Innovative Educational Technologies, Tools and Methods for E-learning
Scientific Editor Eugenia Smyrnova-Trybulska
"E-learning", 12, Katowice–Cieszyn 2020, pp. 268–278
DOI: 10.34916/el.2020.12.23

FLEXIBLE SYSTEM OF REMOTE APPLICATIONS AND TEACHING WITH SENSORS

Miroslav Kamenský[1], Eva Králiková, Mikuláš Bittera, Jozefa Červeňová, Karol Kováč
Institute of Electrical Engineering, Faculty of Electrical Engineering and Information Technology, STU in Bratislava
Ilkovičova 3, 812 19 Bratislava, Slovakia
[1]miroslav.kamensky@stuba.sk

Abstract: *Methods of remote teaching and e-learning have become necessity in recent times. The paper describes modular system of real workplaces accessible remotely. The system was designed in LabVIEW environment offering fast implementation of new modules even for inexperienced programmer. On the other hand, its structure is open for adding parts and modifications. The paper explains method of achieving such modular structure and flexibility for implementing of new workplaces. Sensor applications were added as needed for the education process at our institute. Web Service tool included with LabVIEW and possibility of application interfacing with html and JavaScript code is presented. This allows presentation of data on remote computer without the need to install LabVIEW Runtime Engine.*

Keywords: LabVIEW, DataSocket, remote experiments, measurement, Web Services.

INTRODUCTION

Nowadays, many common measuring or laboratory devices have a digital interface for communication with PC via internet and remote access tools. The measurement and control technology as a whole becomes more complex. The remote access workplaces are becoming increasingly important. The progress in web technologies influences development of software aids oriented for measuring applications. One of most typical measurement software certainly is LabVIEW (Laboratory Virtual Instrument Engineering Workbench). This approach for the implementation of laboratory measurements has the advantage of flexibility, and the possibility for remote measurement and control over a local network or internet. In the paper we present a software system designed in LabVIEW which allows overcoming some weaknesses of a common remote access system. It has a modular structure which improves soft-

ware security of workplaces. The system is based on communication between the DO module, controlling a workplace, and the GO module, distributing data and access. Several workplaces were designed, as described in Kamenský et al. (2018), and used for distance learning or for demonstration. We continue building other new applications. Two of them are presented in this paper with other improvements including the use of Web Service.

1. MODULAR AND OPEN CONCEPT

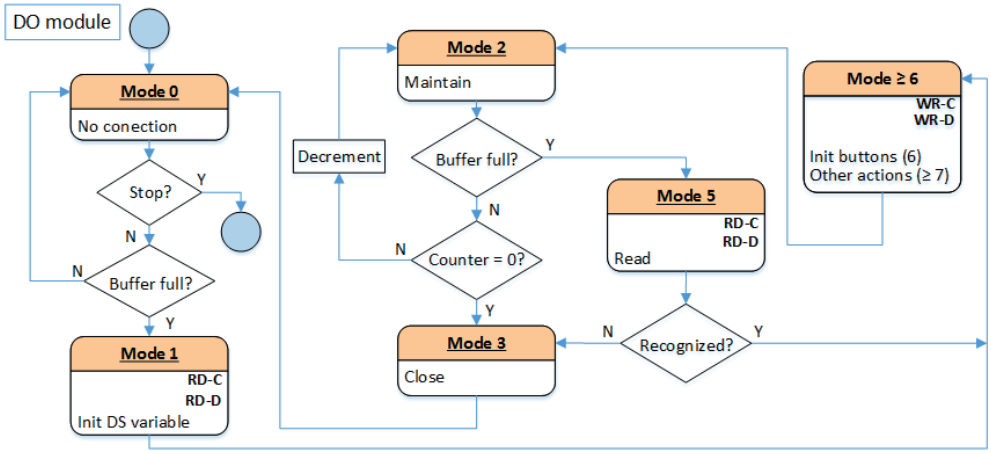
There have been many attempts to create applications with remote accessed at our institute e.g. based on C#, Java, JavaScript. We gradually realized two main problems preventing their usage for education. The first was software or rather network security issue when network administrator was unwilling to make the target PC available externally. The second obstacle was related to staff turnover in the workplace when even minor software modifications essential for adapting to actual pedagogical process were time consuming or impossible for a teacher which was not designer of the original application. Finally, a few years ago we started the creation of a new system with the aim to overcome those problems. We decided to use the graphical programming language G of the LabVIEW environment and chose modular and open concept allowing the design of parts of the system in other languages too. Hence our intention was also to create system with the prospect of a rising designer/user community. The LabVIEW environment and related tools and libraries belong to standard equipment at academic and industrial workplaces dealing with distributed or automated measurement. It offers high efficiency for the development of small applications interacting with laboratory equipment especially for non IT oriented community. It can simplify the task of measurement control, automation and distribution. It is still possible to design large applications and add modern features such as Web Services. Plenty of references can be found in the academic community referring to remote access established using LabVIEW, e.g. in García-Guzmán et al (2017) or Singh et al. (2015). A general dedicated web publishing tool is in development and there is no steady solution. Designers are naturally looking for tools intended for the same environment which they use for the type of work – such as Matlab2Web for Matlab in Gula and Žáková (2017). Our department is oriented for instrumentation and measurement and the LabVIEW is an essential tool of our everyday work. Encouraged by the progress towards internet distribution of data and control, we decided to develop our new educational and training system in LabVIEW.

