

Software Solution for a Renewable Energy Microgrid Emulator

Eniko LAZAR, Dorin PETREUS, Radu ETZ, Toma PATARAU
 Department of Applied Electronics, Technical University of Cluj-Napoca
 Cluj-Napoca, Romania
 eniko.lazar@ael.utcluj.ro

Abstract—The paper presents a software solution and communication network used to implement and test management algorithms for different microgrid configurations. The results obtained are represented in a Graphical User Interface (GUI). Data is transferred between the devices and a central processing unit that has a communication protocol interpreter implemented using an RS-485 network. After being interpreted, the raw data containing useful information for the management algorithm is converted in numerical or Boolean values. These values are stored and used by the management algorithms implemented for cost optimization. A microgrid emulator is used for the proof of concept. The software solution and communication network together with the data interpreter can be easily used for other microgrid structures with or without modifications, depending on the number and types of equipment used. In the presented case the microgrid management control algorithm tries to keep the state of charge of the batteries between two values by using efficiently the available resources depicted by: solar energy, geothermal energy, and energy obtained from biomass. The microgrid setup has emulators for the geothermal and biomass generators and a photovoltaic system with storage capability and two inverters, a grid forming capable and a grid follower.

Index Terms—energy management, graphical user interfaces, microgrids, renewable energy sources, SCADA systems.

I. INTRODUCTION

The most used energy resources in the world are the fossils fuels such as coal, oils and natural gases. Their natural reserve heavily decreased due to the demographic growth and the technological devices diversity used every day. Furthermore, the energy consumption of the world is increasing about 2.3% every year. In addition, using conventional fuels seriously damages the environment. Thus, it became needed to involve alternative energy resources in the global energy market [1-4].

Renewable energy resources such as wind, sunlight, geothermal and biomass are mostly used. They are working together and their integration into the energy market can improve the sustainability and reliability of the power systems [5, 6]. A microgrid is an autonomous electric distribution system which combines one or more energy resources with the loads, has its own management and control system and works as an independent controllable entity [7]. It can operate in one of two ways: connected to the main grid—grid connected mode or disconnected from the main power grid— islanded mode. Its generation capacity varies from some kW to MW. Off-grid microgrids are

recommended for remote areas where the main grid was not built or available with acceptable price [8, 9].

The proper energy production of the microgrids is also a substantial issue. The balance between the energy flow and the load demands is their basic rule considering the availability of the resources. Numerous structures of microgrids with different energy management algorithms have been mentioned in literature [10-12].

A properly functioning energy management system can ensure the best solution and meet the load requirements continuously and in short time. The gradient-based systems are too slow to be used in real-time energy management systems, so the articles from this area focus on the off-line applications. Reference [13] is a previous study which presents an optimal scheduling of an energy system by using the mixed-integer programming language and the genetic algorithm. Other strategies are based on fuzzy logic and particle swarm optimization [14-16]. Several studies have been reported with tested energy management algorithms in real experimental microgrids [17-20] but the communication aspects are treated with less importance or not at all. This paper treats the communication network aspects and is shown that with a proper design the problems that may occur can be eliminated.

The experimental platform in this paper it's structured in three main parts: a communication network, a control system and a power microgrid. The main contributions and novelties are listed as follows: 1) To propose a new, fast and flexible communication environment based on the inverters' and generators' telegram format and communication protocol; 2) A user-friendly graphical interface with data representation; 3) Experimental testing and validation of the proposed communication system in a real microgrid based on energy producing emulators.

The remainder of this paper is structured as follows: Section 2 briefly describes the architecture of the microgrid used for testing the communication network and software solution. Section 3 outlines the functionality of the communication system. Section 4 presents the proposed GUI with the used functions. Section 5 and 6 contain the results and conclusions.

II. MICROGRID ARCHITECTURE

The tested single phase 230V/50 Hz islanded microgrid has four parallel connected generating units: a geothermal emulator, a biomass emulator, a photovoltaic system (twelve 250 W photovoltaic panels with a Sunny Boy SB36000TL21 inverter) and an energy storage device (eight

VRLA 12 V-200 A/h batteries with a Sunny Island 6.0 grid forming capable battery inverter).

