International Conference Cybernetics & Informatics 2022
Visegrád, Hungary, September 11th - September 14th, 2022

Remote Data Acquisition Using Cloud Database

Filip Žemla and Ján Cigánek
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Bratislava, Slovakia
filip.zemla@stuba.sk, jan.ciganek@stuba.sk

***Abstract*— The presented paper deals with the design and implementation of a complex SCADA system for the control of a production line model and for the remote data acquisition using cloud database. The control of the educational model of the production line is based on a PLC device using the visualization application of the production process in real time and recording of process data in a cloud database. The main components of SCADA systems used in control processes, advanced software for creating SCADA applications and basic information about cloud databases are also defined. It is described the design and implementation of the control program in the TIA Portal environment and the design of the visualization application in the Promotic environment. The form of a cloud solution based on the Microsoft Azure platform for the needs of process data storage is presented.**

Keywords— SCADA system; HMI; Visualization system; PLC; Production line; Cloud database; Data acquisition;

I. INTRODUCTION

Digitization with automation is significantly changing our world and the world of industry. The volume of data in modern industrial plants is constantly growing, which brings new challenges for SCADA system solutions. In the long term, it is necessary to manage and archive huge volumes of data that are demanding on the capacity of storage devices. The task of modern SCADA systems is to monitor and control production processes as reliably, efficiently and productively as possible, while it is necessary to constantly optimize them. The means of automation have expanded from large automated lines to smaller ones, to individual machines, buildings or even smart homes [1].

The possibility of checking and obtaining data in a more accessible way, mutual communication between devices and their control or editing over local networks or via the Internet is increasingly required. These devices enable the remote collection of data into databases, based on which it is possible to evaluate both the devices themselves and the units to which they are connected [2, 9].

The advantage of the automation of production processes is almost continuous operation with minimal human intervention, which mainly takes care of inspection or service. It also enables production in life-threatening environments, such as explosive environments or chemical operations, where it is possible to limit the occurrence of threatening situations and close production to separate rooms or cells without the presence of people [3].

The main goal of this article is the design and implementation of a complex SCADA system for managing a reduced production line model and subsequent remote collection of process data using cloud database. This model will be controlled by a PLC device that turns on and off individual output devices based on the designed logic. The course of the automated production process is displayed in a visualization application on the main computer station, from which the line can also be controlled manually. In the event of a malfunction or error on any of the devices, an alarm is issued to notify the user of the situation. Acquired process data on the number of manufactured pieces, machine hours and generated alarms will be securely stored in a database created in the cloud space. The transition from data collection on physical database servers to cloud databases offers many advantages, but also difficult challenges. It is important to constantly optimize and modernize application solutions so that they keep up with developing technologies and, above all, to bring the highest possible efficiency of investments.

II. SCADA SYSTEMS

SCADA (supervisory control and data acquisition) systems are used for real-time monitoring and remote control of processes in various industries. It derives information, transfers it to the system, displays it on the computer screen and alerts the supervising operator. It also performs the necessary analysis and checks, for example by determining the severity of the fault, and displays this information in a logical and organized manner.

SCADA systems can be relatively simple, such as one that monitors environmental conditions in a small office building, or incredibly complex, such as a system that monitors all activity in a nuclear power plant or municipal water supply.

A SCADA system allows an operator at a location central to a widely distributed process, such as an oil, gas, pipeline, or irrigation system, to make setpoint changes to remote process control units. This means opening or closing valves, operating switches, monitoring alarms and collecting required data. The benefits of SCADA systems are best appreciated when processes are very large - hundreds or even thousands of kilometers from one end to the other. In this way, operating costs are reduced, as routine checks are not required to monitor

978-1-6654-8775-7/22/$31.00 ©2022 IEEE

the operation of the equipment. The value of these benefits increases even more if the facilities are very far away and require extra effort to visit them [1, 2].

A. Basic parts and functions of SCADA

A SCADA system consists of several remote terminal units (RTU, PLC) that collect process data and send this data back to the master station via a communication system. The master station displays the acquired data and also allows the operator to perform supervisory control. Key features of SCADA software include user interfaces, graphic displays, alarms, trends, RTU (and PLC) interface, scalability and data access [3].