Our system is based on a pair of modules: GO and DO. To access the remote workspace or a target application called DO module the user communicates over a superior GO module. For every DO and GO module pair the GO module is identical. It means that with the same GO module the addressed target application can be chosen by ID. If several workplaces should be accessed in parallel, then administrator leaves running more GO modules on server computer. The user actually accesses remotely one of those GO modules and the connection with the target PC is managed inside the GO module application. Its purpose is to accept commands from a remote user and forward them to a target application or to distribute results for the remote user. As we

divided the part controlling measurement from the module providing web publishing, once the GO module will be updated, it can be employed for all DO modules without any code change.

The DO modules are unique, each carrying its own identifier (ID). The target application could be located on any computer within the local network, which stays not directly visible from outside and hence protected against software attacks.

Software module works as a state machine and can be simply presented by a block diagram. In Figure 1, a block diagram of a generalized DO module is shown. A state is called mode here. The modes 1 till 5 are designed for managing communication with GO module. The application starts in mode 0 waiting for initial command coming into opened DataSocket buffer. In mode 1, the application is being prepared for cooperation with GO module. DataSocket variables are initiated – we use separate variables for commands and data and for every direction. In this phase user button labels has to be initiated. Therefore the "*Button Init*" command is sent to GO module followed by jump to mode 6 where data variable with labels is generated. Mode 2 is then dedicated to maintaining the communication by checking if a new command was received or time expired, which means the connection with GO module was interrupted. Note that every data exchange here is initiated by GO module which also has to send "*Refresh*" command within time out for the case when there is no action from a remote user. Otherwise the connection and the variables are closed in mode 3. After regular command was received in mode 2 it is subsequently recognized in mode 5.



F i g u r e 1. A block diagram of the DO module

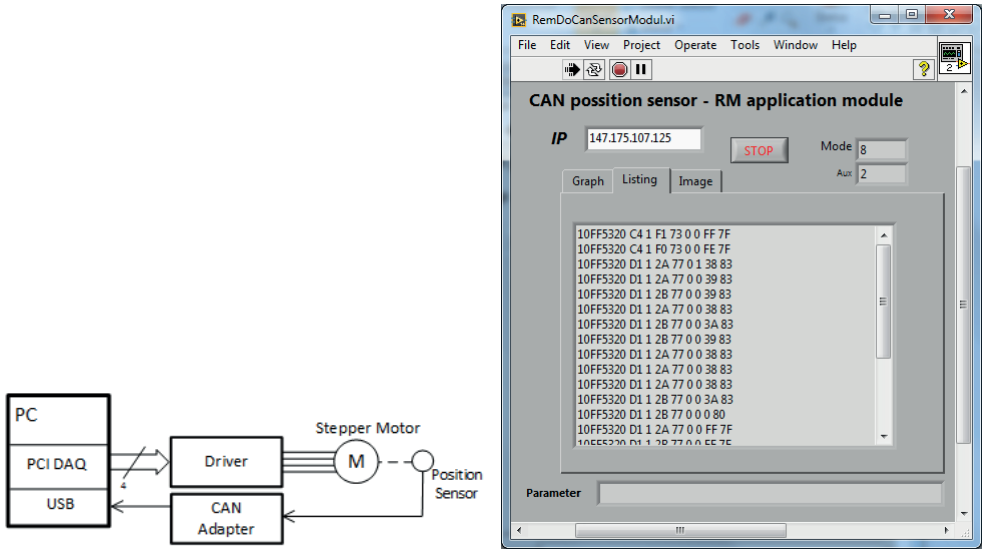
S o u r c e: Own work.

