International Conference on Applied Internet and Information Technologies, 2016

DOI:10.20544/AIIT2016.13

An IoT Concept of the Small Virtual Power Plant Based on Arduino Platform and MQTT Protocol

Dmytro Zubov

School of Engineering and Sciences, University Tecnológico de Monterrey
Av. Eugenio Garza Sada 300, Lomas del Tecnologico, 78211 San Luis Potosi, S.L.P., Mexico
dzubov@ieee.org

Abstract. In this paper, a new IoT concept of the small virtual power plant is proposed. The main principle is the smart control of the high-voltage power lines by the low-cost and low-voltage Arduino platform with Mosquitto open-source message broker that implements the secure TCP connection based on MQTT IoT protocol using specific topics, passwords, and SSL cryptography. The C# Windows form app and Arduino sketch were developed with MQTT publisher and subscriber, respectively. The hardware simulation of the proposed concept is realized using the solar panel (12.5 V, 180 mA), two relays SRD-05VDC-SL-C, buck (DC/DC YEC SD200) and buck/boost (DC/DC Tusotek TS-IPS-VO2) converters, and two small power banks (first one is to supply the consumption part simulated by Arduino Uno with LED, second one is to supply the smart power distribution block). The hardware was tested successfully in three regimes: the charge of battery by the solar panel, the power supply of the Arduino Uno board with LED using the power bank or solar panel. The proposed infrastructure can be directly applied to the power off-grid solutions of the smart houses if grid-tie inverter is added as well as generation and storage parts are more powerful.

Keywords: virtual power plant, IoT, Arduino, MQTT, C#.

1. Introduction

Nowadays, the virtual power plant (VPP) is the leading concept to manage different types of the renewable (solar, wind, biogas, geothermal mainly), classical (e.g. coal, gas, and hydroelectric), and nuclear energy resources [1, 2]. This synergetic approach is based on the smart power grids with distributed generation facilities and possibility to control the consumption using different levels of the power quality and reliability. Currently, many companies (e.g. Bosch, Schneider Electric, Siemens, and RWE) propose industrial-scale solutions for VPP. However, the small-scale VPPs (e.g. smart house with off-grid electric power system) with unified soft- and hardware are not well represented in literature and Internet. For instance, the simulation of the small VPP in MATLAB 7.6 is presented theoretically in [3]. In [4], the microrenewable sources of energy and electric vehicles are discussed as the parts of a microgrid concept. In [2], the structure, application, and optimization of VPP are reviewed in general. In addition, the budget solutions for VPPs are discussed conceptually, as well as not the whole system but some parts of it are proposed mainly nowadays. For instance, Arduino Mega board is used for the control and monitoring of the wind turbine in [5]. In [6], the high-precision solar algorithms are implemented by Arduino platform. Other example is the solar and wind battery charging regulator processed by Arduino board (http://www.herbandbarbara.com/index_files/Page4001.htm).

Over the past fifteen years, the VPP together with Smart Energy application domain have been developing within the Internet of Things (IoT) methodology because the IoT soft- and hardware, as well as algorithms are applied directly to VPP [7]. The well-known IoT platforms are Arduino/Genuino and Raspberry Pi [8]. However, the price of Raspberry Pi is several times greater than Arduino analogues usually. The Arduino Uno run-time is 50 days approximately (http://forum.arduino.cc/index.php?topic=191416.0). Arduino is used in industry (e.g. the industrial microcontroller Industruino is a fully featured Arduino Leonardo compatible board; https://industruino.com). Hence, Arduino platform is used widely in IoT. In addition, new WiFi Arduino-compatible development boards like NodeMcu Lua ESP8266 ESP-12E (https://smartarduino.gitbooks.io/user-manual-for-esp-12e-devkit/content/index.html) are discussed for the wireless connection to the sensors using web-server. In this case, Arduino IDE is used for the software development. Different sensors and actuators [9] provide the interaction among the physical objects. Then, the hardware implements the appropriate algorithm(s) to control the VPP (e.g. the multi-agent system [10]). It should be mentioned that is not the scope of this paper to define the optimal algorithm. The author proposes the infrastructure as a service with the focus on hard- and software realization of the control methodology. In the following, the Arduino sketches include the control algorithms.

