SIELA 2020

Online Conference

3 - 6 June 2020, Bulgaria

Laboratory Automation System Using IOT Devices

Dimitar MINCHEV, Atanas DIMITROV***

** Burgas Free University, Faculty of Computer Science and Engineering, Burgas, Bulgaria; ** IBA Bulgaria, 9 "Bulair" Str., Burgas, Bulgaria*

I. Introduction

This paper describes a Laboratory Automation System using IoT devices. The main hardware modules consist ESP8266 micro-controller by Espressif and energy meter HLW8012 by HLW Technology. Communication between the IoT devices uses IEEE 802.11g WiFi network, TCP/IP stack and MQTT protocol. The minimum configuration of the project includes one control device called Gateway and two or more manageable Host devices. The devices can be controlled remotely by user friendly, simplified web-based interface. The project is open software and hardware.

Internet of Things (IoT) is becoming more and more a part of our lives on a daily basis. IoT devices are primarily used to control, monitor, and manage the technology that we use every day. This means that the devices are typically designed to be easily installed and managed by the costumer. Industry analysts estimate the number of connected devices to be approx. 50 billion by the end of the 2020 year. This paper describes the implementation of Laboratory Automation System (LAS) using IoT Devices with energy measurement capabilities.

III. Software

The source code of the project is written in C/C++ using Arduino IDE [4]. During the development of the firmware, additional external libraries were used as follows:

- ESPAsyncTCP [5] - Asynchronous TCP library, aimed at enabling trouble-free, multi-connection network environment for Espressif's ESP8266 MCUs.
- ESPAsyncWebServer [6] - Arduino compatible Asynchronous HTTP and WebSocket Server for ESP8266.
- PubSubClient [7] - This library provides a client for doing simple publish/subscribe messaging with a server that supports MQTT.
- uMQTTBroker [8] - Arduino compatible MQTT Broker library for Espressif's ESP8266 MCUs.
- ArduinoJson [9] - JSON message protocol library implementation in C++ for Arduino and IoT.
- HLW8012 [10] – Arduino energy measurement library by Xose Perez.

The full source code of the project is available as open software in GitHub Repository and can be freely downloaded on the Internet at address: https://github.com/dimitarminchev/HAS8266

Testing the project software in laboratory-controlled environment on mobile device and laptop is shown on Fig.2.

Fig. 2. Testing the project software in laboratory-controlled environment

II. Architecture

The proposed system architecture model is modular and consists of multiple end devices, called Hosts. The Hosts are capable of incorporating a variety of laboratory electrical appliances. Each of these end-devices has a wireless network interface which is connected to the single control device and called Gateway. The Gateway controls the operation of all Hosts. It establishes a communication channel between costumer and Hosts, thereby providing their remote control and management.

The Message Queue Telemetry Transport (MQTT) [1] communication protocol is used to transmit data between Hosts and Gateway devices. The choice of this communication protocol is due to its advantages. It is lightweight, open to the public, as described in the OASIS [2] specification, standardized by ISO/IEC PRF 20922 [3] and it working over the Transmission Control Protocol/ Internet Protocol (TCP/IP).

When is powered, a single Host introduces itself to the Gateway with continuous "Hello" message. This identification message contains the name of the Host. After receiving the message, the Gateway is responding back to the "Hello" message with a new "Stop" message. In this way, the Gateway confirms the Host that it is successfully connected to the network, manageable by Gateway and adds its IP address to his routing table.

When the Gateway receive user command to turn-on a specific device, it send "Control" message to the device IP address from the routing table to power on the desired Host. Receiving "Control" message, the Host periodically respond with "Power" message until Gateway send "Stop" message. The "Power" message consists of information for power consumption, measured by HLW8012, and contains three fields: voltage of the power grid, consumed current and active power from managed device.

When the Gateway receive user command to turn-off a specific device, it send "Control" message to the device IP address from the routing table to power off the desired Host.

The UML sequence diagram of the MQTT communication protocol between Gateway and Host device is shown on Fig. 1.