The emulator for the geothermal generator incorporates an induction motor and a synchronous generator. The induction machine is controlled by an inverter, and the synchronous generator is led by the induction motor. The construction of the biomass emulator is based on the same concept, except for an asynchronous generator, which operates as a generator.

The master grid forming generator is the geothermal emulator. This master generator sets the frequency and the voltage of the grid. In case the geothermal generator is unavailable, the battery inverter takes its place and performs the grid forming tasks. The solar inverter is a Sunny Boy SB36000TL-21 (SB) inverter.

The SCADA (Supervisory, Control and Data Acquisition system) is an essential part of the microgrid which is responsible for the communication and the control energy management algorithms, to develop an energetic framework.

III. COMMUNICATION SYSTEM

This system has a two-level control structure. The first control level encompasses the voltage versus reactive power (V/Q) and the power versus frequency (f/P) droop characteristics of the generators and inverters. Voltage adjustment is a vital issue for the reliability of the grid [18].

The secondary level management control algorithm ensures the balance between the powers produced and the loads. This secondary level is based on an RS-485 network Fig. 1. The communication protocols used are equipment dependent, but a software interpreter is used to convert the useful data. Due to this interpreter different equipment with their own communication protocols can be included into the microgrid, i.e. SMA devices and Modbus standard capable devices.

A bidirectional communication is established between the control GUI and the connected units in the test grid: SMA inverters Sunny Island and Sunny Boy, two Modbus to Voltage Converters used for controlling the biomass and geothermal emulators and Modbus capable Power Meters.

A. Modbus compatible units

The geothermal and biomass emulators have Modbus compatible devices that can send or receive commands and power meters. Using the data provided by these devices a management algorithm can decide the actions that need to be taken to reduce the energy production costs and can control the power delivered by each device into the microgrid. The telegram frame for the mentioned devices is presented in the Table I. Each device has a unique network address i.e. 0x01 for the geothermal emulator and 0x03 for the biomass emulator. Based on this address, the register address, that has to be modified or read, and the number of bytes needed for data representation, a telegram is constructed and sent via RS-485.

For example, to read the biomass emulator output power the requested data are: address – 0x02, register address – 4140, and register quantity (no. of bytes) – 4. These data depend on the equipment used but the software solution permits easy integration of any Modbus capable device.

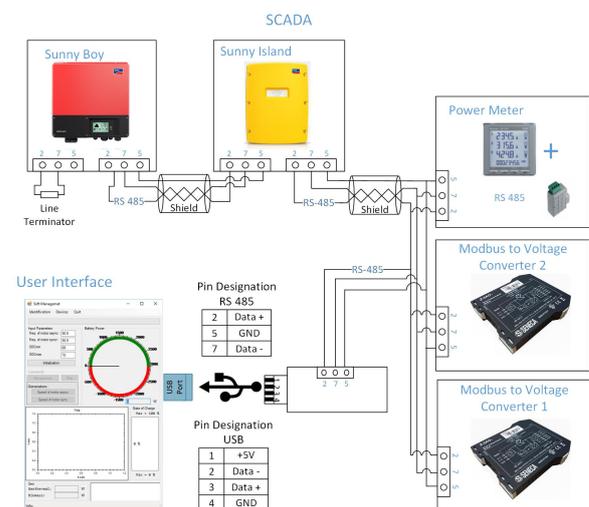


Figure 1. The proposed RS-485 communication network

It is important to note that these emulators can be replaced with others and the microgrid can be expanded with additional emulators too, scalability being one of the advantages of this proposed solution.

B. SMA bus compatible units

The SI and SB inverters use a modified Modbus proprietary communication protocol, called SMA Net. The identification of the contents of SMA messages was based on SMA Data Specifications and through reverse engineering using RS-485 port listeners (sniffers) and analyzers.

The telegram frame of SI and SB inverters is presented in Table II. This telegram frame encompasses the actual contents of the Data (min 7, max 262 bytes). The length of the telegram is variable and depends on the respective command. The start, address, control, protocol header and stop bytes have predefined values. For error control, a 16-bit Frame Check Sequence (FCS) is calculated. The first *start byte* indicates the start of a new message transmission, the *next same byte* is the stop byte. If the address is set to 0xFF, then the message arrives to the all participants in the network. The inverters have the responsibility to decide if the message is addressed to them based on the received device address (destination address) from the encapsulated Data and prepare the answer if it's the case. To accomplish a transparent transfer, neither the start nor the stop characters can appear in the middle of the telegram, for this an escape character is inserted. This character is followed by the value which is the result of the operation exclusive OR between the special character and 0x20. The 4th row in Table II contains the values used for testing the communication.