Different IoT devices are connected by special type of IoT protocols because of real-time communication requirement, low performance of IoT hardware, as well as Internet connection is slow sometimes. Nowadays, two IoT protocols, MQTT and CoAP, are in use mainly (https://eclipse.org/community/eclipse_newsletter/2014/february/article2.php; https://www.linkedin.com/pulse/iot-communication-protocols-james-stansberry). First one is based on the Internet TCP (Transmission Control Protocol), second one – on the UDP (User Datagram Protocol). The viable alternative of the IoT protocol depends on the project. If a message is going to be published from one node to many nodes, the MQTT protocol is recommended to use. In the systems with the traffic limitations, CoAP is recommended because it uses UDP, which eliminates the overhead of TCP/IP. It makes a big difference in traffic if system has 1000s of nodes. In the following, the MQTT protocol is applied to transmit/receive the data because of the author’s background, feasibility to provide the secure connection among clients/broker, and the correspondence with the multi-agent methodology [10], where MQTT clients represent the agents.

Based on the above-stated brief analysis of the previous studies, this paper main goal is to discuss a new IoT concept of the small VPPs. The main principles are as follows:

1. The power grids’ control based on the low-cost reliable Arduino open-source soft- and hardware.

2. The high-voltage power lines' control based on the low-voltage devices like the relay module SRD-05VDC-SL-C for Arduino (output voltage is up to 250 V AC).

3. Communication between the different parts of the small VPP based on MQTT IoT protocol using specific topics.

In the following, the secure TCP connection using the MQTT protocol's passwords and Secure Sockets Layer (SSL) cryptography are not in use because of the small scale of prototype and the isolated network.

This paper is organized as follows: In Section 1, a new IoT concept is proposed for the small VPPs based on the analysis of the previous studies. In Section 2, an infrastructure of the small VPP is presented using the consumption (Arduino Uno with light-emitting diode, LED), generation (solar panel), and storage (power bank) components. In Section 3, the small VPP software for the supervisory control and data acquisition (SCADA) is discussed based on the MQTT protocol for the communication among the physical objects. Conclusions are summarized in Section 4.

2. The Infrastructure of the Small Virtual Power Plant

Classical infrastructure of the VPP consists of three main parts, generation, consumption, and storage, which are connected by bidirectional (depends on the role – generation or consumption) power lines. The proposed integration of renewable energy sources and consumers is similar to presented in [11] idea, where the connecting/disconnecting of these sources are made via the appropriate contactors, automatically (or manually) controlled according to the energy consumption/generation. The structure of the small VPP is presented in Fig. 1. The main distinction is the control of the consumption part. In the system, the roles (generation or consumption) of the components may be changed vice versa. Smart power distribution block (SPDB) is responsible for the realization of commutation between the generation and consumption parts. This communication is described by the states of the connectors (i.e. relays) KS_1 – KS_N and KL_1 – KL_M, where N is the number of the KS connectors between energy sources and SPDB, M is the number of the KL connectors between energy sources and SPDB. SPDB can be realized based on Arduino Uno and Ethernet Shield or NodeMcu Lua ESP8266 ESP-12E WiFi development board. Energy source and load controllers are the buck/boost converters or/and the grid-tie inverters. SCADA app controls the system in general, as well as it stores and represents the info for the end-users. For instance, Python and C# programming languages can be in use for the development of the web and desktop apps with MQTT protocol support, respectively.

An example of the small VPP's hardware is presented in Fig. 2. The generation part is based on the solar panel (12.5 V, 180 mA). The consumption part is based on the Arduino Uno with LED and Ethernet Shield. Buck (DC/DC YEC SD200 6-24V to 5V 3A USB step-down stabilized voltage supply module) and buck/boost (Tusotek TS-IPS-VO2 6-35V to 1-35V DC/DC buck/boost power converter module) converters are the controllers: first one converts 12.5 V voltage of the solar panel to 5 V to charge the battery of the load part; second one converts 12.5 V voltage of the solar panel to 10 V to supply the consumption part. Relays are connectors: first one controls the buck/boost converter; second one controls the buck converter. If relay 1 is on and solar panel has enough power, the battery is charged (if solar panel does not provide enough power, battery supplies the consumption part); in this case, the power bank does not supply the consumption part because of the specificity of the power bank's voltage management. If relay 2 is on, the solar panel supplies the consumption part. If relay 2 is on and relay 1 is off, the Arduino Uno with LED is supplied by the solar panel only because of the specificity of the board's voltage management. Because of the small current (180 mA) of the solar panel, the battery can be charged together with the power supply of the consumption part in this small VPP project if the battery is almost charged only (i.e. the charge current is almost zero). SPDB changes the states of the relays according to the SCADA commands sent by MQTT protocol. Here, the Arduino fault tolerance can be improved based on the parallel power and signal lines.