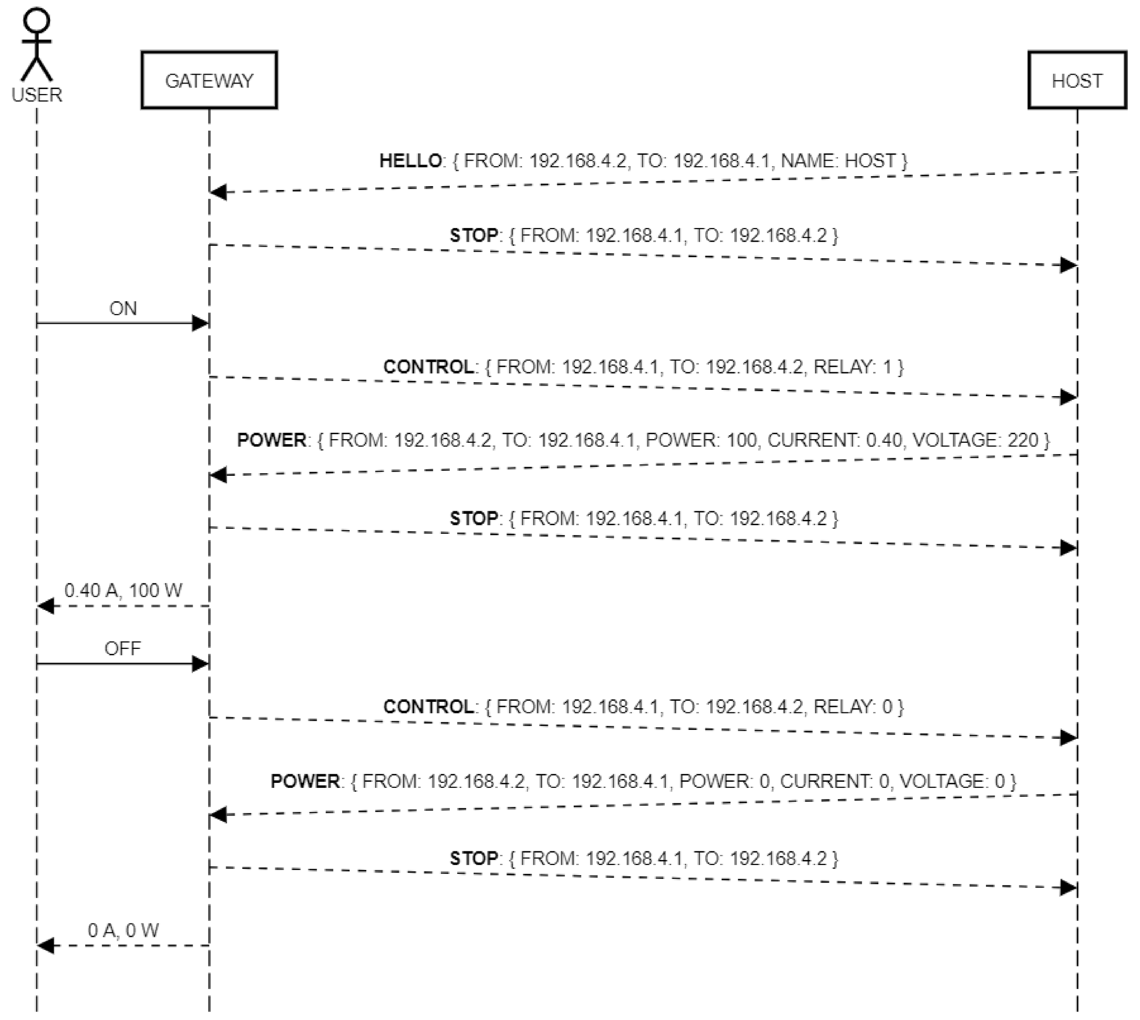

Fig. 1. UML Sequence Diagram of the MQTT Communication protocol

IV. Hardware

The ESP8266 chip integrates an ultra-low power 32-bit MCU architecture. The core of the processor can work with clock speed of 80MHz or 160MHz. The module supports RTOS, IEEE802.11 b/g/n standard and complete TCP/IP stack, which is make it ideal for adding to an existing network device or building a separate network controller. HLW8012 can measure RMS values of current, voltage, and active power with an internal clock and a PWM interface. To operate, HLW8012 chip needs 5V DC voltage. It generates 50% duty square waves with frequency that depends on the magnitude of the measured parameter (power, current, or voltage).

