

Autonomous flexible low power Industrial IoT controller for solar panels cleaning systems

Dumitru-Cristian Trancă, Daniel Rosner

Computer Science Department
Faculty of Automatic Control and Computer Science,
University Politehnica of Bucharest, Romania
{dumitru.tranca, daniel.rosner}@cs.pub.ro

Alexandru Viorel Pălăcean

Automatic Control and Systems Engineering Department
Faculty of Automatic Control and Computer Science,
University Politehnica of Bucharest, Romania
alexandru.palacean@aut.pub.ro

Clean energy production increment is being led by advancements in photovoltaic plants, both for on-grid power generation, and for local usage (farms, industrial plants, etc.). The main challenge faced by PV plants is the dreadful reduction of power production caused by the accumulation of dirt and dust on the panels. This effect has a higher magnitude in areas where dust storms or dusty environments are present (such as deserts, urban areas). In order for the PV plants to remain operational at nominal installed power, periodical cleaning of the solar panels is required. For larger PV plants (with power north to 1MW) human manual cleaning of the panels is not feasible, especially in harsh and torrid environments or in cold weather conditions. In this paper, we present an architecture for a custom built low power autonomous Industrial IoT Controller dedicated for PV solar panel cleaning solutions. The controller, apart from controlling the operation of the mechanical device, is also designed to be connected to the internet, to be remotely monitored and to be capable of running VPN, iptables, and other security features that current devices cannot implement.

Keywords—*PV power plant, solar panel, embedded controller, industrial, IoT*

I. INTRODUCTION

In a world of increasing population, and increased usage of devices, factories, electric cars, the rise of electric power consumption is inevitable. Currently, the energy industry is heading towards a more environmental friendly means of producing electricity. The most common types of eco power plants are photovoltaic power plants, wind turbine power plants and micro-hydro power plants or tide power plants. Micro hydro power plants depend on river flows – flow rate, falling distance, volume rate, wind power plants depend on wind speed, duration and frequency of winds, and tide plants depend on the tides. The opposite of the condition dependent power generation methods mentioned earlier solar power plants require light, a condition that can be satisfied in many use cases. Solar plants can be operated on-grid, off-grid, with or without storing the produced energy and can serve applications that are ran during the day (factories) or they can just compensate the consumption of large energy consumers, reducing costs with electricity. They are widely used for home

appliances or as power production stations for energetic power systems.

Solar power plants can have installed powers from tens of kilowatts up to hundreds of Megawatts. It is shown that based on the irradiance power of the sun, on the efficiency of solar cells (of 10-20%) and on the efficiency of conversion of DC to AC, it takes on average about 1.5-4 hectares (dependent on the geographical and meteorological position and conditions) for producing 1MW of electrical power [1]. Although their wide availability and ease of use, solar panels are affected by one of the most common environmental factors: dust. Dust accumulation has a considerable effect on the power production, reducing power output down to 50% or even less [2]. To overcome the effect of particle accumulation on solar panels, different kind of cleaning methods are used, depending on the dimensions of the solar plant. Cleaning by using human operator, semi-automated cleaning system (that requires the intervention of human operators but the operation itself is done automatically) or fully automated cleaning systems (that automatically asses the conditions and the necessity of the cleaning procedure).

In this article, we will present an architecture for an industrial low power controller specially designed for controlling cleaning robots for fully automated cleaning processes of solar power plants. The controller runs a Linux Debian based distribution, controls the drives of the movement motor and brush motor, monitors the position and all parameters of the system, allows configuration through a web interface and can be connected to the internet via LAN or WLAN. Also, for dispatched power plants, the controller can be integrated in SCADA systems by using Modbus TCP for data transmission and control.

In chapter II we will present existing solutions and current research in the field of solar panel cleaning. Chapter III will present the hardware architecture and special details regarding interconnection with sensors and data acquisition techniques, and Chapter IV is dedicated for the software architecture and algorithms implemented in controlling the robot. A use case of the controller and the results obtained during operation will be presented in Chapter V and in Chapter VI we will conclude the most relevant aspects regarding our solution.

