Development of an Economical SCADA System for Solar Water Pumping in Iran

Mohammad Zamanlou
M.Eng. Student
Faculty of Engineering and Applied Science
Memorial University
St. John's, Newfoundland, Canada
Email: mzamanlou@mun.ca

M. Tariq Iqbal
Professor
Faculty of Engineering and Applied Science
Memorial University
St. John's, Newfoundland, Canada
Email: tariq@mun.ca

Abstract – **In this paper, we developed a cost-effective SCADA system for a solar water pumping system in Iran. The SCADA is based on IoT versions and is comprised of a raspberry pi zero W, Arduino nano, camera, SIM 5320A 3G module, voltage, current, and light sensors. We used Node-RED to design a graphical user interface and published it securely to the worldwide internet. This allows a user to connect to the server via an IP address and monitor and control the system. The implementation of the project resulted in an open-source server and cost around CAD$ 162.38.**

Keywords— SCADA, IoT applications, Node-RED, Arduino, Raspberry pi, Wireless technologies

I. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) is a technology that enables electrical systems to be monitored and controlled from remote areas. SCADA technology has been around for many years, and various types of it were used in the industry to collect data from the system's electrical components such as batteries, electro-motors, etc. In recent years, the introduction and development of the Internet of Things (IoT) version of SCADA technology avail small to medium scale electrical projects such as solar water pumping of SCADA's advantages. It provides such projects with reliable and flexible control while being cost-effective to develop and install. [1-4]

In Iran, agriculture is facing many challenges due to the shortage of water and using new technologies such as drip-irrigation provides continued agricultural growth in the country. One of the applications of IoT based SCADA is in agriculture on solar water pumping systems. Such a SCADA system expands the development of a remote solar water pumping systems in the farming fields, such a system could be located at a far distance from living areas enabling farmers to efficiently operate the irrigation of their garden remotely.

Fig. 1 shows a schematic of a solar water pumping system. In this project, a SCADA system was developed to take measurements of various sensors such as PV voltage and current and control various switches for the submersible pump and photovoltaic panels. In addition, the SCADA provides monitoring for the environmental parameters such as a picture of the field and solar irradiance.

II. LITERATURE REVIEW

In [2], the authors presented a low-cost open source SCADA system to monitor and control a solar system where a customer can log-in to the dashboard and apply controlling over their solar system. They used a raspberry pi, an ESP32, a WiFi router and a few sensors. Also, they used thinger.io as their design hub. This system is only implementable in places where a WiFi is available also they are using a paid IoT hub named Thinger.io which adds up to the total cost of the system. We are advancing their system by adding a cellular network modem to make it possible to implement such system in remote areas and the designed project is programmed and secured completely on the local server using Node-RED. In [3], the authors designed a home automation system which shares a lot of similarities with the common IoT SCADA system. In their work, they made use of a raspberry pi, Arduino uno, and relays to implement controlling and monitoring through a dashboard. While this proposed system was successfully able to control the IoT devices, it uses Dataplicity to secure and expose the server to the internet which is a third-party application and puts a high risk over the server that data might be accessed through third-parties. Besides, their proposed system is not able to connect through a cellular network and is only available in urban areas. In our system, we

978-1-7281-9615-2/20/$31.00 ©2020 IEEE

are using NGINX to secure and proxy pass the IP requests to the server and a 3G modem to connect to the cellular network [2, 3].

III. SCADA DESIGN

Fig. 2 shows a diagram of the proposed SCADA system. In this diagram, an Arduino nano is connected through a USB port to the raspberry pi and allows it to control sensors and LEDs connected to the ports of the Arduino. Also, a camera is used to picture the crop field or pumping system so that the user can monitor them. For wireless connectivity, we used both WiFi and 3G cellular network. The design procedure and explanation of the project are included in the following.