Data Content (Table III.) also has two parts. The protocol header includes the source and destination address of the telegram, a control byte to handle the packets, a packet counter byte (PckCnt) and the actual command byte. The control byte is 0x80 in case of transmitted messages and 0x40 in case of received message. Different commands can be implemented by modifying the value of the Cmd byte. The User Data maximum length is 255 characters, but this length depends on the command. If the telegram contains more characters than the maximum length, the data will arrive in sequential packets. The actual number of the sequential packet is marked in the PckCnt.

TABLE I. MODBUS STANDARD TELEGRAM

Device address	Functional code	Data	Checksum
1 byte	1 byte	4 bytes	2 bytes

TABLE II. TELEGRAM FRAME

Frame			Content		Frame	
Start	Address	Control	Protocol header	Data	FCS	Stop
1 byte	1 byte	1 byte	2 bytes	min 7, max 262 bytes	2 bytes	1 byte
0x7E	0xFF	0x03	0x4041			0x7E

TABLE III. DATA CONTENT

Protocol Header					User Data
Source	Destination	Ctrl	PktCnt	Cmd	Data
2 bytes	2 bytes	1 byte	1 byte	1 byte	max. 255 bytes
Sum: 7 bytes					Sum: max 255 bytes

Two categories of the main commands were used based on the presented telegrams:

1) *Commands for System Configuration*

This chapter presents the commands which provide the registration and configuration of the system. The presented software must use these commands in a preestablished order to identify the system properly. First, a network configuration request is transmitted on the RS-485 for the detection of the active SMA devices connected in the microgrid. The devices, that can be reached, respond directly to this request by giving their name, serial number and actual network address. The GUI saves the recognized device's data. The second command assigns a unique network address to the detected devices. These steps repeat until all devices are identified and registered.

To fulfill the control over the system, it is essential to have all the necessary information about the channels of the inverters. The third type of command sends a request and all devices responds with a list of own memory channel information. The structure of a channel divides into two parts: a type independent header (Table IV.) and a type specific extension (Table V.).

Each channel has a name, an index, a channel type and data format. A channel can be analog, digital, counter or status. Besides this they can be classified as input, output, parameter, instant values, archive data and test channel.

The sum of bytes in Table IV. is 23, but the length of one channel varies depending on its specific extension.

In case of SI, 350 channels were identified: 48 counters, 90 status type and 212 analogs. The necessary channels for the management algorithm used in this case are the following:

- State of charge of the batteries (*BatSoc*): Index = 14, Channel Type = Analog and Instant&Input, [%]
- The battery power (*PacSI*): Index = 55, Channel Type = Analog and Instant&Input, [kW]
- The power of the geothermal generator (*ExtPwrAt*): Index = 38, Channel Type = Analog and Instant&Input, [kW]
- The Geothermal Generator Status (*GnManStr*) – it indicates if the geothermal generator is connected or not: Index = 38, Channel Type = Analog and Parameter, Status = "Auto_Stop_Start".

TABLE IV. TYPE INDEPENDENT PART

Name	Description	Type	Value
Index	The current channel number of this channel type	byte	1...FF
Channel Type	0000 0000 0000 0000 Bit: F... .. 3210 Signal Type: – 0: Analog – 1: Digital – 2: Counter – 3: Status – 4-7: Reserved Signal Group: 8: Input; 9: Output; A: Parameter; B: Instant Value; C: Archive Data; D: Test channel; E-F: Reserved	word	
Data Format	F... .. 3210 0000 0000 0000 0000 Array Reser 0000 Byte Depth ved 0001 Word 0010 Dword 0100 Float 1xxx Array	word	
nFill		word	
Channel name	Channel name – text	char 16	
	SUM	23 bytes	

TABLE V. TYPE SPECIFIC EXTENSION

Type	Description	Sum of bytes
Analog	Unit: Unit of the channels – char 8 Gain: Gain of the channel – float4, Offset: float4	16
Digital	TxtLo: Text Signal = 0 – char 16 TxtHi: Text Signal <>0 – char 16	32
Counter	Unit: Unit of the channels – char 8 Gain: Gain of the channel – float4	12
Status	SizeT: Length of the following status string - word StatT: Status text – dynamic	2+Size T

The SB inverter has 393 channels: 60 status type, 137 counters and 196 analogs. The required channel is the *PacSB* which is the solar power. Its index is 10, its type is Analog (Instant value & Input), it is measured in [W].

