additional code was included to handle air humidity. If the humidity is higher than 90x, a notification is sent to both the MTU and RTU side. An LED indicator turns on at the RTU side and on the MTU side warning message appears.

The sensor data is saved as a text file on the RTU side and can be used to create various new algorithms, as well as implement Machine learning solutions, AI, etc. In the article, the implementation of a Machine Learning algorithm was not described, but the capabilities resulting from acquisition of such data was presented. Furthermore, articles which describe the implementation of Machine Learning utilizing such data were referenced.

Apart from the RTU side, the data is also capable of being stored on the MTU side, where it will be processed further if a need for that arises. Of course, the RTU side would be aware, in this case, of the MTU side, as well as able to request additional data in order to make a more informed decision, should it need to perform an action autonomously on the controlled devices.

This awareness of the RTU side opens up several capabilities, such as the exchange of data among RTU devices in the same environment. Given that the RTU devices are connected to a central server, the central server itself can send or forward information to any and all devices which are connected to the same central server. If we connect multiple RTUs with this implementation, we automatically receive an update. This option was demonstrated by connecting the RTU and Client app to the same central server, and both displayed the same data in real time, and according to any changes that we made.

As an example, several RTU devices can exchange information and thus make upgrades to their own Neural Network, thereby improving the algorithm which performs required actions. The best results noted on an RTU can then be shared with other RTUs, in order to be recalculated into the algorithm, to ensure better performance.

Adding a new RTU device would, in this case, be significantly more efficient because it would acquire the last and most optimal set of updated data, which would make its decision-making easier. The new RTU device would thus skip many previously discarded sets of data, and immediately begin on the last step, guaranteeing maximum efficiency. The following advantages can be gained from such RTU awareness:

(1) Better performance when making decisions
(2) Easier adding of new devices
(3) Decrease of loss during malfunction, due to RTU synchronization
(4) Availability of initial data and starting point for new RTUs
(5) Real-time data exchange
(6) Larger quantity of data

Every additional device can also be used as a PC which will lead to an increase in processing power of the entire system. This portion can be handled through delegating a portion of the task at hand to each of the RTUs, and then combining the results and sending them to other RTUs or the central hub.

It is important to note that auto-maintenance of the system must be active, and that should a manual action occur by the person controlling the MTU, this action would take priority over any action performed by the algorithm on the RTU side. This action would then be stored on the RTU side, and included in the data available to the algorithm. Furthermore, the MTU maintainer would receive real-time communication that their action is faulty, as well as a suggested solution, should a need arise. The entire process would occur in real time, ensuring full awareness of all parties.

IV. Proof of Concept

Given the complexity of SCADA systems, especially highly sophisticated PLC's, there is a common trend of attempting to lower SCADA costs on RTU side.

The convergence of the technologies has made low-cost automation a viable reality [5].

(7) Low-cost data loggers/controllers
(2) A growing variety of inexpensive sensors
(3) Expanding use of solar-energy systems
(4) Innovations in communication equipment
(5) Rapid advancements in the PC industry and
(6) The phenomenal growth of the Internet.

SCADA is a process control system that enables a site operator to monitor and control processes that are distributed among various remote sites. [6]

Experimental results showed that the concept is proofed, and that the system can be up-scaled to a real-life SCADA implementation. Since the average cost of PLCs, industrial scale PLCs and especially PLC systems intended for SCADA usage is significantly higher than standard general-purpose credit-card sized computers, it is easy to identify the first source of SCADA implementation cost decrease. Further source of decrease of implementation cost is based on the fact that the maintenance cost of credit-card sized PCs is significantly lower than the PLC maintenance. Also, the learning curve for developers and maintenance staff is much faster in case of credit-card sized PCs as well as power consumption, robustness, safety, AI capabilities, remote connection capabilities and many other elements.