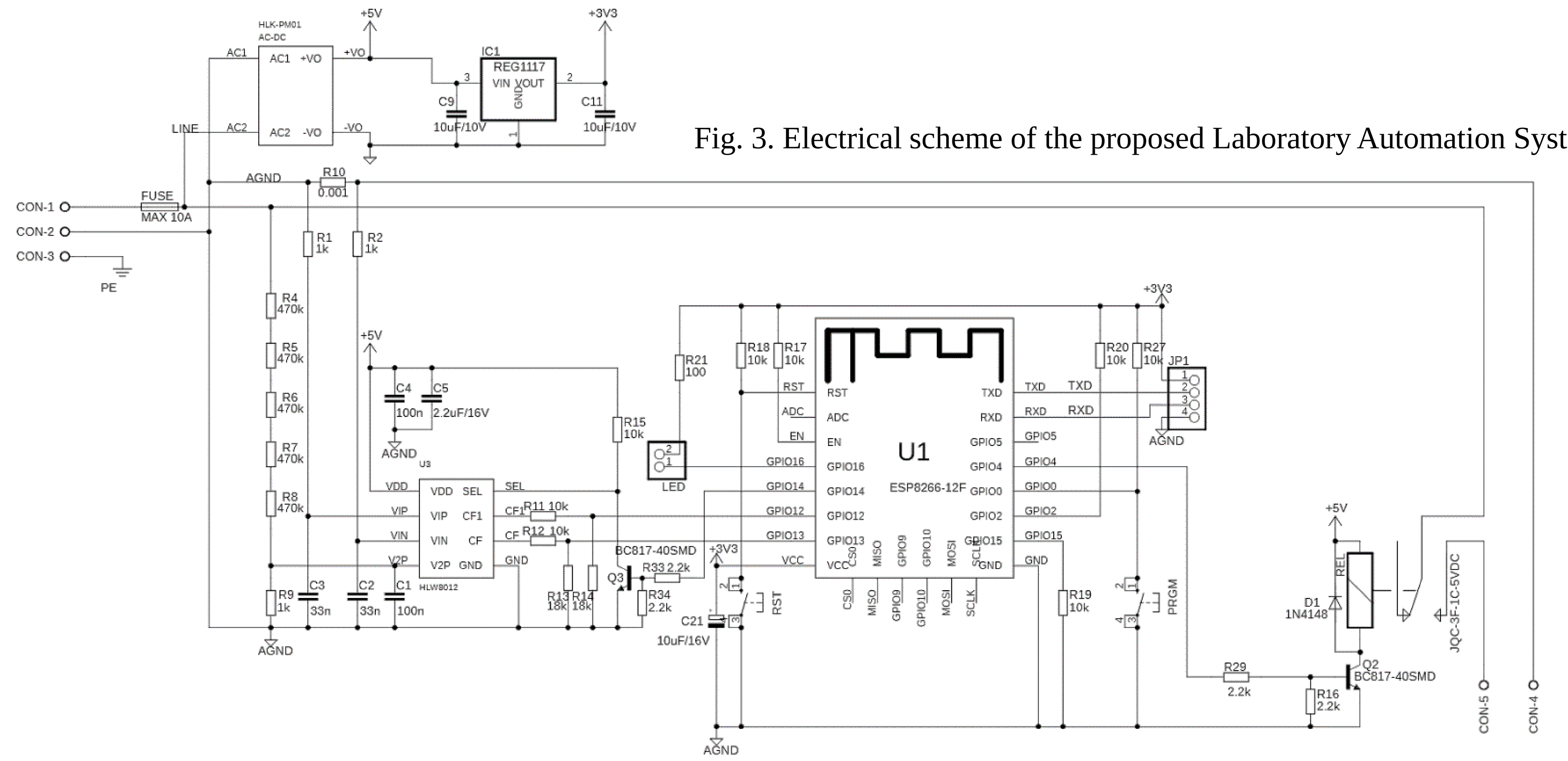

Fig. 3. Electrical scheme of the proposed Laboratory Automation System

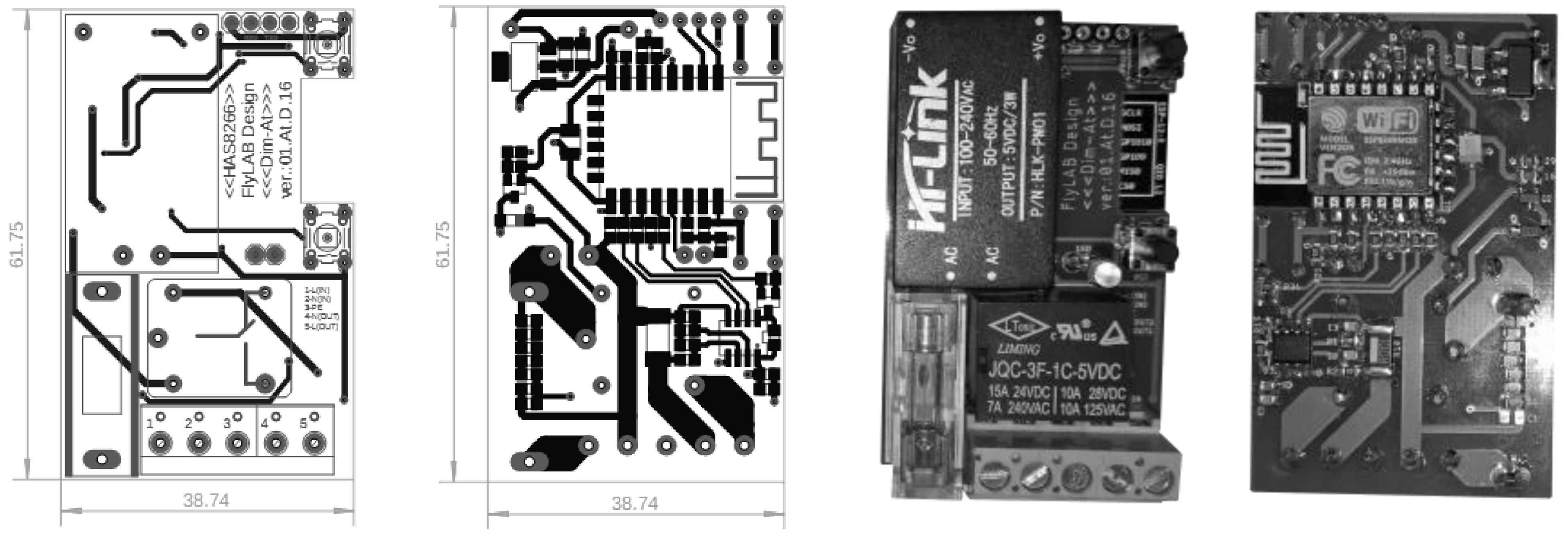

Fig. 4 and 5. Graphical originals and PCB of the proposed Laboratory Automation System

V. Conclusion

This paper describes the implementation of a Laboratory Automation System (LAS) using Internet of Thing (IoT) devices. The main hardware modules consist ESP8266 micro-controller by Espressif and energy meter HLW8012 by HLW Technology. Communication between the IoT devices uses IEEE 802.11g WiFi network, TCP/IP stack and MQTT protocol. The minimum configuration of the project includes one control device called Gateway and two or more manageable Host devices. The devices can be controlled remotely by user friendly, simplified web-based interface. The project is open software and hardware.

REFERENCES

[1] Message Queue Telemetry Transport (MQTT), http://www.mqtt.org
[2] OASIS Standard Incorporating Approved Errata 01, MQTT Version 3.1.1 Plus Errata 01, http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html
[3] ISO/IEC 20922:2016, Information technology, Message Queuing Telemetry Transport (MQTT) v3.1.1, https://www.iso.org/standard/69466.html
[4] Arduino IDE, https://www.arduino.cc/en/Main/Software
[5] ESPAsyncTCP, https://github.com/me-no-dev/ESPAsyncTCP
[6] ESPAsyncWebServer, https://github.com/me-no-dev/ESPAsyncWebServer
[7] PubSubClient, https://github.com/knolleary/pubsubclient
[8] uMQTTBroker, https://github.com/martin-ger/uMQTTBroker
[9] ArduinoJson, https://github.com/bblanchon/ArduinoJson
[10] X. Perez, HLW8012, https://bitbucket.org/xoseperez/hlw8012
[11] ESP8266EX, Espressif System, https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
[12] HLW8012, Hiliwi Technologies, www.hiliwi.com/products_detail/&productId=36.html