According to the command value, the application jumps from mode 5 to one of execution modes starting from mode 6. It is generally possible to jump again into mode 6 and resend labels. In our case we do not change labels after initialization and we jump to modes 7–12 during reactions to user actions. It is not strictly necessary to follow this DO module structure until the communication rules like command and data syntax or appropriate replies etc. are met.

For the GO module the block diagram is not very different (Červeňová et al, 2016). The blocks which have equivalent counterparts on the GO module side are underlined in Figure 1. Beyond that, also mode 4 is implemented in GO module as waiting for a reply and upper modes for processing replays and distribution data.

2. IMPLEMENTATION OF A NEW MODULE WITH SENSOR

Many types of workplaces can find application in education process. They are subjects of changes according actual needs. For the purpose of teaching diagnostics we were preparing new workplace corresponding to block scheme depicted on the left-hand side of Figure 2. The main part here is the rotating position sensor RT8CN with the CAN output. The exercise is aimed as an introduction to data distribution via the CAN bus (see CAN data in Figure 2) to get students prepared for lessons oriented on systems used in cars. For the remote control of the position a stepper motor has been added. A local PC sends sequence of pulses to motor driver and collects new sensor data from CANSUB adapter. A new LabVIEW DO module has been designed which distributes control and data to the remote user (via GO module) with functionality predefined by the teacher in the role of a designer.

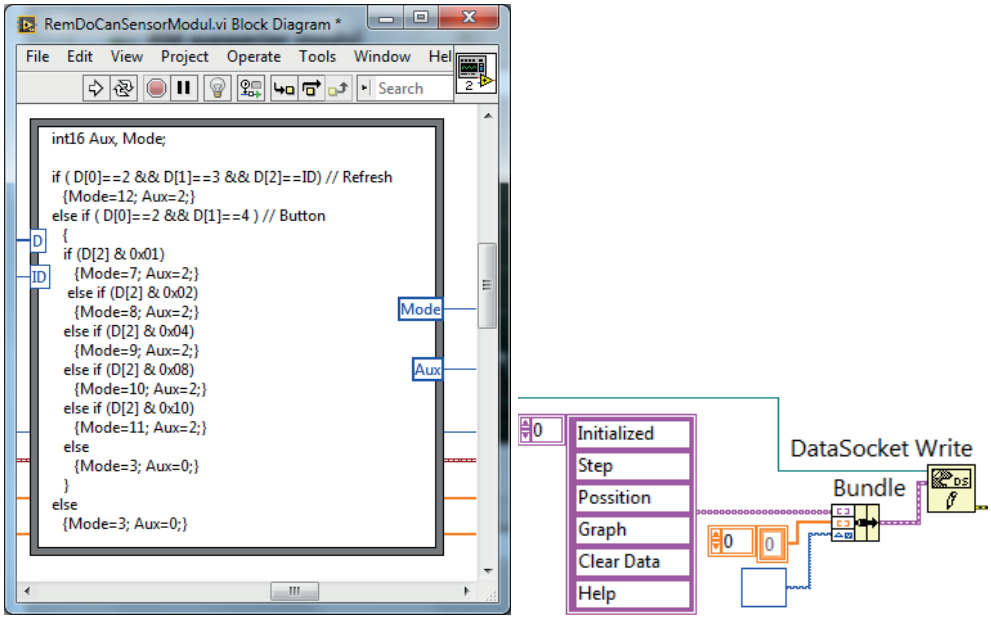


F i g u r e 2. Ablock scheme of the rotary position sensor (left) and its CAN data, as shown in the Listing tab of the DO module (right)

S o u r c e: Own work.