2) *Commands for Data Acquisition*

These commands requests and/or sets the data of the SMA devices. To treat the memorized channels properly, it is necessary to build a transfer mask and transmit it. This mask communicates the selected types of channels, which will be requested or modified. It works as a filter; all the selected channels are affected by the command. It has two parts: a mask for channel type (2 byte) and the channel index (1 byte).

Data Request

Before each request cycle, a synchronization command must be sent to all participants, thereby simultaneous acquisition of online data can be guaranteed. This message is followed by the data request.

I.e. for reading *BatSoc*, *Pac* and *ExtPwrAt* from the SI inverter the transfer mask used for channel type is: 0000 1001 0000 0001 – analog signal type, Instant and Input (bit significance is according to Table IV, channel type). The channel index is set to 0, so the response contains the all channels that are selected in the mask. The control interface chooses from the above listed channels' values, depending on the incoming answer, considering their index and

position in the memory.

SB inverter has an additional function, which allows reading only one channel's value by setting the channel index (different value than 0x00) in the transfer mask.

Send Data

The digital outputs, counter contents, parameters and status signals can be varied. First it is recommended to request the channel which will be modified, then set its value and request again the channel value to after the value is modified. In this way, the pertinence of the action can be checked. Just as in the case of data request, a transfer mask is also needed.

IV. ENERGY MANAGEMENT CONTROL INTERFACE

The communication and control responsible device collects the data from the microgrid and manages the system. The algorithm used to test the system is based on the state of charge of the batteries. The read values are the following: *BatSoc*, *PacSI*, *ExtPwrAt*, *GnManStr*, *PacSB* and *Pbio* (power of the biomass generator).

A. Graphical User Interface

The GUI is built in Microsoft Visual Studio C# and presented in Fig. 2. The user must define two *SOC* (state of charge of the batteries) considering the actual *SOC* - *BatSoc*: one for the lowest value (*SOCmin*) and one for the highest value (*SOCmax*). An energy management control algorithm verifies these three *SOC* values and handles the generator units holding the *BatSoc* between the limits. The frequencies of the generators are also user dependent values.

The energy management process begins with the detection of the connected devices. It is a mandatory step, after starting the software application. The GUI must acquire all the necessary data of the detected devices (name, id, address) to complete the next steps. Here are two possibilities: *Detection* through the communication system or *Detection* from File. The first option takes about ten minutes, the devices' channel list is discovered by the presented commands in chapter III. The channel list is saved as a text document at the end of the channel discovery step. The second less consuming time option uses prior saved text document to identify the channel list.

Afterward, the application waits for pressing of the *Management button*, and the recursive process begins.

B. Software classes and functions

The presented software contains three classes: a class for the modbus standard - *modbus.cs*, a class for the SMA protocol - *SMAbus.cs*, and a class for holding the information and description about the detected devices - *Device.cs*. The *modbus.cs* class includes two main functions: *SendFc3* and *SendFc16*, which transmit the request message for writing and reading registers.

The transmitted messages are locally built. Additionally, functions CRC computation, Check Response, and Get Response are used. The architecture of *SMAbus.cs* is very similar, except there is a function (*Build_Message_SMA*) which is liable to piece the request messages properly together – in case of SMA devices there are 3 types of commands: system configuration commands, get data and set data commands. The *Verification* function checks the

special characters in the message and changes them according to Section III. The *SendMessageSMA* function transmits the request and saves the responses in a list.

The name, serial number and channel list of SMA devices (table IV and V) are saved in the *Device* class. Two objects are used: *deviceSI* and *deviceSB*.

The main function uses the presented classes to achieve the control of the energy production and transfer into the microgrid. A short description about the functions used is found in Table VI.