Conclusion

The object of this research was to show how SCADA implementation costs can be further reduced by using credit-card sized PCs, as well as to shed light on other advantages of such a change. It was shown clearly that it is possible to install a credit-card sized PC on the RTU side; a Raspberry Pi in this case. The way in which communication between the MTU and the RTU can be established was also explained in detail. The use of modern technologies was presented, as well as ways to handle the resulting data and make it functional in terms of real-time execution. It was further demonstrated how various sensors could be connected to the system, increasing functionality exponentially. The sensor data was collected and stored on the RTU side, but the same data could be stored on the MTU side should a need arise for additional analyses.

The data stored on the RTU side was also used for purposes of optimization. Furthermore, the research demonstrates the ability to make use of this data to manipulate the connected devices in cases of loss of connection. This type of implementation on the RTU side was substantiated through use of Machine Learning. By executing an algorithm on the aforementioned data, functionality was demonstrated even in cases of total loss of connection. The number of interactions with the RTU side was also significantly reduced, through self-adjustment to the Neural Network by the RTU, which led to improvements to auto-maintenance capabilities of the manipulated devices.

Additionally, a way in which the data processing could be optimized was shown through simultaneous use of the processing power of several connected devices. This connection was also used to exchange data between the RTUs, so that all the RTUs could have synchronized results. In this way, any new RTU device would immediately be made aware of the last recorded state, and thus able to continue working without delay.

As it was demonstrated, the RTU side can also be used to warn a person operating the MTU if an action that they wanted to perform was not adequate, and to propose more optimized action to be performed. Additionally, the capability of the RTU side to autonomously request action from the MTU side was demonstrated, in order to achieve timely intervention. Apart from web notification, it is possible to implement a different type of notification such as email, SMS etc.

For further research, it would be possible to perform an analysis of how the various systems which are not interconnected or controlled by the same MTU could be connected and used together. A secure connection between the systems would need to be established, and an analysis of data usability across systems would need to be performed. Additionally, an analysis where the RTU side would have several sources of input publicly available for collection could be performed. This type of research would have as its object the determination of whether such an increase in data is relevant or simply increases the possibility of an error.

[1] E. Babovic and J. Velagić, "Lowering SCADA development and implementation costs using PtP concept," *2009 XXII International Symposium on Information, Communication and Automation Technologies*, Bosnia, 2009, pp. 1-7.

[2] Aung Naing Myint, Hla Soe, Theingi, and Win Khaing Moe (2008), Implementation of Control Unit using SCADA System for Filling System, World Academy of Science, Engineering and Technology Volume 36 December 2008 ISSN 2070-3740

[3] Randy Dennison (2007), „A Pre-SCADA System Assessment“, pp. 1

[4] Alexandra Moraru, Marko Pesko, Maria Porcius, Carolina Fortuna1 and Dunja Mladenic: Using Machine Learning on Sensor Data, ournal of Computing and Information Technology - CIT 18, 2010, 4, 341–347doi:10.2498/cit.1001913

[5] R. Hansen1, A. Hilton, B. Berger, W. Pullan, Z. Gao and C. M. Lee (2001), Low-Cost Automation and SCADA: A Pacific Rim Perspective, pp. 1

[6] V. Rakocevic, J. A. Meech, S. Kumar, I.V. Samarasekera And J.K. Brimacombe, Computational Intelligence in a Real-Time Scada System to Monitor And Control Continuous Casting of Steel Billets, University of British Columbia, Centre for Metallurgical Process Engineering, Vancouver, Canada (http://www.mining.ubc.ca 2009)

[7] Amiya Ranjan Panda, Mishra Debahuti, Implementation of SCADA / HMI System for Real-Time Controlling and Performance Monitoring of SDR based Flight Termination System, July 2016, 10.1016/j.jii.2016.07.001

[8] Kyosuke Yamamoto; Takashi Togami; Norio Yamaguchi; Seishi Ninomiya: Machine Learning-based Calibration of Low-cost Air Temperature Sensors Using Environmental Data, Sensors (Basel). 2017 Jun 5;17(6). pii: E1290. doi: 10.3390/s17061290