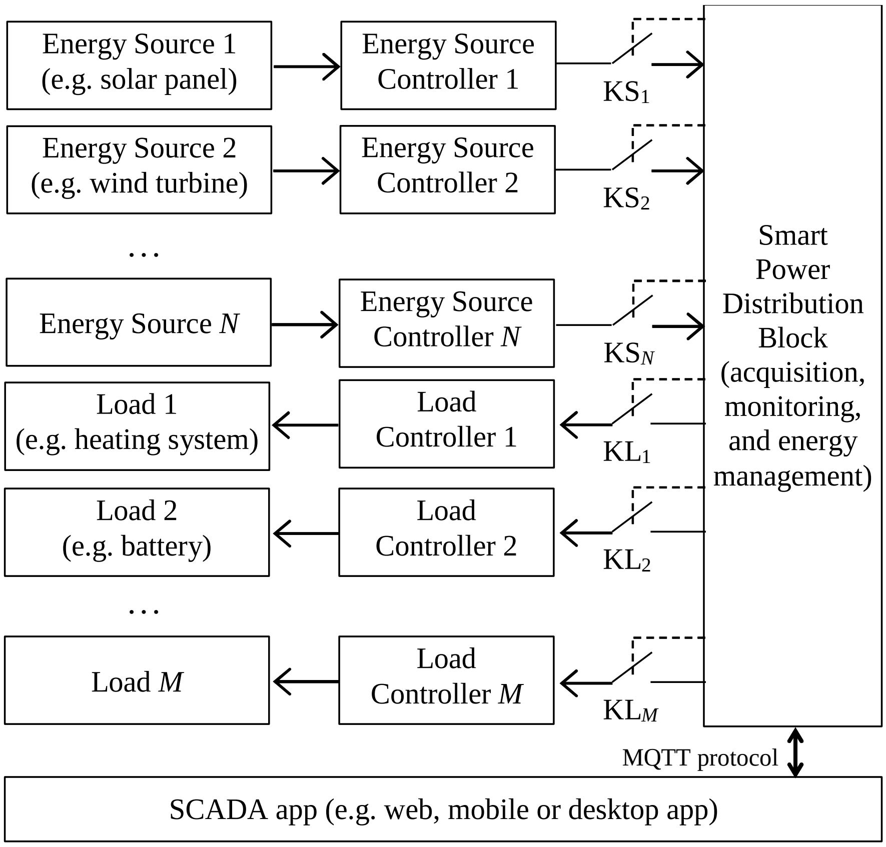

Fig. 1. The functional blocks scheme of the small VPP

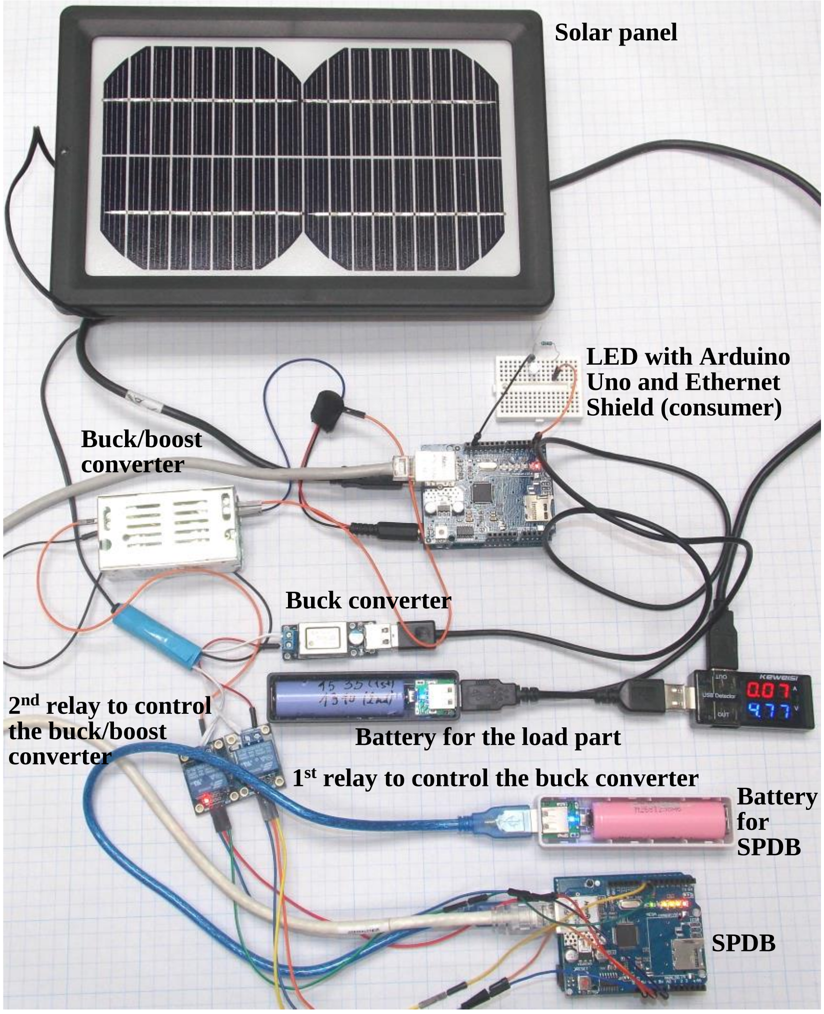

Fig. 2. An example of the small VPP hardware part

3. Small VPP Software for the Supervisory Control and Data Acquisition Using MQTT protocol

The SCADA software has two parts, low-level back-end (i.e. SPDB or microcontroller; Arduino sketch here) and high-level front-end (C# Windows form app here). They are connected based on the open-source message MQTT broker Mosquitto (http://mosquitto.org/). Hereafter, the programming code is discussed in detail with purpose to simplify the development of the similar VPP.

MQTT Client Library for .Net and WinRT (https://github.com/ppatierno/m2mqtt) is used to start MQTT publisher and subscriber in the C# Windows form app. For this purpose, the namespaces are declared as follows:

```
using uPLibrary.Networking.M2Mqtt;
using uPLibrary.Networking.M2Mqtt.Messages;
```

The MQTT subscriber is started using the following code:

```
var client = new
MqttClient(System.Net.IPAddress.Parse("10.24.206.126"));
client.MqttMsgPublishReceived +=
client_MqttMsgPublishReceived;
string clientId = Guid.NewGuid().ToString();
client.Connect(clientId);