In a more complex SCADA system (Fig. 1), there are essentially five levels or hierarchies:

1. industrial instrumentation and control devices,
2. controllers and terminals (PLC, RTU),
3. communication system (infrastructure),
4. main station (HMI),
5. database system.

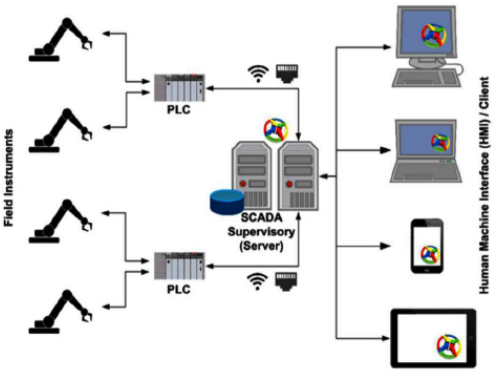


Fig. 1. Basic architecture of the SCADA system [12]

In general, SCADA systems can be described as a combination of telemetry and data acquisition techniques. SCADA collects data through a remote terminal unit (RTU/PLC) and transmits it back to the central workplace. It also performs the necessary analysis and control, and then displays this information on several screens or operator displays. The required control activities, i.e. action interventions either automatically or by the operator, are then transferred back to the process.

Accurate and timely data enable process and equipment operation to be optimized. The advantages include more efficient, more reliable and, above all, safer operation [3].

B. PLC devices

A programmable logic controller (PLC) is a special form of microprocessor controller that uses programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic for machine and process control purposes. It is designed to be operated even by engineers with limited knowledge of computers and programming languages. PLC designers have pre-programmed it so that the control program can be entered using a simpler and more intuitive form of language. The term logic is used because programming primarily concerns the implementation of logical and switching operations. Input devices (sensors, switches) and output devices (motors, valves, etc.) are connected to the PLC in the controlled system. The programmer enters a sequence of instructions (program) into the memory of the PLC, which then monitors the inputs and outputs accordingly and sends the control signals for which it was programmed.

PLCs have the great advantage that the same basic controller can be used in a wide range of control systems without the need for new wiring. All that is needed to change the control system and its rules is to upload a new program (instructions) to the device. The result is a flexible and cost-effective system that can be used with control systems that vary widely in nature and complexity.

The input-output interface used in a PLC can take two forms: compact and modular. The fixed type is connected to a small or micro PLC system, where all functions are integrated into one unit. The number of I/O ports is fixed within each such model and cannot be changed. Modular types use an expandable panel to hold I/O modules, allowing the number and type of modules to be changed. Input interfaces receive signals from machines or process equipment and convert them into signals that the control unit can work with. Output interfaces are used to control actuators that control some process [6, 7].

Programmable logic machines were designed so that the developer does not require a high level of knowledge of programming languages.

The entire program is executed in one cycle. It can be interrupted and then returned to continue or restarted. The IEC 61131 standard for PLC standardization, standardizes and defines the methods used for PLC programming. The most frequently used PLC programming method is based on the use of ladder diagrams (Ladder Diagram - LD), also known as the language of contact (relay) diagrams. So writing a program is equivalent to drawing switching circuits. A ladder diagram consists of two vertical lines representing electrical wires. The circuits are connected by horizontal lines, which represent quasi rungs of the ladder between the two verticals.

Another commonly used standardized programming language is the Function Block Diagram (FBD), which is used for PLC programs described in terms of graphic blocks. It is a graphical language for displaying signal and data flow through blocks, which are reusable software elements. The function block is shown as a rectangular block with inputs entering from the left and outputs exiting from the right.

A programming method that can be considered as inputting a ladder diagram using text is the instruction list (IL). Programming is done using a series of instructions, with each

instruction on a new line. Each instruction consists of an operator followed by one or more operands.

The method of programming using a sequential function chart (sequential function chart - SFC) is used to visually represent the operation of the system and display the sequence of events involved in its operation. An operation is described by a number of separate sequentially connected states or steps, which are represented by rectangular boxes, each of which represents a specific state of the controlled system.

The last standardized type is the method of programming using structured text (Structured text - ST), when programs are written as a series of commands separated by semicolons. Predefined commands and subroutines are used to change variables that are defined by values, internally stored values, or inputs and outputs [6].