C. Control algorithm

The architecture of the proposed software application together with the energy management control algorithm is presented in Fig. 3. The first part of this flowchart focuses on the System Configuration: detection of the connected SMA devices and their first-time start-up settings. This procedure is presented as follows: the geothermal generator is disconnected by using the *GnManStr* channel (the value of this channel can be 1-AutoRun, 2-Off, 3-On) and set using the *SMAbus* class and its specified functions; the biomass generator is set to standby mode and its availability can be ensured by the offered functions of *modbus* class.

The application reads and saves the current values of solar energy, generators' output powers, the batteries' charging/discharging values and state of charge, and adjust the energy management decisions considering the listed values and the initial *SOCmax* and *SOCmin*. The control management algorithm verifies the allowed maximum and minimum charge of the batteries, and, by using cycle charge, controls the generators depending on their availability [18].

V. RESULTS

The device responsible with the communication and proposed software solution GUI are presented in Fig. 5. The data presented are derived from a test using the renewable energy microgrid.

Fig. 4 is split in 5 sections used for a better explanation of the GUI interface:

Section A includes the Input values: the frequency of the emulators and the minimum and maximum states of charge used to implement a cycle charge energy management algorithm. The user can modify these values during the runtime, the application takes into consideration the new values and modifies the algorithm in real-time.

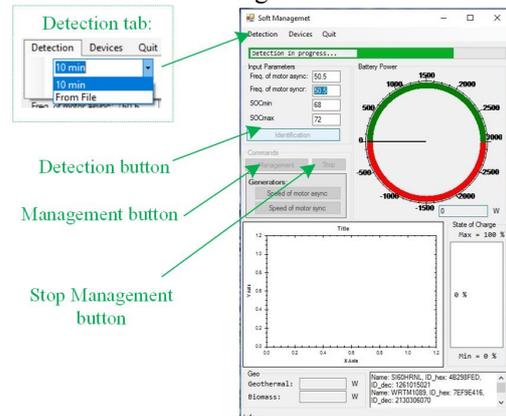


Figure 2. GUI Interface

TABLE VI. FUNCTIONS

Name	Description	Location
bool Open (string portName, int baudRate, int databits, Parity parity, StopBits stopBits); public bool Close ();	Open/Close the serial port	Modbus.cs, SMAbus.cs
void GetCRC/GetFCS (byte[] message, ref byte[] CRC); bool CheckResponse (byte[] response);	CRC/FCS Computation Check the response	Modbus.cs SMAbus.cs
private void GetResponseSMADATA (ref List<byte> response)	It reads a byte stream of set length from the serial port and places in this result into the response array.	SMAbus.cs Modbus.cs
bool SendFc3/SendFc16 (byte address, ushort start, ushort registers, ref byte[] values...);	Write/Read Registers	Modbus.cs
Boolean SendMessageSMA (ref List<byte> values, ushort netadr, ref byte pckCounter, byte[] serial_nr, byte idx, byte adcs, byte iopi, string dataformat, float valuetowrite)	Adcs: analog/ digital/ counter or status Iopi: Input/ Output/ Parameter/ Instant Value Dataformat: byte, word, dword, float, array	SMAbus.cs
void BUILD_Message_SMA (ref List<byte> message, ushort netadr, byte[] serial_nr, byte pckCnt, byte adcs, byte iopi, byte idx, string dataformat, int command,...);	Build the request message depending on the command (configuration, set network address, get & set data)	SMAbus.cs
void btnDetection_Click (object sender, EventArgs e);	Serial number and device name detection	MainForm
void IdentificareCanalelor ();	Channel list detection	MainForm
void btn_Management_Click (...)	Start the management algorithm;	MainForm
void ReadNemo (ref float NemoP, byte address, ushort StartAddress, ushort RegisterQuantity)	Read the Biomass Power: <i>Pbio</i>	MainForm
void SetFrequency (string s, float frec);	Set the frequency of geothermal/biomass generator	MainForm
void SendMessage (int command, string device, float valuetowrite, ... , ref float[] results)	System Configuration + Read BatSoc, PacSI, ExtPwrAt + Read/Write GnManStr;	MainForm
void GetBytesToBuildOneCh (string device, ref byte adcs, ref byte iopi, ref byte idx, ref int id, ref int poz, ref string dataformat, string name);	Return the requested channel's id, types, position, dataformat	MainForm

Section B Battery Power Circle chart represents the batteries' power (*PacSI*). The red negative value means that the generated energy is less than the load demand. In this case the batteries are discharging. The positive green value indicates that the energy generated by the active generators satisfies the load demand, the excess energy is used to charge the batteries. The numeric value of the SI power is also available below the chart.

