NAPREDNI SOFTVERSKI SISTEM ZA MONITORING SOLARNOG NAPAJANJA

ADVANCED SOFTWARE SYSTEM FOR MONITORING OF SOLAR PANELS

Vuk JOVANOVIĆ*[1], Ilija RADOVANOVIĆ[2], Zoran STEVIĆ[3]
[1] School of Electrical Engineering - University of Belgrade, Computing center
School of Electrical Engineering in Belgrade, Serbia
[2] School of Electrical Engineering - University of Belgrade, Innovation center of
School of Electrical Engineering in Belgrade, Serbia
[3] School of Electrical Engineering - University of Belgrade, Technical Faculty Bor,
University of Belgrade, CIK, Belgrade, Serbia

https://doi.org/10.24094/mkoiee.020.8.1.155

Predstavljen napredni solarni sistem za proizvodnju električne energije sa akcentom na softversko unapredjenje monitoringa, prikaz i analizu podataka od značaja u cilju povećanja energetske efikasnosti. Računarski monitoring sistema prikazan je na bazi LabView softverskog alata sa proširenjima i unapredjenjima korišćenjem drugih specifičnih tehnologija i tehnika za obradu podataka. Sistem je testiran, a zatim i pusten u rad u laboratoriji za energetske pretvarače na Elektrotehničkom fakultetu u Beogradu.

Ključne reči: *solarni sistemi; obnovljivi izvori energije; monitoring; LabVIEW, Java*

In this paper an advanced solar system for power supply is presented, with special attention on software improvement of system monitoring, data display and analytics in order to achieve better energy efficiency. Computer monitoring is displayed using LabView software tool with extensions and improvements using other specific techniques and technologies for data analysis. System was tested, then commissioned in laboratory for power converters, School of Electrical engineering in Belgrade.

Key words: *PV system, renewable energy, monitoring, LabVIEW, Java*

1 Introduction

Nowadays, the important part of the energy production from renewable energy sources is to cut the costs and to create the more efficient systems. The monitoring of such a system, environment and the conditions in which the energy production is happening is crucial [1]. For PV systems the monitoring of the environmental conditions, as well as its operation is high priority task, following the exposure to different sources of performance degradation and faults especially in urban areas. Different application services are needed in order to convert raw data from sensors to useful information. Since the obtained information is time dependent and corresponds to the measured physical property in the exact time instance, the data volume that would be transferred between end devices and data base should be significantly decreased, so that real-time operations would not be effected [1-3].

Modern era advancement in technology brought the need for fast data exchange, acquisition and processing in almost any field of work. Software solutions are present in every aspect of everyday life and its job is to help us get results, statistics and representation of figures easier, faster and with less trouble than before. Nowadays it is important to have different kinds of reports about the acquired data, with ways to get valuable information from them that can help make future adjustments, decisions and right steps in order to upgrade the system as a whole. PV systems are widely spread and used, with different software solutions used for data acquisition, monitoring and exchange. This paper

* Corresponding author, email: vuk.jovanovic@etf.bg.ac.rs

presents an open source monitoring solution, developed for the PV system at hand. Its purpose is to acquire and parse the data that is generated by the LabView software tool using File I/O protocol [4]. Process and store the data to the relational database and make it available to the end user at real time.

Monitoring software solution developed for the PV system in this paper integrate four very important aspects of Big Data, Volume, Velocity, Variety and Veracity. LabVIEW software tool is used for reading the data from the PV system once every second, and storing it to an output file, at first glance the volume of this data is small, but as this is a continuous reading, the data is linearly growing high velocity data acquiring. Variety of data at the moment is low, but with future addition of different types of sensors and the upgrade of the PV system, the potential of variety in data collected is very high. Using File I/O protocol, as well as scheduling and polling system, the software at hand utilizes periodical file check. It accesses the file location, reads the contents of the file and checks if the file has grown since the last file check operation and by how much. Then it reads and parses the differential between already stored and processed data and the new one in the file. This is the new data acquired by the LabIEW application from the PV system since the last data check. Once the data is processed and stored to the relational database, it is displayed to the end user in different data-table and chart views which allows different sets of actions and manipulation.

2 Implementation

Modern monitoring systems in various branches of engineering often use one or more different software tools in their line of work. Though not directly associated with IT technologies PV systems have a wide range of possible applications and coexistence with informational technologies. This paper shows only a fragment of possibilities how different technologies can work and cooperate with diverse applications and software tools. As shown, this newly developed software system communicates with existing LabVIEW software tool via File I/O protocol, thereby enabling the cooperation of these two, from the implementational point of view, completely different software packages. It enables the cross-origin communication between two unrelated programs using a simple file I/O protocol, this way enabling data encapsulation, manipulation and processing in real time.