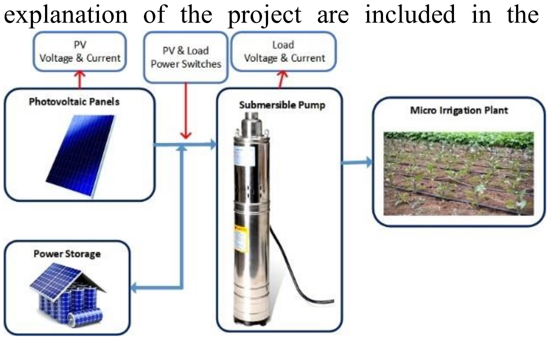

Fig. 1 – Schematic diagram of a solar water pumping system

A. WIRELESS CONNECTIVITY

A wireless connection is a great advantage for controlling systems in the era of modern technologies. In this regard, firstly, we took advantage of the raspberry pi's onboard WiFi module to connect it to the home router. Ref. 5 provides instructions to operate the raspberry pi without any need to plug in a monitor, mouse, or keyboard, which, in turn, made the system more cost-efficient. Secondly, we set up a cellular connection using instructions in [6, 7] to connect the server to a cellular 3G network. The cellular connection provides the server with the advantages of a cellular network, such as its availability.

B. WEB-BASED DASHBOARD

In our system, we used Node-RED in programing the raspberry pi to create a server. Node-RED is a programming environment using various types of nodes (i.e., serial port node) and mouse driven wiring of nodes. Arduino is controlled through Node-RED through a very user-friendly editor based on the internet browser. We used the Node-red-dashboard node to create a graphical user interface where the operator of the server can execute the system for monitoring and controlling purposes.

The final front end of our system is a web-based graphical user interface, so-called dashboard shown in figure 3. There are two tabs in the dashboard: Main tab and Environment tab; the main tab contains historical charts for presenting electrical data such as voltage and current, gauges for showing instant electrical measurements of voltage and current, and switches to control two LEDs which represent the power switches in the system. The environment tab includes a jpg picture box with a button to take a photo that allows the user to remotely monitor the field and sunlight data. Using this dashboard, a user has complete remote control over the system from anywhere in the world through 3G internet. Fig. 3A, and Fig. 3B shows the results of the dashboard that the user can interact with. [8-9]

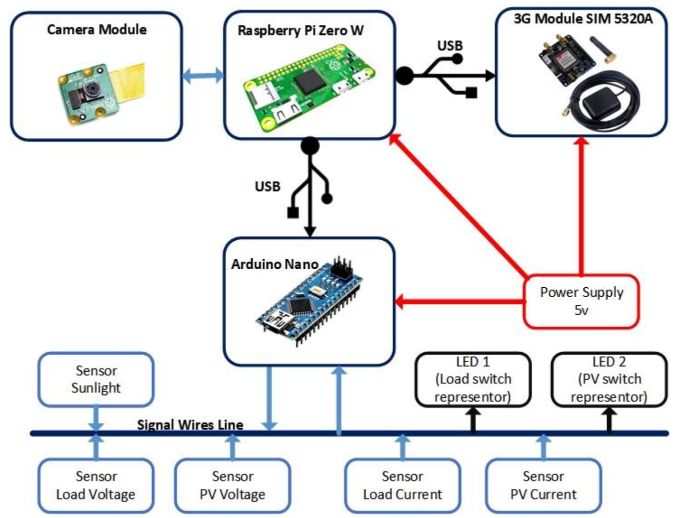

Fig. 2 – Block diagram of the server with various components

C. COMMUNICATION WITH ARDUINO NANO AND ELECTRICAL SETUP

Arduino nano is a small microcontroller board based on ATmega328 and has a very low power consumption. Using Arduino nano allows the server to sense and control various types of signals, such as analog signals for voltage measurement. There are many ways to communicate with an Arduino using Node-RED, such as Serial, Firmata, and Johnny-Five. We chose Firmata because it provides simplicity to the server and allows the computer to have direct access to the input and output pins of the Arduino.

In this setup, we designed a circuit to control two LEDs which represent load and PV power switches and sensors to measure load and PV voltages, load and PV currents, and sunlight.