The target application is based on case structure where every mode from Figure 1 corresponds to one case. During the design of a new workplace we simply use an older DO module application as a template and modify it. The modes 1–5 operating general connection with GO module remains almost unchanged. Just the formula note in mode 5 has to be modified according functionalities of upper modes. On the left-hand side of Figure 3, the node already adapted for workplace with rotary sensor is

depicted. It implements conditional branch which reads command located in input array *D*[] and decides about mode where the reaction of the DO module will take place. Jump to modes 7–11 correspond to button pressed on the GO module side and mode 12 handles a timeout command. Node 6 stays apart of formula node as it is used only during initial process when button labels are defined. On the right-hand side of Figure 3 the preparation of DataSocket data variable containing button labels is shown. DataSocket variable is a cluster comprising three different types: string array, double array; image. During the initialization of labels only a string array is filled where the zero items is always a text to be shown in *Parameter* text indicator and items number 1–5 are desired button labels. The button marked as *Step* moves the stepper motor in one cycle of steps, *Position* reads new set of CAN data from the rotary sensor into the *Listing* tab, the Graph refreshes time trend of positions in th e*Graph* tab, *Clear Data* erases memory of positions and *Help* shows help in the *Image* tab. In Figure 2, the *Listing* tab of the DO module with CAN data has been shown; (uttons will be available on the GO module side.



F i g u r e 3. A block diagram of parts of the DO module: The Formula node used in mode 5 for processing the received command (left) and the initialization of labels via the DataSocket variable of the cluster type (right)

S o u r c e: Own work.

2.1. Sensor data available over Web Service

Web publishing does not automatically mean that data will be available on all devices with a web browser. The LabVIEW environment offers handy web publishing tool – Remote Panel – which assumes LabVIEW Runtime Engine running in the background of remote PC. The engine is not compatible with systems of small mo-

bile devices. Fortunately, LabVIEW supports Web Services representing technology compatible with the environments of mobile phones. Request for modification of data distribution capabilities in our system and for Web Services arose during last coronavirus-affected semester.

The students attending the course Microcomputer technique were not allowed to come into laboratory and stayed working from home. Firstly, we had to adapt laboratories to a new situation. Some partial tasks were performed by simulation using an online tool Tinkercad. Later students got alternative job to an original Arduino project – the old project involved temperature/humidity sensor. As they usually own Arduino Uno without a peripheral board, they had to design own capacitive position sensor built from two metal plates sliding towards each other. Such a capacitive sensor should be connected in a serial with a known resistor and they had the task of designing the Arduino firmware generating pulses to RC circuit and measuring deceleration of the signal edge at the capacitor as a delay.

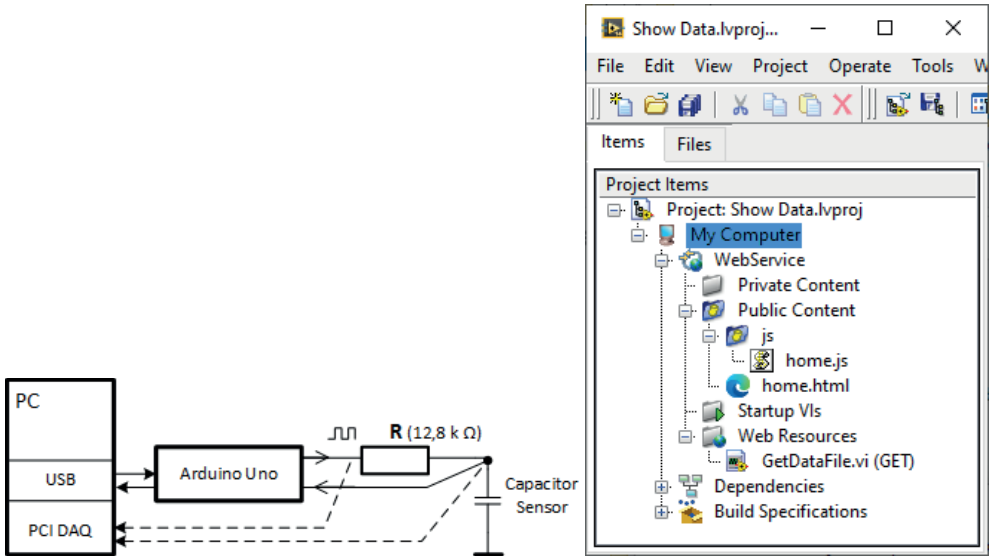


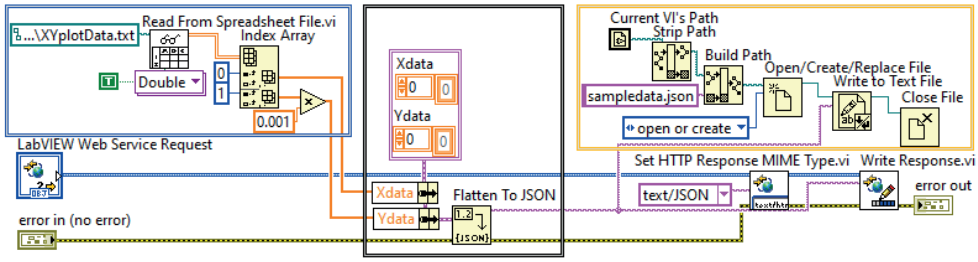
Figure 4. A scheme of the new workplace with Arduino and the home-made capacitive linear position sensor (left) and the LabVIEW project items window with Web Service used for data distribution (right)

Source: Own work.