client.Subscribe(new string[] { "/VPP/Relays" }, new
byte[] { MqttMsgBase.QOS_LEVEL_EXACTLY_ONCE });
```

Here, 10.24.206.126 is the IP address of the computer with MQTT broker, "/VPP/Relays" is the topic of the message. In addition, the following methods are in use:

```
public void client_MqttMsgPublishReceived(object sender,
MqttMsgPublishEventArgs e)
{ SetText(Encoding.UTF8.GetString(e.Message)); }
public void button2_Click(object sender, EventArgs e)
{ textBox1.Text += '1'; }
delegate void SetTextCallback(string text);
private void SetText(string text) {
  if (this.textBox1.InvokeRequired)
  { SetTextCallback d = new SetTextCallback(SetText);
    this.Invoke(d, new object[] { text });
  }
  else
  { this.textBox1.Text += text + "\r\n"; }
}
```

The MQTT publisher is started using following code as follows:

```
byte[] a1 = new byte[1];
var clientS = new
MqttClient(System.Net.IPAddress.Parse("10.24.206.126"));
string clientId = Guid.NewGuid().ToString();
clientS.Connect(clientId);
a1[0] = Convert.ToByte(textBox2.Text);
clientS.Publish("/VPP/Relays", a1);
```

Here, the command (i.e. number) is published from the array *a*1 with topic "/VPP/Relays". Then, this number is read by the MQTT subscriber in Arduino sketch.