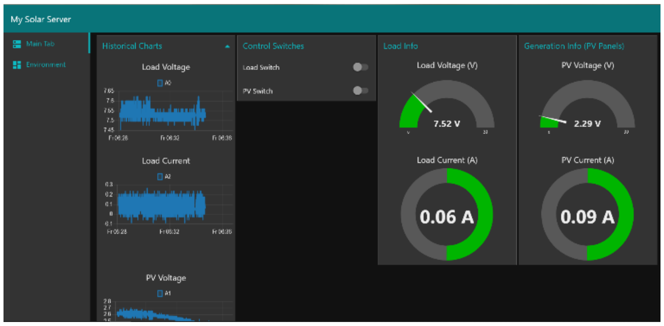

Fig. 3A - Dashboard Main Tab

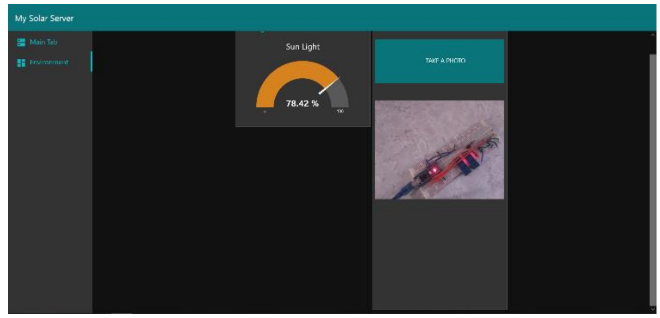

Fig. 3B - Dashboard Environment Tab

D. Exposing the Server to the WEB and Security

Since the user needs to remotely access the server, it needs to be published on a WEB server. On the raspberry pi, the NGINX web server application was used to create a web server and publish the dashboard on that server. However, publishing a local server on the world wide web where a tremendous number of threats exist, can put a very high risk on the server which is designed to monitor and control the solar water pumping system. In order to solve security concerns, we set up an HTTP authentication with NGINX; hence, the user needs to enter the username and the password of the server whenever connecting to the server IP address. All in all, using this method, the server is securely accessible through worldwide WEB and server IP is also known to the user. [10]

E. Cost Analysis

The designed system is very cheap to install and use. The cost of the system includes a low power raspberry pi zero w (CDN$ 37.60), an Arduino nano (CDN$ 4.80), and a camera (CDN$ 13.99), and a 3G internet module SIM 5320A (CDN$ 105.99). We should note that the raspberry pi zero W has an onboard WiFi module that allows it to connect to an internet router when available. In this case, we do not need a 3G module, and its price would be waived.

In addition, the cost of high-speed internet service is very negligible; for instance, the cost of a one-year plan with 48 GB internet is about CAD$ 6.5 from a carrier named Irancell in Iran. This advantage allows our system to be very profitable in the long run. The total cost of the system is between CAD$ 56.39 to CAD$ 162.38, depending on which internet type we choose to use.

There are many IoT hubs in the market that share similarities with our proposed system. As an instance, Microsoft Azure IoT Hub is a product that provides a platform to connect, monitor and control IoT devices. While the components required to implement an IoT SCADA application with Azure is almost at the same cost, it is required to pay an expensive monthly fee costing minimum 12.80 $CAD per each IoT Hub unit to implement the SCADA with it. It is worth mentioning that the number of messages communicated through IoT Hub in each plan is highly limited. This example shows the advantage of our designed system over other systems. Hence, the system is cost-effective in comparison with available IoT hubs such as Microsoft Azure in which the server is based on an internet cloud and are required to purchase their expensive plans. [11]

The experimental setup of the SCADA server is shown in Fig. 4.