Students do not own complex measuring devices and we needed to give them possibility to remotely investigate signal shapes in described sensor circuit. Therefore a new remotely accessed workplace was built in our laboratory according to scheme shown on the left-hand side of Figure 4 with Arduino Uno as a pulse generator and a DAQ (Data AcQuisition) card sampling signals at the input and output of sensor RC circuit. The structure of DO module of workplace with capacitive linear position sensor is similar to previous case of rotary sensor. Number of cases and also branching in mode 5 is the same. The button labels sent in mode 6 had to be changed to:

Delay, *RCinWave*, *RCoutWave*, *Trend*, *Help/Clear*. The main task was changing implementation of modes 7–11 which has to handle button pressed commands. If *Delay* was pressed, then the delay value of pulse edge slowed by the RC circuit is obtained from Arduino and added into the Listing indicator and internal array. The *RCinWave* button is processed by sampling of pulse signal at the input of RC circuit using DAQ card and drawing the curve into *Graph* tab. *RCoutWave* means the same for the output of the RC circuit. *Trend* replaces the curve in *Graph* tab by a stored trend of delays. The *Help/Clear* button has dual functionality. It erases stored trend data and shows next help page in *Image* tab.

Another challenge in the adaption of laboratories for home work was to distribute data even to small devices. Many students had to move from dormitories to the countryside where probably only mobile phone internet connection was available. For such cases control of remote application was not possible in our system as GO module was published by Remote Panel. We decided to expand the application with Web Service. In the block scheme of the GO module, we implemented saving graph data to a file every moment when a new graph draw command (reply) is received from the DO module. Then a LabVIEW project was started (Bauer, Ionel, 2013) which includes Web Service. Here we added new .vi file working as a small application which reads the data from the data file and passes it in the JSON (JavaScript Object Notation) form for next processing.



F i g u r e 5. Block diagram of *GetDataFile.vi*

S o u r c e: Own work.

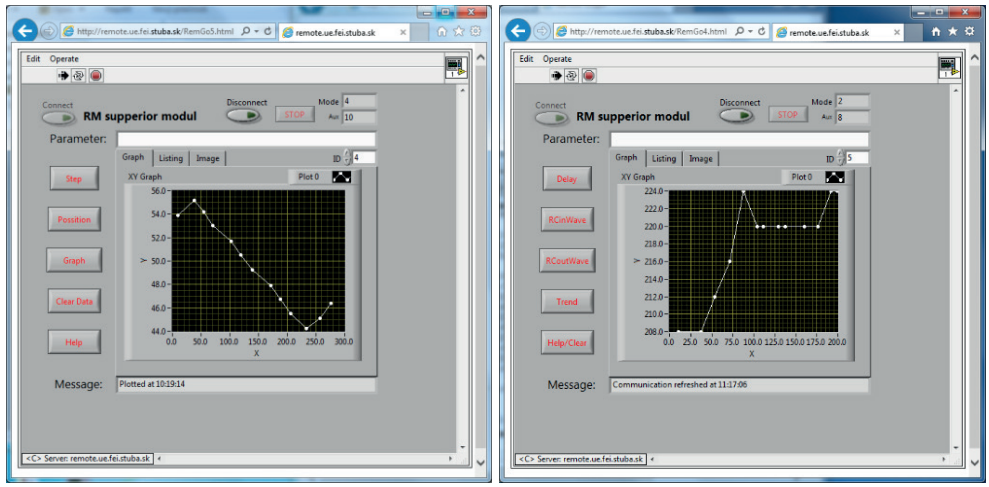
On the right-hand side of Figure 4, the project items window is shown. It mainly consists of one .vi file and of a Public Content directory of WebService composed from .html and .js file. Block diagram of *GetDataFile.vi* is depicted in Figure 5. It receives Web Service request and writes JSON data in response. Except nodes serving the Web Service the block diagram is visually divided into three parts (frames): reading data from spreadsheet file, converting to JSON format, saving JSON data to another file. The main implementation of publishing data to the remote user is achieved in JavaScript file *home.js*. Here AJAX HTTP GET request is realized over *getJSON()* method with the parameter of URL address where the request is going to be sent. A copy of local *document.URL* has to be modified to point to *GetDataFile: URL = URL.replace* (*"home.html", ""*) + *"GetDataFile"*; JSON data of the response are visualized using Google Charts based on HTML5/SVG (the latest evolution of HyperText Mark-