C. Communication system

A communications network is a collection of computer hardware and software, along with cabling, networking devices such as switches, routers, and the like, that allow computers to communicate with each other. A network provides a means of sharing information between industrial computers, PLCs, or input/output devices.

A network consists of two or more devices connected via a shared media to share files or peripherals such as modems, printers and others. It serves to connect the main central station with control units (RTU, PLC), HMI and can use industry standard or manufacturer-protected proprietary protocols. Both the RTU and PLC work autonomously in real time to control the process using the last command provided from the control system. A failure of the communication network does not necessarily mean that the process control of the device stops, but when communication is restored, the operator can continue to monitor and control [4].

Network topology is the first point to consider when designing how devices in a SCADA system are to be interconnected. Topology refers to the pattern of interconnection of various elements of computers. There are two types of network topologies, physical and logical.

Physical topology refers to the layout of network cables, their connection to computers or other network devices. A logical topology defines how data is transferred from one device to another regardless of the network hardware link. In general, the logical topology is defined by the network protocol, while the physical topology is mainly defined by the physical arrangement of network cables and devices [9].

D. Human-machine interface

The master station (MTU) is essentially a supervisory computer system. This station serves as a central processor and has two main functions that are the core of the SCADA system. The first is the regular acquisition of process data from the RTU (PLC) and secondary stations, and the second is the control of remote devices through the operator's station. MTU is a device that issues commands to RTUs located at remote locations from the control element, collects required data, processes, stores and displays information in the form of images, curves, tables for operators and thus helps to make control decisions. Communication between MTU and RTU is two-way, however, the main difference is that the RTU cannot initiate the conversation. Communication between the MTU and the RTU is initiated by programs within the MTU unit that are run either by operator instruction or automatically. In 99% of cases, instructions and messages for the RTU sent from the MTU are triggered automatically. When the master station (MTU) requests the necessary information, the RTU sends it. This principle of communication is referred to as master-slave. After receiving the required data, the MTU communicates with other devices and HMI screens for operators with which it is connected via a LAN/WAN network [4].

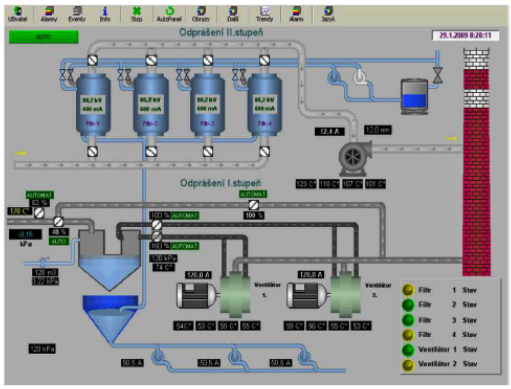


Fig. 2. Example of graphic display for controlled process [8]

HMI or Human Machine Interface is a device that allows you to visualize and work with collected process data. In most cases, this is done through a graphical user interface (GUI), such as a computer monitor, which serves the supervisory system operator and thus creates an interface between the operator and the SCADA software. This human-machine interface provides process information to operating personnel graphically in the form of diagrams and trends, which are a schematic representation of the controlled process. It is also used to display important recorded events and alarms. HMI panels are connected to the main SCADA computer, which provides real-time data for creating live diagrams, alarm displays and trend graphs.

Displayed diagrams consist of line graphics and schematic symbols that represent process elements, or they may consist of digital photographs of process equipment accompanied by animated symbols.

When controlling the operation of equipment through HMI, operators can enter commands using a mouse, keyboard or touch screen. For example, an animated tank symbol can show when the tank is full, and a colored valve symbol can show whether liquid is flowing into the tank. The operator can control the valve by clicking the mouse or touching the screen. The HMI will then display the change in liquid level in the tank in real time (Fig. 2).

In smaller SCADA systems, the master station may consist of a single computer - in which case the HMI is part of that computer. In larger SCADA systems, a master station may contain several HMIs hosted on client computers, multiple data acquisition servers, distributed software applications, and disaster recovery sites [4, 5].

E. Database system

The database system consists of two parts that can be operated by the database application or the end user:

- database,
- database management system.

The database management system handles all user requests for access to the database - it can be adding a new file to the database, deleting it or working with the data itself (creating, reading, updating or deleting data). Among the most used commercial products are Microsoft Access, Microsoft SQL Server, MySQL or Oracle Database.