Section C is the battery State of Charge Bar Graph. This value with *PacSI* is used for the control algorithm based on batteries cycle charge method.

Section D is part of the output values. After detecting the connected devices, their names and id appear in the text box. In that moment when the management starts, this information will be replaced with the power of solar, geothermal, biomass and batteries.

All the data are saved in a txt file with the current date's name (Report-current date.txt), so they can be accessed and verified in hindsight. This text document is useful to create different statistical charts.

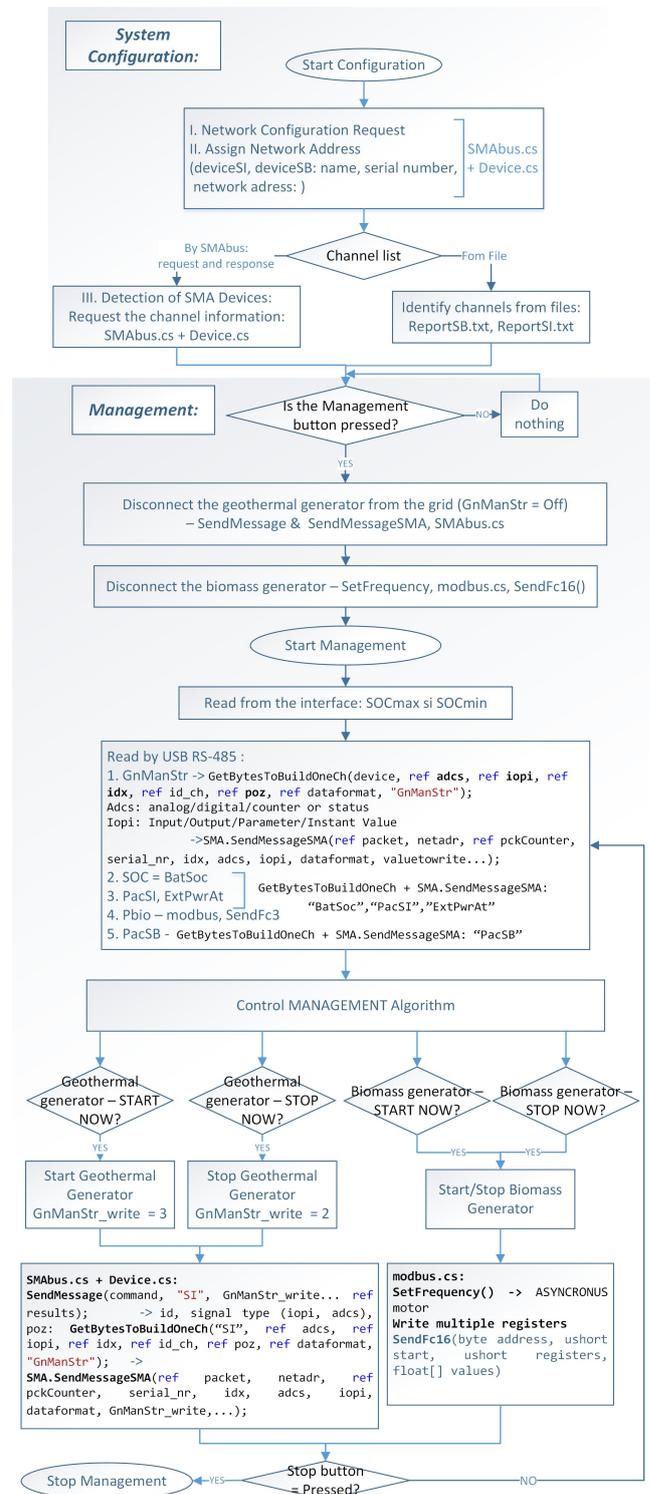


Figure 3. System Configuration and Control Algorithm

The presented interface allows new generators and inverters to connect to the existing microgrid, under the condition that they are compatible with the presented communication protocols. If the newly added devices have different communications protocols small adjustments must be performed considering that the presented functions are general. Furthermore, it is possible to build an entirely new microgrid. Moreover, the interface can be expanded with additional graphs and displays.