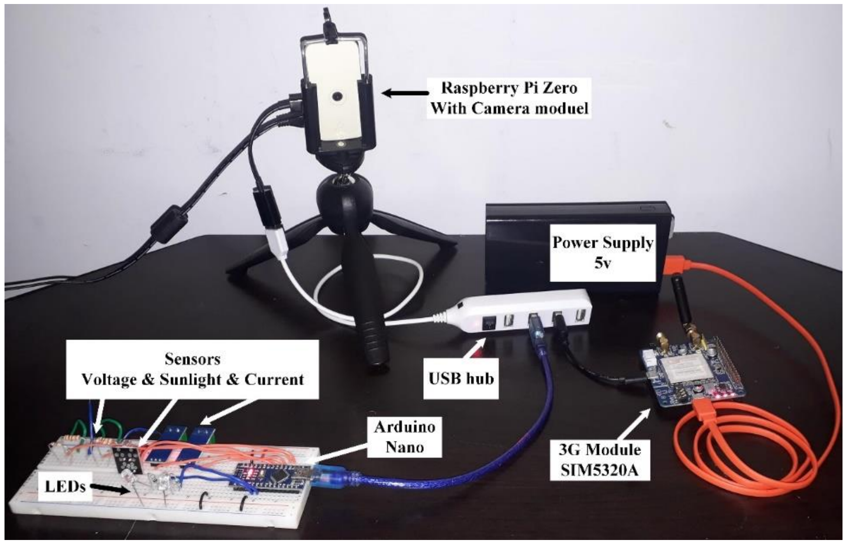

Fig. 4 - Experimental Setup of the SCADA server

IV. CONCLUSION

In this paper, the design of a cost-effective SCADA system for a solar water pumping system was presented. It consisted of a raspberry pi zero, Arduino nano, camera module, 3G module, and sensors for voltage, current, and sunlight and two LEDs. This server provides the system with a secure and complete monitoring and controlling GUI hub where the user can connect from anywhere through the internet and monitor and control a remote solar water pumping system. It is worth mentioning that the proposed system is implementable in any solar pumping system in the world where there is a connectivity to the internet available. The only difference will be the cost of a internet plan from the internet service providers.

Acknowledgment

The authors would like to thank Avrin Goom Azaran for providing funding for the research.

REFERENCES

[1] S. Phuyal, R. Bista, J. Izykowski, and D. Bista, "Performance Analysis of New SCADA Interface Developed in C# Environment," 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2020, pp. 1-4, doi: 10.1109/SCEECS48394.2020.209.

[2] L. O. Aghenta and M. T. Iqbal, "Development of an IoT Based Open Source SCADA System for PV System Monitoring," 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 2019, pp. 1-4, doi: 10.1109/CCECE.2019.8861827.

[3] C. Bepery, S. Baral and A. &. H. F. .. Khashkel, " Advanced Home Automation System Using Raspberry-Pi And Arduino," International Journal Of Computer Science And Engineering, vol. 8(2), pp. 1-10, 2019.

[4] S. R. Nath and R. Mandal, "Some studies on performance analysis of two different laboratory scale solar photovoltaic water pumping in irrigation systems," 2016 21st Century Energy Needs - Materials, Systems and Applications (ICTFCEN), Kharagpur, 2016, pp. 1-4, doi: 10.1109/ICTFCEN.2016.8052739.

[5] "How To Setup Raspberry Pi Zero W Headless WiFi" https://core-electronics.com.au/tutorials/raspberry-pi-zerow-headless-wifi-setup.html

[6] "How to setup a USB 3G Modem using usb_modeswitch and wvdial" https://tchekbo.wordpress.com/2018/03/31/how-to-setup-a-usb-3g-modem-using-usb_modeswitch-and-wvdial/

[7]" Raspberry Pi - Using a GPRS Modem for Network" https://www.youtube.com/watch?v=T2NBpfgfPaI

[8] "Node-RED", https://nodered.org/

[9] "Node red Dashboard" https://flows.nodered.org/node/node-red-dashboard

[10] "How To Set Up HTTP Authentication With Nginx On Ubuntu" https://www.digitalocean.com/community/tutorials/how-to-set-up-http-authentication-with-nginx-on-ubuntu-12-10#prerequisites

[11] "Azure IoT Hub pricing" https://azure.microsoft.com/en-us/pricing/details/iot-hub/