Above described monitoring software is a secure web application developed using a three-layer architecture stack additionally adapted for PV system functionalities. First (bottom) layer used for data storage is implemented using open source RDBMS PostgreSQL, a popular solution known by its performance and stability. Second (middle) layer is used for two purposes. Firstly, for Big data manipulation, reading and processing data retrieved from the PV system, and the process of storing it using Hibernate ORM and JPA included in Spring Data package. Second usage is for delivering data upon client request via secure RESTful web services. This layer is implemented using JAVA based technologies, Spring framework, Spring REST[5], Spring data, Oauth 2.0, JAVA IO etc. Third (front) layer is a javascript (typescript) web client, a frontend application implemented in Angular v.10[6], one of the most advanced open source frameworks, with various extension packages such as PrimeNG, Bootstrap, ngx-charts and others. These three layers communicate securely and share data in order to get the wanted result and present it to the end user via web browser over intranet or internet.

The figure 1. shows the software architecture stack of the system developed and used with the PV system described in this paper as well as the directional flow of data between certain parts of the application.

On the hardware side, the signals from PV panels and the LM35 temperature sensor were led to the input channels (from AI CH0 to AICH7) of the analog-to-digital converter (ADC USB 6009) [7]. The obtained digital signals through the USB port are connected to the computer (PC) in order to be processed by the software.

The application for primary measuring, displaying and storing the data was implemented in the software package LabVIEW [8,9]. Analog signals from the sensors are measured by DAQ Assistant, standard module of the LabVIEW package for the data acquisition.

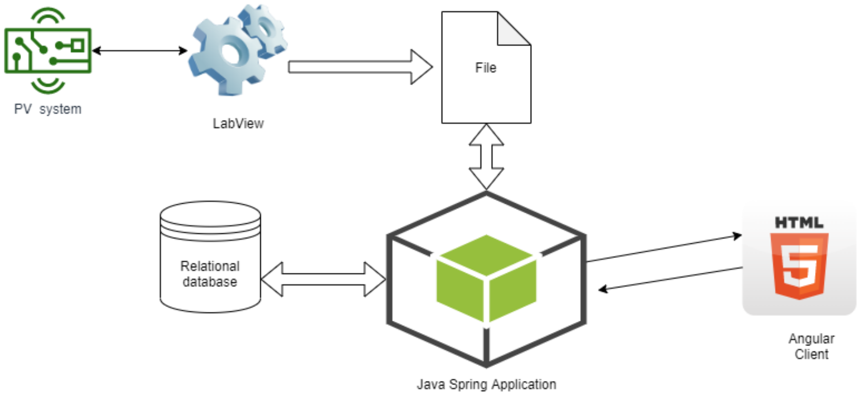

Figure 1. *Software technology stack diagram*

3 Results

Over a period of 5 days in August data was collected and monitored using the software system described in this paper. The data was continuously read from the PV system and stored to the files via Labview. The files were read accordingly, and parsed values were stored in the database.

The Figure 2. shows data-table data representation, the columns hold values averaged over one minute of time. The functionalities allow the end user to search the table over a period, at exact moments etc. and the search results as well as the initial set of data, can be sorted in ascending or descending order, thereby allowing the user to recognize the points of interest in the data pool presented.

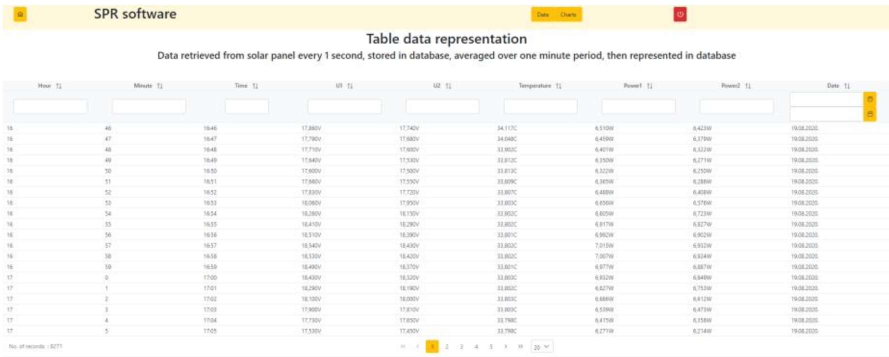

Figure 2. *Data-table representation of derived data stored in the database*

Second part of the application is a graphic representation of data over time. Averaged values over one minute of time are shown in Figure 3. as a multi-valued, multi-axis chart with additional functionalities such as period selection, data series inclusion and exclusion, data point focusing with value comparison etc. For further analysis it is possible to zoom each interval needed, as shown for one-day interval in Figure 4.