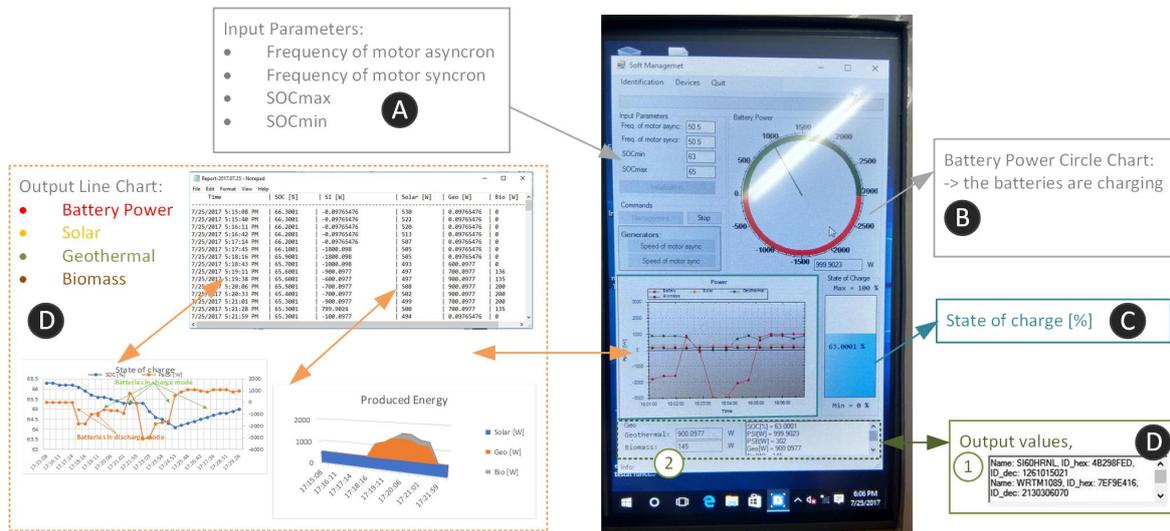


Figure 4. Results

VI. CONCLUSIONS

The proposed user-friendly GUI establishes a transparent communication between the connected devices in the tested microgrid, requests or modifies the output powers and states of charge of the batteries, starts/stops the connected generators, displays the collected data and achieves an energy management algorithm. A communication protocol interpreter is also implemented. New generators and inverters can connect to the system with the condition that they are compatible with the presented communication. The interface is expandable with newer charts, graphics, progress bar and text boxes. Also, other management algorithms can be added. Additionally, further work is necessary to find the optimal scheduling of the participants of the microgrid, to decrease the functional cost of the system, to use the generated energy consciously.