II. STATE OF THE ART

In [3] Jawale et al. start with the premises that accumulation of dust on solar panels considerably reduce the efficiency of the plant, and introduce a cleaning bot to help in ameliorating the efficiency of solar panels. The authors show the difficulty in cleaning the solar panels is affecting solar panel owners. The solution described by the author is simple, yet they manage to obtain relevant results - showing an increase of 70 to 130% of electric power after the cleaning process.

In their work related to solar panel robot cleaning system [4] M.A. Jaradat et al. analyze in detail the effects of dust accumulation on solar panels and the financial lost due to power output reduction in PV power plants. Based on a review of multiple studies, the authors show that, based on field data from multiple PV systems in USA, power production loss is between 1% to 38% during one year. In their studies, different power losses varying from 10% up to 40% per year are caused by different type of dust particles accumulating on the solar panels. The authors propose an automated cleaning system based on using a single robot. The described system consists of the cleaning robot and a rail platform carrier that allows the transfer of the robot from one panel to the next as illustrated in Fig. 1. The design based on a symmetrical mechanism allows easier adaptation for larger solar panels as seen in Fig. 2.

The system showing the three-step operation mode of the cleaning system:

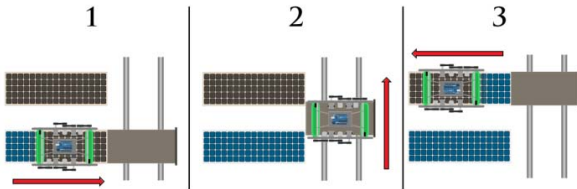


Fig. 1. Robot cleaning execution steps [4]

The authors show a microcontroller based system that connects to a motor driver used for movement and brush motion. The paper presents the results of the design as well as the measured conditions in which the robot can work.

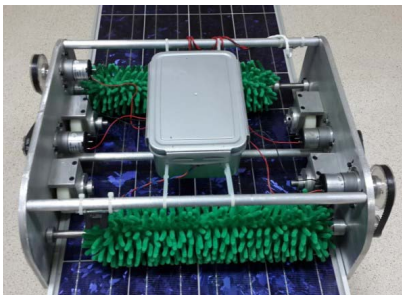


Fig. 2. Cleaning robot in action [4]

There are other research topics focused on dry cleaning methods that allow both automatic cleaning of PV panels and dry-cleaning technology. In [5] the authors aim to ameliorate the efficiency and high power consumptions of current dry-

cleaning systems. The system proposed in their paper is designed to work at a preset timing and to be simple, user-friendly, robust, lightweight precise and to eliminate the demand of workforce due to its autonomy.

As illustrated in Fig. 3, the design consists of three cleaning mechanisms: air compressor, polyurethane foam roller and polywool synthetic duster (electrically charged). Their system starts the operation just when the constraints regarding humidity are satisfied. The panels are brushed, cleaned with the polyurethane foam roller, and after that by the polywool duster. The mechanical design is based on a triangular shape on which the main components are placed. The structure is attached on two rails with both mechanical functionality and electricity distribution for the cleaning robot.

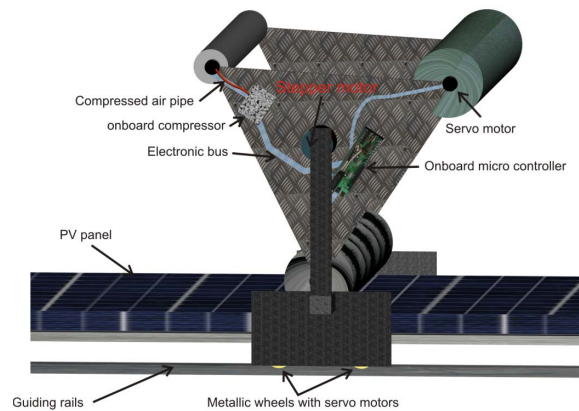


Fig. 3. Cleaning robot structure [5]

Besides experimental cleaning systems, commercial cleaning systems are available. Ecoppia presents their water-free robotic PV panel cleaning system (Fig. 4) implemented in over 300MW of deployments.