A database is a shared integrated computer structure that includes end-user data and metadata, i.e. data about the structure of the database.

User data is organized into tables that represent a set of data describing the data model of the database. Table columns define individual properties (attributes) of a given data type, i.e. data type and name. The most common data types are text (alphanumeric), numeric (numeric), logical (TRUE/FALSE), and date (dates). Rows (records) of tables represent entries in the database, and each entry contains data from all defined columns in a precisely given structure.

A set of tables, each of which represents a certain type of data, can be called a relational database. Relational databases make it possible to capture not only data about researched objects, but also their mutual relationships, which makes them closer to the real world. Each record of the table is uniquely identifiable by a primary key, and the order of the records does not matter. A primary key is a property (attribute) or combination of properties that uniquely identifies a record, i.e. this column must be different in each record [10].

III. Microsoft Azure

Azure is a professional cloud-based platform provided by Microsoft. Thanks to the advanced functionality, it is possible to operate virtual machines, manage SQL databases, create backup copies of resources and many other actions without worrying about failure or out-of-date equipment and software. All operating systems and a wide variety of programming languages are supported.

MS Azure provides access to integrated services, including computing power, storage options, applications and data, thanks to which it is possible to quickly manage the infrastructure, implement a larger number of tasks and, above all, achieve significant financial savings. MS Azure is a combination of IaaS (Infrastructure as a Service) and PaaS (Platform as a Service) and was designed for access to all services, computers and applications without interrupting availability. The Azure system provides an SLA contract guaranteeing operation at the level of 99.95%, continuous monitoring of the services used and 24-hour permanent support. Azure offers access to services from anywhere in the world. In addition, this system is based on a worldwide network of data centers that is constantly expanding [11].

IV. Case Study

The equipment used in this article is a basic Siemens S7-1200 series PLC equipment that controls a laboratory model of a production line. We monitor the course of the production process in a visualization running on a personal computer connected to the PLC.

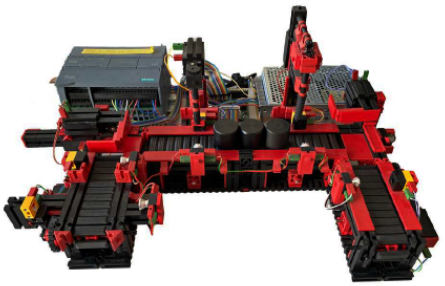


Fig. 3. Laboratory model of the production line

We created the control program in the TIA Portal tool due to compatibility with the used PLC device. Programming consisted of ladder diagram programming. In addition to control, we also ensured diagnostics of the line in the form of alarms, which were triggered when individual motors were started for a long time and stopped the entire production.

Another task was the design and creation of a visualization application. For this purpose, we chose SCADA software Promotic because of its simple, intuitive operation, high scalability, reliability and previous experience with this software.

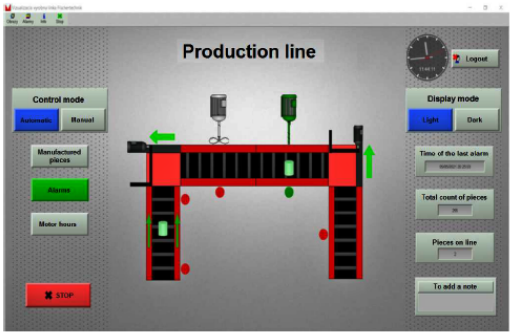


Fig. 4. The main visualization panel

Our SCADA application consists of two main screens – first one with automatic mode and the other with manual mode. From these main screens, it is further possible to switch to modal pop-up screens that display alarms, the number of manufactured pieces and an overview of the machine hours.

The application also works with a local database, in which generated alarms, engine hours and the total number of manufactured pieces are recorded. After launching the application, the first thing that appears is the login screen, which prompts the user to enter a login name and password. After entering the login information, the "Main Panel" will appear.

The main home screen of our application consists of a graphic visualization of the production line, which runs in fully automatic mode. On the graphic shown, we can start the production process and monitor the current position of the workpieces on the line, i.e. specifically at which of the optical sensors the workpiece is located. In case of movement of any belts or sliders, this movement is represented by green arrows next to the given object.