The screen shot of the C# Windows form app is shown in Fig. 3. The text field in the left part of the figure includes four numbers 4, 5, 2, and 3, which are commands for

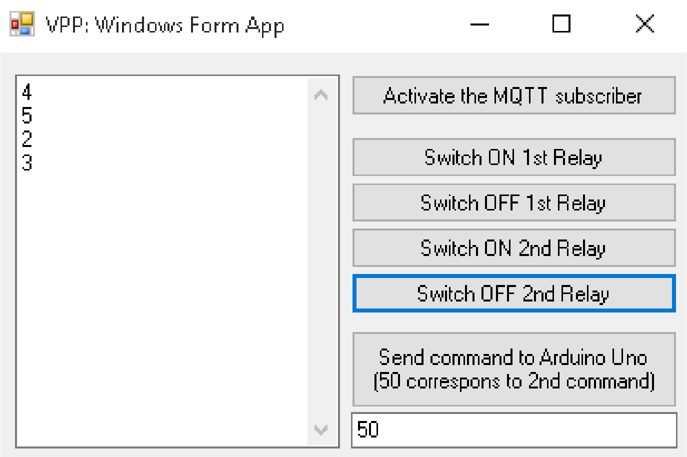

Fig. 3. Screen shot of the C# Windows form app

SPDB: 4 – switch on relay 1 (52 is coding), 5 – switch off relay 1 (53 is coding), 2 – switch on relay 2 (50 is coding), 3 – switch off relay 2 (51 is coding).

MQTT Arduino client (http://pubsubclient.knolleary.net) is used to start MQTT subscriber in the Arduino sketch. The following library is declared as follows:

```
#include <PubSubClient.h>
```

In addition, Ethernet client is started as follows:

```
int Relay1 = 9;
int Relay2 = 8;
int regime;
// MAC address of Arduino Uno board:
byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x03 };
// IP address of Arduino Uno in LAN:
byte ip[] = { 10, 24, 206, 56 };
// IP address of the computer with MQTT broker:
byte server[] = { 10, 24, 206, 126 };
EthernetClient ethClient;
PubSubClient client(server,1883,callback,ethClient);
void callback(char* topic,byte* payload,unsigned int
length) {
  for (int i=0;i<length;i++) { Serial.print("Regime ");
    Serial.print((char)payload[i]); }
  regime = (int)((char)payload[0]);
  if ( regime == 50) { digitalWrite(Relay2,HIGH);
    Serial.print(": Relay 2 is HIGH"); }
  if ( regime == 51) { digitalWrite(Relay2,LOW);
    Serial.print(": Relay 2 is LOW"); }
  if ( regime == 52) { digitalWrite(Relay1,HIGH);
    Serial.print(": Relay 1 is HIGH"); }
  if ( regime == 53) { digitalWrite(Relay1,LOW);
    Serial.print(": Relay 1 is LOW"); }
  Serial.println();
```

```
}
void setup() {
  pinMode(Relay1, OUTPUT); pinMode(Relay2, OUTPUT);
  Serial.begin(9600); Serial.println("Starting...");
  Ethernet.begin(mac,ip);
  if(client.connect("Arduino1 (relays) Client")) {
    Serial.println("Client connected");
    client.subscribe("/VPP/Relays");
  }
  Else { Serial.println("Clientnotconnected"); }
}
void loop() { client.loop(); }
```

Here, the first relay is connected to digital pin 9, the second one – to pin 8.

The hardware was tested successfully in three regimes as follows:

1. The charge of battery using the solar panel (relay 1 is on, relay 2 is off).
2. The power supply of the Arduino Uno board with LED using the power bank (relay 1 and 2 are off).
3. The power supply of the Arduino Uno board with LED using the solar panel (relay 1 is off, relay 2 is on).

The fourth regime (the charge of the battery together with the power supply of the Arduino Uno board with LED) was not realized because of the solar panel's low power (the charge current is half ampere approximately if battery is fully discharged).

4. Conclusions

In this paper, a new IoT concept is proposed for the small VPPs. The main principles are as follows:

1. To use the low-cost reliable Arduino open-source soft- and hardware for the smart control of the power grid.
2. To use the low-voltage devices like the relay module SRD-05VDC-SL-C for Arduino (output voltage is up to 250 V AC) for the control of the high-voltage power lines.
3. Communication between the different parts of the small VPP based on MQTT IoT protocol using specific topics and possibility of the secure TCP connection (passwords and SSL cryptography).