Fig. 4. Commercial solar panel cleaning system [6]

The producer adverts the system to be capable of cleaning over 80 million solar panels in the harsh desert conditions. Such devices use custom made controllers and electronics optimized for the chosen components [6].

III. CURRENT CONTROLLER SOLUTIONS

Regardless of the mechanism and mechanical construction of the cleaning solutions or robots, one of the important components in such a structure is the controller. In some applications, simple microcontrollers are chosen as the “brain” of the robot. When professional, industrial graded, rugged robots dedicated as an off the self solutions are designed, more complex controllers are used.

When used in grid tied large photovoltaic power plants, legislation and maintenance costs constrains force the need for remotely controllable and remotely monitored systems. In energetic dispatch and SCADA systems, as well as in local monitoring system of grid tied PV plants, standard industrial protocols such as Modbus, Fieldbus, Profibus, OPC, are used. In order to respect declared power, so that to fulfil energetic requests and set points from the national energetic dispatch system, PV plants must, at all times, maintain the solar panels clean and know when was the last cleaning done and what is its current status [7].

Commercial controllers have integrated libraries used for industrial communication protocols. Although extendible, with integrated analog inputs and outputs standard off the shelf controllers are not meant for portable battery operated systems.

In some robotic cleaning systems for PV plants robots are battery operated. The rows of solar panels range from tens of meters to hundreds of meters and energy efficiency of the controller is important. Most of the energy is used by the motors to power the brush thus the controller should be as power efficient as possible.

Industrial controllers (e.g. Fig. 5) are usually powered at 24VDC and have power consumptions of 7-30W or even more depending on the number of inputs, outputs, additional modules, and communication extensions. When systems like the ones describer at [3], [4], [5] and similar versions use batteries of 120-200Wh capacity, overnight consumption of usual industrial controllers can discharge the battery more than 50%. Besides energy efficiency, mechanical dimensions are an important factor in autonomous embedded systems.



Fig. 5. ILC 151 With Ethernet [8]

Depending on the client needs, the controller should be capable of connecting through Wi-Fi or 3G/4G to the internet, via VPN for remote control purposes. Usually, controllers do not offer easy methods of adding 3G or Wi-Fi capabilities. For

each separate functionality, the integrator needs to attach a special module to the controller to offer the desired functionality. Extending functionality or memory via USB modules like in laptops, tablets and PCs is not available in industrial equipment and dedicated equipment is mandatory (e.g. Fig 6, GPRS module).

For low power automation systems, researchers have been using different embedded boards with low power microcontrollers or in some cases more powerful solutions with system on a module / system on a chip solutions.



Fig. 6. Industrial GPRS module for ILC Controller [8]

In custom build applications, such as [3], [4] and [5] development board with low power controllers are used. Depending on the number of functionalities, of inputs and outputs and of computational power needed for the algorithm, simpler (using microcontroller) or more complex boards (that use System on Chip) are used. In research and prototypes, controllers are clocked between 1-20MHz (for microcontroller boards) up to 1-2GHz (for SoC), have a memory of 1-64kB (for microcontrollers) or run an OS from an SD card (for SoC boards).

IV. HARDWARE ARCHITECTURE

Most of the power consumed during the cleaning process powers up the motors (50W brushless motor used for moving the cleaning trolley and 100W brushless motor dedicated for the brush rotation). For recharging the batteries and making the cleaning trolley energy independent, a solar charger and solar panels are used. Since the cleaning process is repeated at every few days or weeks, the main concern is to reduce the discharge cycle due to the power consumption of the controller. This paper presents a research and a solution to obtain a low power controller for a custom-built cleaning robot.