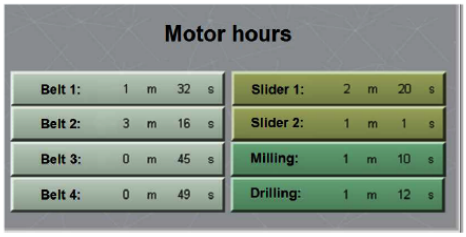


Fig. 5. Modal watch window

By pressing the "Motor hours" button, we open a new modal pop-up window, which contains an overview of the output devices and their total operating time since the PLC was turned on. Activity time is displayed in minutes and seconds for the needs of our project. In real operation, this information can be used for device diagnostics and maintenance planning.

The "Manufactured pieces" button opens a modal window displaying a table with production statistics at intervals of hour, day, week and month. In real operation, this information is essential for the higher corporate layer and production planning.

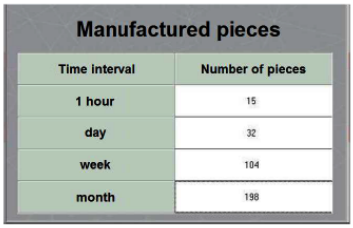


Fig. 6. Modal window of manufactured pieces

As a cloud service, we chose Microsoft Azure, specifically the SQL Database, which is used to create and manage databases. Subsequently, we chose to create a new database. Here we chose to simultaneously create a new SQL server, for which we defined the name, data center location, capacity and administrator account. There are also other setup configurations to choose from, such as security features, restoring a database backup, or adding tags. Azure offers an integrated SQL editor for managing databases and executing queries, but for better clarity we will use a different tool. We will access our cloud database using Microsoft SQL Server Management Studio (SSMS), which is clearly the most robust SQL database server management tool. It is a fully compatible and verified database management tool, but similar editors such as HeidiSQL or DBeaver can also be used.

In the next step, it was necessary to create tables in our database, in which we will write individual process data. Similar to the local database, here too we have created tables for the number of manufactured pieces, motorcycle clocks and alarms. To each table we assigned the appropriate attributes with the corresponding data type.

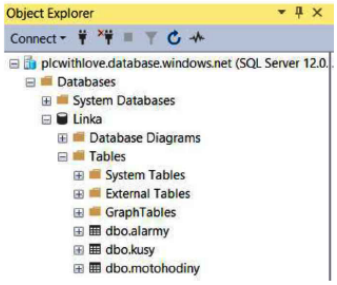


Fig. 7. Database tables

We will use the dataFEED OPC Suite tool to connect the PLC to the database and then record the process data. The configuration concept of this tool is based on so-called data sources and data targets. The source is PLC variables (tags) storing process data (engine hours, number of manufactured pieces, alarms) and their target is individual database tables.

In data destinations, there are various services to choose from such as MQTT Broker, Siemens MindSphere, OPC clients and data files to which data can be sent or written. We chose SQL Database as the data target, since we will send the data to the SQL database. First of all, we had to set up the ODBC driver, with which the communication will take place. ODBC (Open Database Connectivity) is a standardized, open interface for accessing various database management systems. Through ODBC drivers, applications can directly issue instructions to databases or execute queries. Here we have set up our SQL server address, login information and the specific database we will be connecting to. Successful execution of the database connection test completes the process. In the end, we just wrote the process data into the connected database at certain time intervals or when the value of the variable

changed. We chose a new database action, where we had the choice of direct preconfigured writing with the INSERT command, connecting a stored procedure or executing a custom SQL command. In the case of recording the number of pieces and engine hours, we used the user-friendly option of direct input, in the case of recording alarms, we executed our own SQL command.