The remote control of the small VPP components is based on the relays and commands sent by the Mosquitto open-source message broker that implements the MQTT IoT protocol. For this purpose, C# Windows form app and Arduino sketch were developed with MQTT publisher and subscriber, respectively.

The hardware part consists of solar panel (12.5 V, 180 mA), two relays SRD-05VDC-SL-C, buck (DC/DC YEC SD200) and buck/boost (DC/DC Tusotek TS-IPS-VO2) converters, two small power banks (first one is to supply the consumption part simulated by Arduino Uno with LED, second one is to supply the SPDB). The hardware was tested successfully in three regimes: the charge of battery using the solar panel (relay 1 is on, relay 2 is off); the power supply of the Arduino Uno board with

LED using the power bank (relay 1 and 2 are off) and the solar panel (relay 1 is off, relay 2 is on).

The most likely prospect for the further development of this work is the usage of more powerful components (e.g. 24 V solar panel(s), power bank with greater capacity), as well as design of smart house with embedded VPP based on proposed concept.

5. References

1. Jennie C. Stephens, Elizabeth J. Wilson, Tarla Rai Peterson: Smart Grid (R) Evolution: Electric Power Struggles. Cambridge University Press, UK. (2015)
2. Łukasz Nikonowicz, Jarosław Milewski: Virtual Power Plants – General Review: Structure, Application and Optimization. Journal of Power Technologies, 92 (3), 135-149. (2012)
3. Das, C.K., Das, N.K., Islam, M.M., Sazzad Hossain, S.M.: Virtual Power Plant as A Remedy to The Power Crisis of Bangladesh: A Case Study-CUET. The annual technical journal of MIST-Galaxy (Dhaka), Bangladesh, Vol. 3 No. 3, Sl. No. 10. (2011)
4. Christos Ioakimidis, Konstantinos Genikomsakis: Design of a Virtual Power Plant in the Presence of Microrenewables and Electric Vehicles in a Microgrid Concept for Real-time Simulation as part of a Remote Lab. In Proceedings of the International Conference on Renewable Energies and Power Quality ICREPQ'13. Bilbao, Spain, 1-5. (2013)
5. Wind Turbine Control and Monitoring. Eds. Ningsu Luo, Yolanda Vidal, Leonardo Acho. Springer, Cham Heidelberg New York Dordrecht London (2014)
6. Gerro Prinsloo, Robert Thomas Dobson: Solar Tracking: High Precision Solar Position Algorithms, Programs, Software and Source-Code for Computing the Solar Vector, Solar Coordinates & Sun Angles in Microprocessor, PLC, Arduino, PIC and PC-based Sun Tracking Devices or Dynamic Sun Following Hardware. Stellenbosch University, Stellenbosch, South Africa. (2015). [Online]. https://www.researchgate.net/publication/263085113_Solar_Tracking_High_precision_solar_position_algorithms_programs_software_and_source-code_for_computing_the_solar_vector_solar_coordinates_sun_angles_in_Microprocessor_PLC_Arduino_PIC_and_PC-based_sun (current May 2016)
7. Dirk Slama, Frank Puhlmann, Jim Morrish, Rishi M. Bhatnagar: Enterprise IoT: Strategies and Best Practices for Connected Products and Services. O'Reilly Media, Sebastopol, USA. (2015)
8. Norris, D.: The Internet of Things: Do-It-Yourself at Home Projects for Arduino, Raspberry Pi, and BeagleBone Black. McGraw-Hill Education, New York, USA. (2015)
9. Charalampos Doukas: Building Internet of Things with the Arduino. Amazon.com, CreateSpace Independent Publishing Platform, Seattle, Washington, USA. (2012)
10. Dimeas, A.L., Hatziargyriou, N.D.: Agent Based Control of Virtual Power Plants. In Proceedings of the 14th International Conference on Intelligent System Applications to Power Systems. Kaohsiung, Taiwan, 536-541. (2007)
11. Laurențiu Alboteanu, Gheorghe Manolea, Sergiu Ivanov: Management of Renewable Energy Sources Integrated in a Micro Smart Grid. Annals of the University of Craiova, Electrical Engineering series, No. 39, 131-137. (2015)