The hardware presented in this paper has been designed to support the following specifications: Total consumed power in full load operation <3.5W, 4 analog outputs, 6 NTC inputs, 6 isolated digital outputs, 5 isolated digital inputs, 12 digital inputs, 12 open drain/open collector digital outputs, Ethernet connection capability, Wi-Fi connection b/g/n, USB ports for extension, RS485 communication line. Besides hardware requests, power consumption and interconnection constraints, the hardware architecture allows flexibility in running control algorithms enabling the user to easily adapt the software for other solutions.

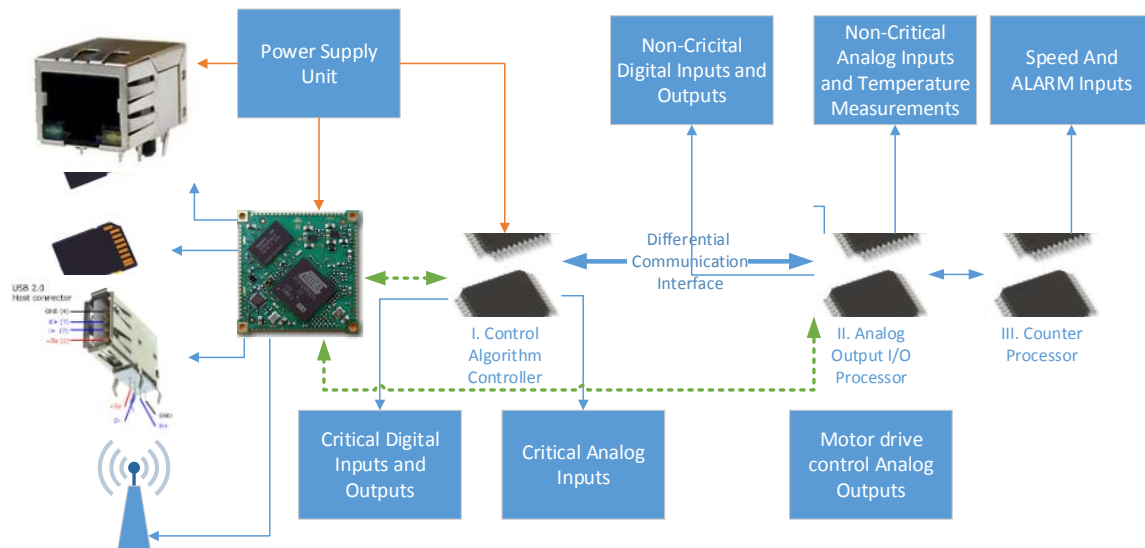


Fig. 7. Hardware architecture

The hardware architecture, although all components are placed on two PCBs, is designed as a distributed and parallel control and monitoring system. The main controller, Aria G25, includes an Ethernet physical interface, 3 host USB 2.0, SD card support, and multiple serial communication interfaces, I2C and SPI. [9] The controller boots a Debian based Linux image from the SD card that contains a web-server allowing the user to configure, monitor and control the state of the robot. Since the ARIA G25 does not have enough digital or analog peripherals to fulfil the needs for controlling all the actuators, two AVR based microcontrollers have been added. The microcontrollers are interconnected with a differential communication line since they are placed on different boards. The interconnection line is multiplexed by Aria and can be commuted either between the two microcontrollers, either between Aria and the non-critical microcontroller. A separate communication line is present between Aria and the microcontroller that runs the control algorithm. In the proposed architecture, critical inputs and outputs refer to the signals that are mandatory for the control algorithm and the outputs that control the motion of the robot. Critical digital inputs represent the signals from the limit switches that don't allow the robot to fell off the frame and from the buttons dedicated for user control. Critical digital outputs refer to the signals that control the driver (start, run/break, direction of rotation) and the power relay that powers down the assembly. Critical analog inputs are used to read the input from current transducer for each drive and from the batteries. In the case when current exceeds a threshold limit, it is considered that a fault has appeared and the robot is halted. Analog outputs dedicated for controlling the speed of the motors are considered noncritical, since the elementary digital controls considered critical (that enable the drive to run or bring the motor to a halt) are enough to serve for emergency stop. Non-critical digital inputs and outputs are connected to LED indicator or user operator buttons (such as Start, Stop, Alarm reset), or to sensors (rain detector). Non-

critical analog inputs are the temperatures, battery voltage, wind speed, solar irradiance and accelerometers that can be attached to the second microcontroller. Aria G25 controller board has access to the reset mechanism of the two microcontrollers allowing a hardware reset when and if they become non-responsive. Aria has its own watchdog timer that assures that in case of failure or software deadlock a reset is executed and the device is restarted so that the control over it is not lost.