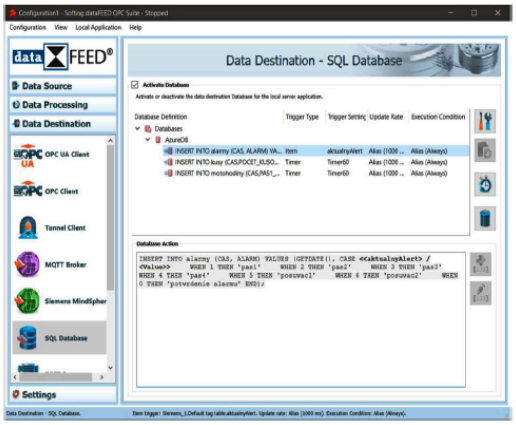


Fig. 8. Configuring the SQL database as a data target

V. Conclusion

The goal of the work was to design and implement a complex SCADA system for managing a reduced model of the production line and subsequent remote collection of process data using cloud database. The architecture of this system is composed of technological, application and cloud layers. To create it, we used available hardware, network and software components found in the school laboratory, which are used for educational purposes. The technological layer of the system consists of a PLC machine and a controlled model of the production line, which can work in automatic mode based on the designed control program. The entire production process is monitored through visualization on the main SCADA station, which fulfills the control and diagnostic function and thus forms the application layer. Acquired process data on alarms, manufactured pieces and equipment hours are stored in a cloud database on a virtual server.

In addition to the required properties, the visualization also contains additional functions such as a login screen, adding notes or changing the color display mode. The implemented complex SCADA system can be used in the educational process in the topics of visualization systems, PLC programming and storage of process data either in the cloud or locally. A more complex solution to the current system design would be to completely transfer the system to the cloud and remotely connect to the control network.

Acknowledgment

This paper was supported by the Slovak Grant Agency VEGA 1/0107/22 and KEGA 016STU-4/2020, 039STU-4/2021, by the Scientific Grant APVV-17-0190 and project No. ZDPVCC.

References

[1] D. Bailey and E. Wright, “Practical SCADA for Industry”, Newnes, 2003. ISBN 9780080473901. Available: https://doi.org/10.1016/B978-0-7506-5805-8.X5000-4.

[2] B. R. Mehta and Y. Reddy, “Industrial Process Automation Systems”, Butterworth-Heinemann, 2014. ISBN 9780128010983. Available: https://doi.org/10.1016/C2013-0-18954-4.

[3] G. Clarke and D. Reynders, “Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems”, Newnes, 2004. ISBN 9780080480244. Available: https://doi.org/10.1016/B978-0-7506-5799-0.X5015-3.

[4] “Master Terminal Units (MTU) in SCADA Systems”, In: ElectricalVoice. Available: https://electricalvoice.com/master-terminal-units-mtu-in-scada-systems/.

[5] A. Gerer, “ HMI – new knowledge and the best skills for the design of operator’s interface. ATP Journal [online]. Available: www.atpjoumal.sk/rubriky/prehladove-clanky/hmi-nove-poznatky-a-najlepsie-skusenosti-pri-tvorbe-operatorskeho-rozhrania-2.

[6] W. Bolton, “Programmable Logic Controllers”, 6th ed. Newnes, 2015. ISBN 9780081003534. Available: https://doi.org/10.1016/C2014-0-03884-1.

[7] E. Alphonsus and M. Abdullah, “A review on the applications of programmable logic controllers (PLCs)”, Renewable and Sustainable Energy Reviews, ISSN 1364-0321. Available: https://doi.org/10.1016/j.rser.2016.01.025.

[8] “What is PROMOTIC”, In: Promotic. Available: https://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsPromotic.htm.

[9] S. McCrady, “Designing SCADA Application Software”, Elsevier, 2013. ISBN 9780124170353. Available: https://doi.org/10.1016/C2012-0-06034-6.

[10] D. Kroenke and D. Auer, “Database”, Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

[11] “What is package of Microsoft Azure?”, In: Senetic. Available: https://www.senetic.sk/azure/.

[12] “What is SCADA?”, In: ECAVA INTEGRAXOR. Available: https://www.integraxor.com/what-is-scada/.

Authors



Ing. Filip Žemla
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
filip.zemla@icloud.com
Currently a student of doctoral studies at Slovak University of Technology in Bratislava. The main focus of his studies is oriented to virtualization and optimalisation of modern manufacture processes. His main skills are SCADA systems, database systems and front-end programming.



Ing. Ján Cigánek, PhD.
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
jan.ciganek@stuba.sk
He was born in 1981 in Malacky, Slovakia. He received the diploma and PhD. degree in Automatic Control from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology (FEI STU) in Bratislava, in 2005 and 2010, respectively. He is now Assistant Professor at Institute of Automotive Mechatronics FEI STU in Bratislava. His research interests include optimization, robust control design, computational tools, SCADA systems, big data, and hybrid systems.