up Language/Scalable Vector Graphics) technology with no plugin requirements. Line Chart is used for drawing a graph and Table for showing the data in a numeric form.

3. APPLYING OF THE SYSTEM AND DISCUSSION

Designed sensor workplaces can find application in several subjects. However, as discussed in previous chapter they were intentionally targeted on subjects Technical diagnostics and Microcomputer techniques. Both workplaces can run in our laboratory and be accessed remotely in parallel. Actually it is possible to let them run on the same computer, however, we used two separate computers every hosting different local DO module. Control was realized via two GO modules running on one server computer, see RemGo5 and RemGo4 in Figure 6. There is no predestination how GO and DO modules are pared. Rotary position sensor with CAN output uses ID 4 while workplace of capacitive linear position sensor has ID 5. The user can establish connection for any ID from any GO module. If the selected ID would be occupied the connection is not established.

If ID 4 was chosen in GoModule5 like on the left-hand side of Figure 6 then student of diagnostics can turn the stepper motor (button *Step*) after that read new CAN data from sensor (*Position*) and draw trend into a Graph (*Graph*). DO module maintains a safe direction of rotation such that there is not risk of damage of rotary position sensor. In Figure 6 the allowed interval was set to 45–55% of 30 turns range. During semester we rescaled y-axes and let students calculate the interval from CAN data (see *Listing* tab in Figure 2).

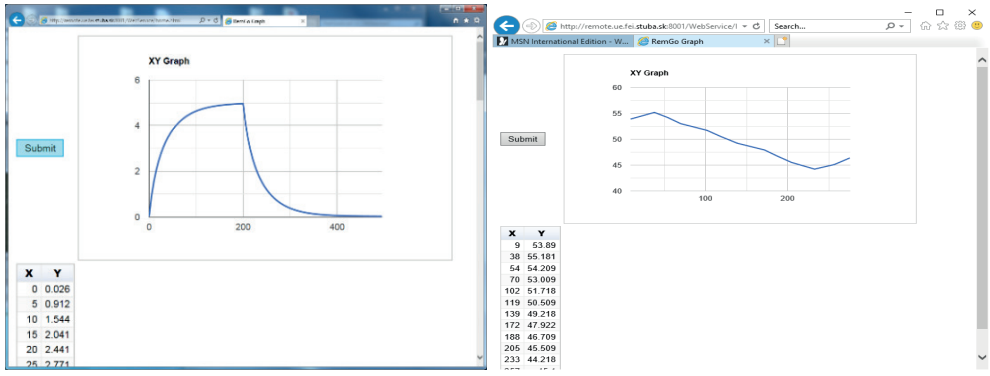


F i g u r e 6. Superior modules RemGo5 (legt) and RemGo4 (right) accessed from the web browser

S o u r c e: Own work.

GoModule4 presented on the right-hand side of Figure 6 was paired with workplace of capacitive sensor where the position should be evaluated from capacitance in RC circuit by Arduino firmware. Such was the task for students of Microcomputer tech-

niques and the workplace enabled them to check output signal (button *RCoutWave*) of RC circuit when pulse signal is at the input (*RCinWave*) which helped them implement Arduino firmware generating pulse signal and measuring delay (*Delay* – shown in *Listing* tab) representing deceleration of rising or falling edge of signal at capacitor. Other possibility of the workplace is to observe stability of delay (*Trend*). For visibility of trend fluctuation during the short time test presented on the right-hand side of Figure 6 (x-axes in seconds), the teacher located in the laboratory manually moved with the electrode of the capacitor.



F i g u r e 7. A distribution of measured data using Web Service: data from the rotary position sensor (left); data from the linear position sensor (right)

S o u r c e: Own work.