The hardware architecture allows executing distributed real-time control algorithms without the need of real-time operating system. The first microcontroller, besides reading the inputs and controlling the outputs, it executes the control algorithm that gives the robot its functionality. Although the second controller is designed to work as an I/O processor it executes a function generator operation: it ramps up or ramps down the analog outputs with a certain slope independent of the master microcontroller and interprets data coming from the speed impulse generator of the driver.

The physical implementation of the proposed architecture from Fig. 7 can be seen in Fig. 8 – the components are split on two boards interconnected with a DB15 male and female connectors.

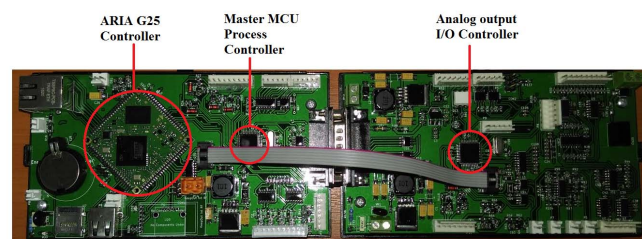


Fig. 8. Controller - Physical implementation

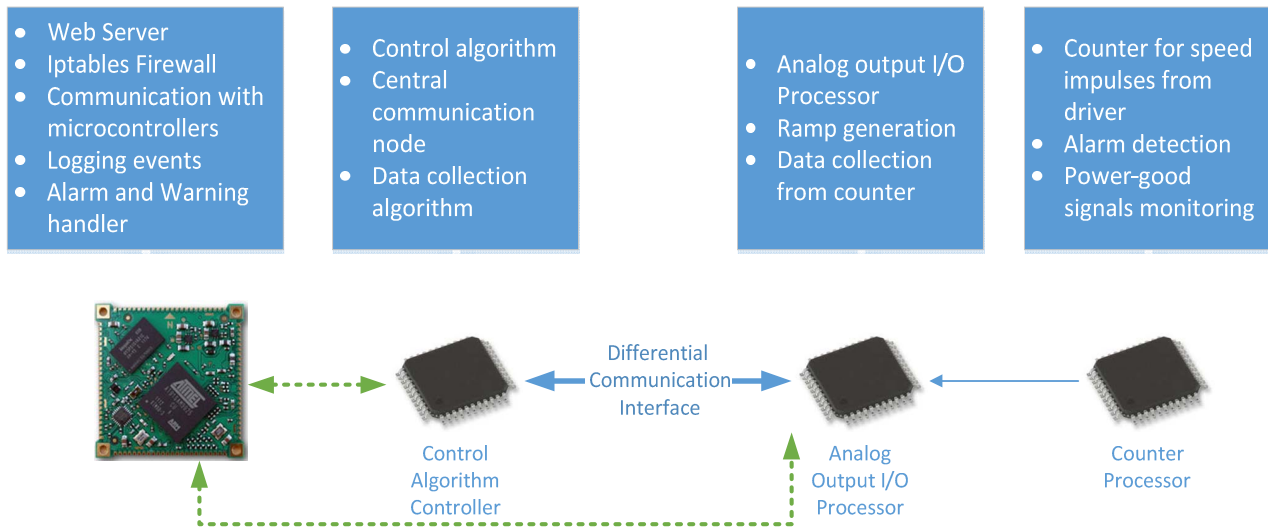


Fig. 9. Software architecture

V. SOFTWARE ARCHITECTURE

The software architecture (Fig. 9) is designed to efficiently map on the hardware and maintain the flexibility of the hardware architecture. It allows running a non-real-time operating system while the critical algorithm dedicated for controlling the robot runs on a separate microcontroller. In this section, we will discuss software aspects for each of the controllers.