REFERENCES

- [1] S. M. Dawouda, X. Lin, M. I. Okbaa, "Hybrid renewable microgrid optimization techniques: A review," *Renewable and Sustainable Energy Reviews*, Vol. 82, Part. 3, pp. 2039-2052, 2018. doi:10.1016/j.rser.2017.08.007
- [2] A. G. Lupu, A. Dumencu, M. V. Atanasiu, C. E. Panaite, Gh. Dumitrascu, A. Popescu, "SWOT analysis of the renewable energy sources in Romania – case study: solar energy", 7th International Conference on Advanced Concepts in Mechanical Engineering, IOP Conf. Series: Materials Science and Engineering Vol. 147, pp. 1-10, 2016. doi:10.1088/1757-899X/147/1/012138
- [3] M. Javad Kasaei, M. Gandomkar, J. Nikoukar, "Optimal management of renewable energy sources by virtual power plant", *Renewable Energy*, Vol. 114, pp. 1180-1188, 2017. doi:10.1016/j.renene.2017.08.010
- [4] M. L. Tuballa, M. L. Abundo, "A review of the development of Smart Grid technologies", *Renewable and Sustainable Energy Reviews* Vol. 59, pp. 710-725, 2016. doi:10.1016/j.rser.2016.01.011
- [5] E. Mengelkamp, J. Gartner, K. Rock, S. Kessler, L. Orsini, C. Weinhardt, "Designing microgrid energy markets: A case study: The Brooklyn Microgrid", *Applied Energy*, Vol. 210, pp. 870-880, 2017. doi:10.1016/j.apenergy.2017.06.054
- [6] M. Smaoui, L. Krichen, "Control, energy management and performance evaluation of desalination unit based renewable energies using a graphical user interface", *Energy*, Vol. 114, pp. 1187-1206, 2016. doi:10.1016/j.energy.2016.08.051
- [7] I. González, A.J. Calderón, J.M. Andúj, "Novel remote monitoring platform for RES-hydrogen based smart microgrid", *Energy Conversion and Management*, Vol. 148, pp. 489-505, 2017. doi:10.1016/j.enconman.2017.06.031
- [8] M. E. Khodayar, "Rural electrification and expansion planning of off-grid microgrids", *The Electricity Journal*, Vol. 30, pp. 65-74, 2017. doi:10.1016/j.tej.2017.04.004
- [9] T. Adefarati, R.C. Bansal, "Reliability and economic assessment of a microgrid power system with the integration of renewable energy resources", *Applied Energy*, Vol. 206, pp. 911-933, 2017. doi:10.1016/j.apenergy.2017.08.228
- [10] H. Morais, P. Kadar, P. Faria, Z. A. Vale, H. M. Khodr, "Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming", *Renewable Energy*, Vol. 35, pp. 151-156, 2010. doi:10.1016/j.renene.2009.02.031
- [11] H. Ren, W. Gao, "A MILP model for integrated plan and evaluation of distributed energy systems", *Applied Energy*, Vol. 87, pp. 1001-1014, 2010. doi:10.1016/j.apenergy.2009.09.023
- [12] P. Garciaa, J. P. Torreglosa, L. M. Fernández, F. Juradoc, R. Langella, A. Testa, "Energy management system based on techno-economic optimization for microgrids", *Electric Power Systems Research*, Vol. 131, pp. 49-59, 2016. doi:10.1016/j.epr.2015.09.017
- [13] E. Lazar, D. Petreus, R. Etz, T. Patarau, "Minimization of operational cost for an Islanded Microgrid using a real coded Genetic Algorithm and a Mixed Integer Linear Programming method", 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), Brasov, Romania, 25-27, May, pp. 693-698, 2017. doi:10.1109/OPTIM.2017.7975049
- [14] X. Lu, K. Zhou, S. Yang, "Multi-objective optimal dispatch of microgrid containing electric vehicles", *Journal of Cleaner Production*, Vol. 165, pp. 1572-1581, 2017. doi:10.1016/j.jclepro.2017.07.221
- [15] M. S. Fakhar, S. A. R. Kashif, M. A. Saqib, T. Hassan, "Non cascaded short-term hydro-thermal scheduling using fully-informed particle swarm optimization", *Electrical Power and Energy Systems*, Vol. 73, pp. 983-990, 2015. doi:10.1016/j.ijepes.2015.06.030
- [16] C. Roldan-Blay, G. Escrivá-Escrivá, C. Roldan-Porta, C. Alvarez-Bel, "An optimisation algorithm for distributed energy resources management in micro-scale energy hubs", *Energy*, Vol.132, 126-135, 2017. doi:10.1016/j.energy.2017.05.038
- [17] M. Marzband, A. Sumper, A. Ruiz-Álvarez, J. L. Domínguez-García, B. Tomoiaga, "Experimental evaluation of a real time energy management system for stand-alone microgrids in day-ahead markets", *Applied Energy*, vol. 106, pp. 365-376, 2013. doi:10.1016/j.apenergy.2013.02.018
- [18] R. Etz, D. Petreus, T. Patarau, E. Lazar, "An islanded renewable energy microgrid emulator for geothermal, biogas, photovoltaic and lead acid battery storage", 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, pp. 2109-2114, 19-21 June 2017. doi:10.1109/ISIE.2017.8001583
- [19] M. Marzband, E. Yousefnejad, A. Sumper, J. Luis Domínguez-García, "Real time experimental implementation of optimum energy management system in standalone Microgrid by using multi-layer ant colony optimization", *Electrical Power and Energy System*, Vol. 75, pp. 265-274, 2016. doi:10.1016/j.ijepes.2015.09.010
- [20] M. Elsied, A. Oukaour, T. Youssef, H. Gualous, O. Mohammed, "An advanced real time energy management system for microgrids", *Energy*, Vol. 114, pp. 742-752, 2016. doi:10.1016/j.energy.2016.08.048