As the task of design of Arduino firmware measuring position with capacitive sensor was performed during substitute study from home it was important to distribute data also for students without availability of complete technical equipment for remote access. Adding Web Services and supplementary page based on JavaScript as described in chapter 2.1 the graph data could be shown also on computer without the LabVIEW runtime engine (cf. Figure 7) or on the mobile phone. This implementation does not allow control of the workplace but still helps to easy check shape of signal from RC circuit. On the other hand, the page with JavaScript also displays data from rotary position sensor if ID 4 is used with RemGo5 module like on the left-hand side of Figure 7. This underlines the flexibility of the modular system where the new GO module with modified functionality can be used with the older DO module.

For the workplace with capacitive linear position sensor we finally took a quick quiz oriented on parameters of PWM signal generated from Arduino for excitation of RC circuit and on parameters of the output signal sampled at capacitor sensor. The quiz was open after preliminary student experiences with the workplace and was strictly time limited. From 10 students of the subject Microcomputer technique the answers of 5 of them were completely right and other two students answered partially right. Remaining 3 students (30%) were not able to answer correctly within time limit.

CONCLUSION

The system of remote accessed workplaces has been described in the paper. It is based on modular concept and on cooperation of pairs of superior and target module. Modularity facilitates more flexible target application design as presented for modules of sensor application. Two such workplaces have been created simply by adapting previous sample application keeping rules of communication with the GO module. As it follows, there are no demands on knowledge about web technologies of designer of a new target module.

The system presented here was originally developed in the LabVIEW environment. Modular structure can help to overcome the requirement for the installation of the LabVIEW Runtime Engine on a remote device. The extension of the superior module has been proposed using Web Services and JavaScript, where Google Charts helped to visualize measurement results. Modified superior module can be employed even for older workplaces and target modules with full functionality. Compatibility with both sensor applications discussed above has been presented. The results of the quick quiz in reference to the Arduino term project with the home-made capacitive linear position sensor demonstrated the usability of the corresponding remotely accessed workplace for students working from home in makeshift conditions.

ACKNOWLEDGEMENTS

This work was supported by the Research and Development Support Agency under project APVV 15-0062 “Electromagnetic compatibility ensuring of monitoring systems of abnormal operating condition of nuclear power plant”.

REFERENCES

B a u e r, P., I o n e l, R. (2013). LabVIEW Remote Panels and Web Services in Solar Energy Experiment – A Comparative Evaluation, Proceedings from the *International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 23–25 May 2013, Timisoara, Romania (pp. 263–268).

Č e r v e ň o v á, J., K a m e n s k ý, M., & K r á l i k o v á, E. (2016). Modular remote access system for real experiments. Proceedings from *Radioelektronika 2016: 26th International conference*. Košice, Slovakia. April 19–20, 2016. Technical University of Košice, 2016 (pp. 340–344). ISBN 978-1-5090-1674-7.

G a r c í a - G u z m á n, J., V i l l a - L ó p e z, F. H., V é l e z - E n r í q u e z, J. A., G a r c í a - M a t h e y, L. A., & R a m í r e z - R a m í r e z, A. (2017). *Remote Laboratories for Teaching and Training in Engineering*. YILDIRIM, Sahin, ed. Design, Control and Applications of Mechatronic Systems in Engineering, InTech, 2017, 2017-05-03. ISBN 978-953-51-3125-0 (accessed 22 March 2018).

G u l a, M. & Ž á k o v á, K. (2017). Matlab2Web – Publishing Matlab Functionality to the Web. Proceedings from the *International Conference Distance Learning, Simulation and*

Communication 'DLSC 2017', 31 May – 2 June 2017, Brno, Czech Republic. http://dlsc.unob.cz/data/DLSC%202017% 20Proceedings%20Selected%20papers.pdf.

K a m e n s k ý, M., K r á l i k o v á, E., & Č e r v e ň o v á, J. (2018). Employing remote access teaching system for diode measurement. In E. Smyrnova-Trybulska (Ed.). *E-learning and Smart Learning Environment for the Preparation of New Generation*. Katowice–Cieszyn, Poland. October 15–16, 2018. Katowice: STUDIO NOA for University of Silesia, 2018 (pp. 465–476). ISBN 978-83-66055-05-6.

S i n g h, A. K., C h a t t e r j i, S., S h i m i, S. L., & G a u r, A. (2015). Remote Lab in Instrumentation and Control Engineering Using LabVIEW. *International Journal of Electronics and Electrical Engineering*, *3*(4), 297–304. DOI: 10.12720/ijeee.3.4.297-304 (accessed 10 January 2018).