The assembly of three microcontrollers works as a cascaded master-slave system where the slave is master for another entity. The main control microcontroller is the master (generates requests, sends commands, initiates the communication), while the analog output MCU responds to requests. Between the analog output MCU and the counter MCU the communication is made using the same master-slave topology. The counter MCU responds to requests and reports the measured data to the analog output MCU that acts as the master of the communication. Aria G25 enables or inhibits the control algorithm or enforces the state of the FSM that executes the control.

A. Aria G25 Software

The software system running on Aria is responsible for the cleaning device control and it is based on several distinct algorithms implemented in C and Python. The main application has the following jobs: to properly drive the cleaning process and to offer a user interface which can be accessed using a secured internet connection.

The communication with the onboard microcontrollers, the Alarm Manager, the Event Data Logging, and the constraints verification are implemented as separate functionalities, each in its own thread. The communication thread's role is to acquire the data from the microcontrollers and to send the appropriate commands to enable or disable the control algorithm or to control individual outputs. The

Alarm Manager thread continuously verifies the values of the sensors and boards internal parameters and compares them with the alarm and warning values set in the interface for each sensor or parameter. If the alarm/warning value is exceeded, an appropriate message is generated in the user interface. The Data Logging thread saves the values of the sensors and parameters at fixed interval of time (seconds) configurable from the interface. The main thread running on Aria aggregates the data from the interface and sensors and based on a list of rules starts the cleaning process.

The user interface (Fig. 10) is a website based on the Flask framework for Python. The website pages were developed using the Bootstrap framework to be responsive and to keep the right aspects and functionalities on any desktop or mobile devices. The communication between the web browser and the decision thread is done using SocketIO and event-based messages. The webserver waits for incoming connections both from the Ethernet interface and the Wireless card and the transmission is encrypted using SSL.

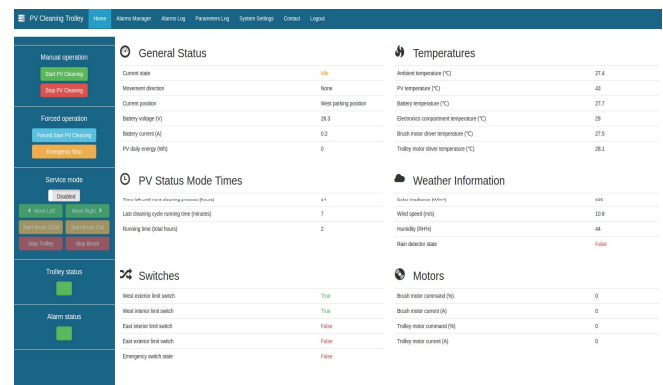


Fig. 10. Controller webpage dashboard

B. Control Algorithm MCU

The control algorithm is implemented using a FSM (Fig. 11). When the algorithm is not enabled, the microcontroller is used just as an I/O controller. Such an approach allows easy implementation of other algorithms on Aria while using all controllers for I/O operations.