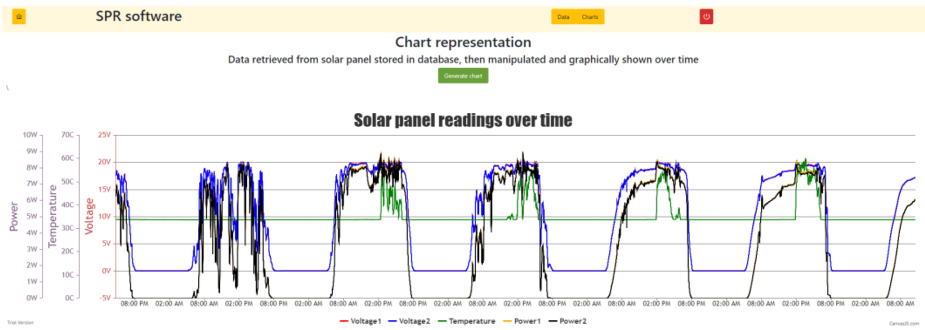

Figure 3. *Multi-chart representation of derived data over a 5 day period*

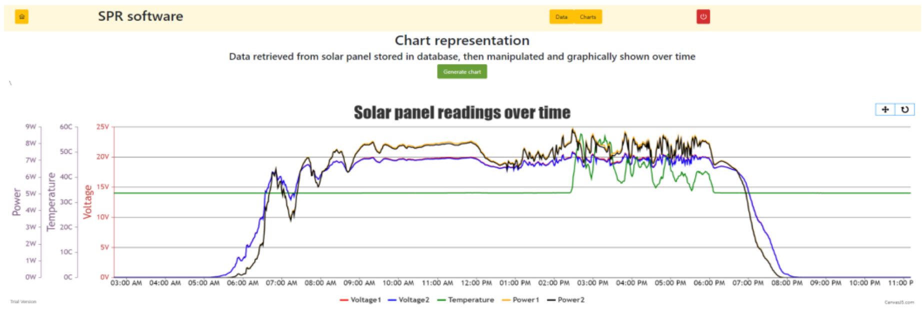

Figure 4. *One-day multi-chart representation of derived data*

4 Conclusion

Benefits of monitoring PV systems are highly diverse. From better insight to power management to identifying flaws and making decisions for future improvement. The system at hand is a scalable web application that can be easily upgraded to collect additional data and at a higher volume than now. Though developed locally and for the purpose of improvement of the scientific PV system at hand, the plan is to make this application available for educational purposes at all times, so the data is accessible via personalized user accounts. Another upgrade that is planned is a feature that will allow the software not only to monitor, but also to send certain command to the LabView application. Aside from mentioned functionalities there are little limits to improvements that can be made for the software to be more complete and have more use-cases available to the end user.

5 Acknowledgement

The authors gratefully acknowledge financial support from the Ministry of Education and Science, Government of the Republic of Serbia.

6 References

[1] **I. Radovanović,** Multi channel sensor Measurements in fog computing architecture for renewable energy sources systems monitoring, V MKOIEE Conference, Beograd, October 2017.

[2] *** OpenFog Reference Architecture for Fog Computing, OpenFog Consortium, 02/2017.

[3] **M. Abdelshkour,** “Iot, from cloud to fog computing,” March 2015. [Online]. Available: https://goo.gl/7zNxEd.

[4] *** File I/O protocol integration documentation - https://www.enterpriseintegrationpatterns.com/patterns/messaging/Chapter1.html

[5] *** Spring REST protocol implementation documentation - https://docs.spring.io/spring-restdocs/docs/2.0.4.RELEASE/reference/html5/

[6] *** Angular v10 framework official documentation - https://angular.io/docs

[7] *** NI USB-6008/6009. National Instruments, USA, 2015, http://www.ni.com/pdf/manuals/371303n.pdf.

[8] *** National Instruments. LabVIEW Basic I Course Manual. September 2006. Part Number 320628G-01.

[9] *** National Instruments. User Manual, 2010 Edition. Part Number 320999C-01.