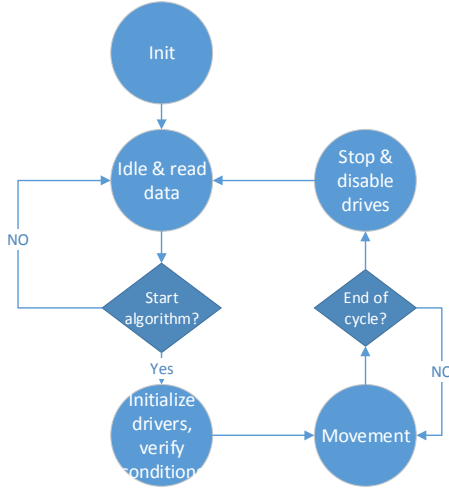


Fig. 11. Real-time control algorithm Finite State Machine

After power on, the algorithm enters the initialization step, where all the variables are initialized alongside SPI and UART communication with Aria and with the analog I/O MCU. After initialization, the state machine cycles in the idle state where it reads the data from the slave and from the local inputs and sends them back to Aria. In the eventuality of a command issued by Aria, the state machine assures initialization of all digital and analog lines to control the brush and movement motors. After initialization, the algorithm ramps up the motors and starts the movement of the cleaning robot. The moving robot will toggle mechanical limit switches wired to the critical digital inputs. On the limit switch event, the control algorithm ramps down the motors bringing the robot to a halt, thus finishing the cleaning cycle. At the end of the cleaning cycle the state machine returns to the idle state.

C. Analog Output MCU

The analog output MCU implements the functionality of generating a linear ramp of the output voltages that control the speed of the motors. Along with the analog output functionality, the MCU reads all non-critical analog inputs, locally stores the latest value, and reports it to the master controller when interrogated.

D. Counter MCU

The counter MCU has the role of measuring the period of the speed and alarm output signal from both motor drivers and reporting the values when interrogated. Besides this, it has the role of counting the number of impulses generated by the solar charger, measuring the daily energy generated by the solar charger. Furthermore, we have used the integrated

analog to digital converter of the microcontroller to measure the voltages on the board, assuring information of the onboard power supplies states.

VI. RESULTS AND CONCLUSIONS

The controller boards with a router device are encased in an ABS IP67 enclosure (Fig. 12). To maintain the IP67 rating of the enclosure, wall-mount connector with 3, 9 and 12 pins have been mounted to assure the connections with the exterior. The boards are 152x82 mm in dimensions with a height of 30 mm, allowing the usage of standard ABS IP67 enclosure.



Fig. 12. Boards with the enclosure

From the hardware point of view, we have analyzed the consumption of the controller without a running algorithm on a range of voltage varying from 8 to 32V DC. We have analyzed the impact on energetic consumption of the Ethernet interface and the WLAN interface and also the influence of the supply voltage on the consumption. From Fig.13 one can observe that the power consumption is lowest at low voltages (at 8-9V and at 13-15V). The power consumption is lower than the limit discussed in chapter IV regardless the communication interfaces that are enabled. The WLAN interface consumes about 0.45-0.55W, while the Ethernet interface increases the power consumption with 0.28-0.35W.

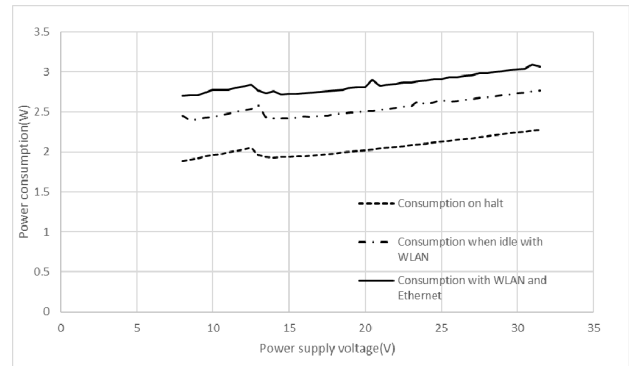


Fig. 13. Power consumption with different communication interfaces without software running

Taking in consideration that the software running on the controllers imply a higher power consumption than in the idle mode, we have analyzed the situation when all the controllers execute the specific algorithms and when the Ethernet, respectively the Wi-Fi are polled by clients.

As one can conclude from Fig. 14, even in full load with all the communication interfaces turned on, the power consumption is lower than 3.5W. Enabling and disabling the communication interfaces have the same impact on power consumption as in the idle state case. The increase of power consumption with all the algorithms running on all controllers is of about 0.45W.

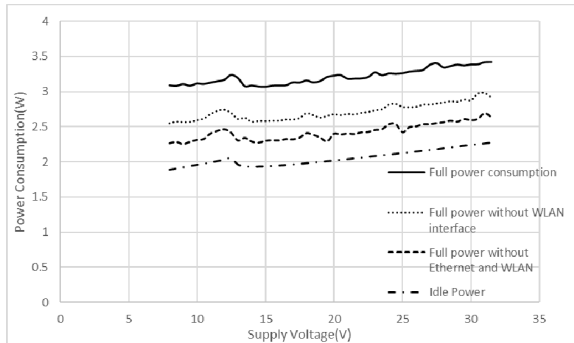


Fig. 14. Power consumption dependency of communication interfaces when software is running

The most complex in the current architecture is the Aria G25 controller that runs a Debian based operating system. Since all the “heavy” software (webserver, communication with controllers, encryption) are running on it, we have studied the usage of processor and memory. In Fig. 15 we observe that most of the processing power (78%) and memory (63%) is used by the Python application (web server, data logging and alarms), and the C application uses about 18% of processor and less than 5% of memory.

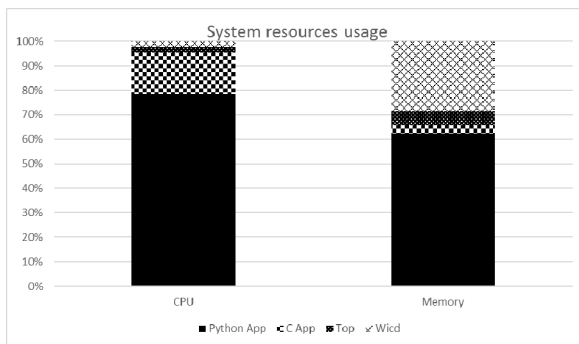


Fig. 15. System usage of software applications running on Aria G25

The controller presented by the authors regardless the conditions has a consumption less than 4W through a large voltage range. The hardware design results in a low height board of 3cm, with Ethernet, Wi-Fi, 20 Digital inputs, 17 Digital Outputs, 24 Analog Inputs, 4 Analog Outputs, 1 one wire interface, and 2 RS485 communication lines compared to heights of 8-10cm or larger of industrial controllers.

ACKNOWLEDGMENT

We would like to thank Enevo Group SRL Romania for supporting us with the PCB manufacturing and electronic components for the device and InGear applied electronics laboratory for supporting us with the instrumentation and testing conditions.

REFERENCES

- [1] Sean Ong, Clinton Campbell, Paul Denholm, Robert Margolis, and Garvin Heath, “Land-Use Requirements for Solar Power Plants in the United States”, Technical Report NREL/TP-6A20-56290, June 2013
- [2] Shaharin Anwar Sulaiman, Atul Kumar Singh, Mior Maarof Mior Mokhtar, Mohammed A. Bou-Rabee, “Influence of Dirt Accumulation on Performance of PV Panels”, The International Conference on Technologies and Materials for Renewable Energy, Environment and Sustainability, TMREES14, pp 50-56
- [3] J. B. Jawale, V. K. Karra, B. P. Patil, P. Singh, S. Singh and S. Atre, “Solar panel cleaning bot for enhancement of efficiency — An innovative approach,” 2016 3rd International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, 2016, pp. 103-108
- [4] M. A. Jaradat et al., “A fully portable robot system for cleaning solar panels,” 2015 10th International Symposium on Mechatronics and its Applications (ISMA), Sharjah, 2015, pp. 1-6
- [5] S. P. Aly, P. Gandhidasan, N. Barth and S. Ahzi, “Novel dry cleaning machine for photovoltaic and solar panels,” 2015 3rd International Renewable and Sustainable Energy Conference (IRSEC), Marrakech, 2015, pp. 1-6
- [6] <http://www.ecoppia.com/>
- [7] D. Căpatina, I. Stoian, T. Sanislav, O. Ghiran, E. Stancel and I. Filip, “Integration Techniques of the Embedded Distributed Systems Using Programming Environments and Industrial Standard Communication Protocols,” 2006 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, 2006, pp. 430-435
- [8] <https://www.phoenixcontact.com/online/portal>
- [9] <http://www.acmesystems.it/aria>