

Remote Control of an Automatic System for Handling Fragile Objects

Ph.D. Eng. Ionel Laurențiu ALBOTEANU^{1,*}, Eng. Ionel Bogdan ILIE¹

¹ University of Craiova, Faculty of Electrical Engineering

* ialboteanu@em.ucv.ro

Abstract: The paper presents a remote control solution based on wireless technology of a small-scale fragile object handling system. The description of the handling system and the local control solution by means of a PLC have been published in Hidraulica Magazine no. 4/2020 [1].

Keywords: Remote control, wireless, automatic system, handling, electro-pneumatic drive, vacuum technology

1. Introduction

The term wireless communication was introduced in the 19th century, and wireless communication technology developed over the following years. It is one of the most important means of transmitting information from one device to another. In this technology, information can be transmitted through the air without requiring cable or wires or other electronic conductors by using electromagnetic waves such as IR, RF, satellite, etc. [2].

Currently, wireless communication technology refers to a variety of wireless devices and communication technologies, from smartphones to computers, tabs, laptops, Bluetooth technology, printers.

Wi-Fi is a low-power wireless communication that is used by various electronic devices such as smartphones, laptops, etc. In this configuration, a router acts as a wireless communications hub. These networks only allow users to connect in close proximity to a router. WiFi is very common in networking applications that provide wireless portability [2].

The aim of the work is remote monitoring and control of an automatic fragile object handling system through the Blynk application. This is achieved by attaching a wireless module to the automatic handling system designed to remotely monitor and control the handling of various fragile objects using a vacuum suction cup [3], [4].

The wireless module used is NodeMCU ESP8266, as it provides a complete and self-contained Wi-Fi network solution, allowing it to host the application or offload all Wi-Fi network functions from another application processor.

2. Structure of the handling system

The proposed automatic fragile object handling system consists of a stock of fragile plates at the entrance, an output stock and a manipulator system that transfers the plates to be processed from the input stock to the output stock (Fig.1).

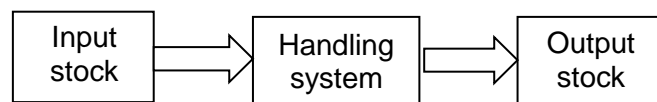


Fig. 1. Structure of handling system

Starting from the previously presented structure of the fragile object handling system, it is proposed to create a manipulating system that performs the handling operation for a single piece per work cycle, and the input stock has a capacity that can include several fragile plates.

The entire structure is integrated in an electro-pneumatic drive system with linear (pneumatic cylinder) and rotary (electric motor) actuators, controlled by monostable and bistable valves [5],[6], [7], [8].

The block diagram of the handling system is shown in Fig. 2.

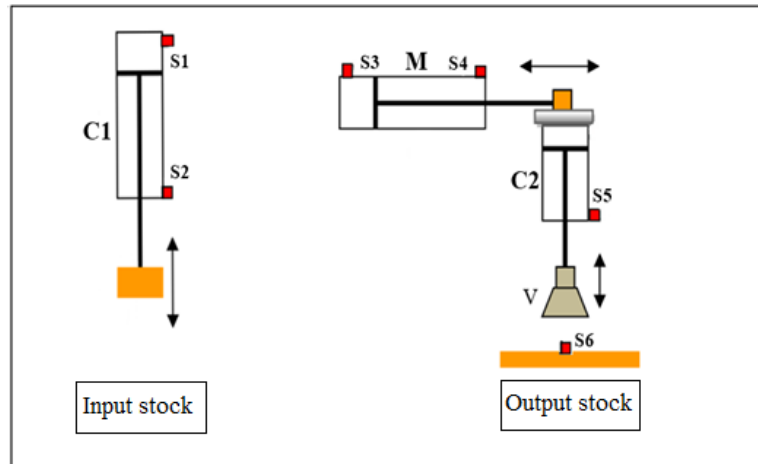


Fig. 2. Block diagram of handling system

Based on the structure of the pneumatic drive, the general scheme of the automatic handling system (Fig. 3) has been drawn up.

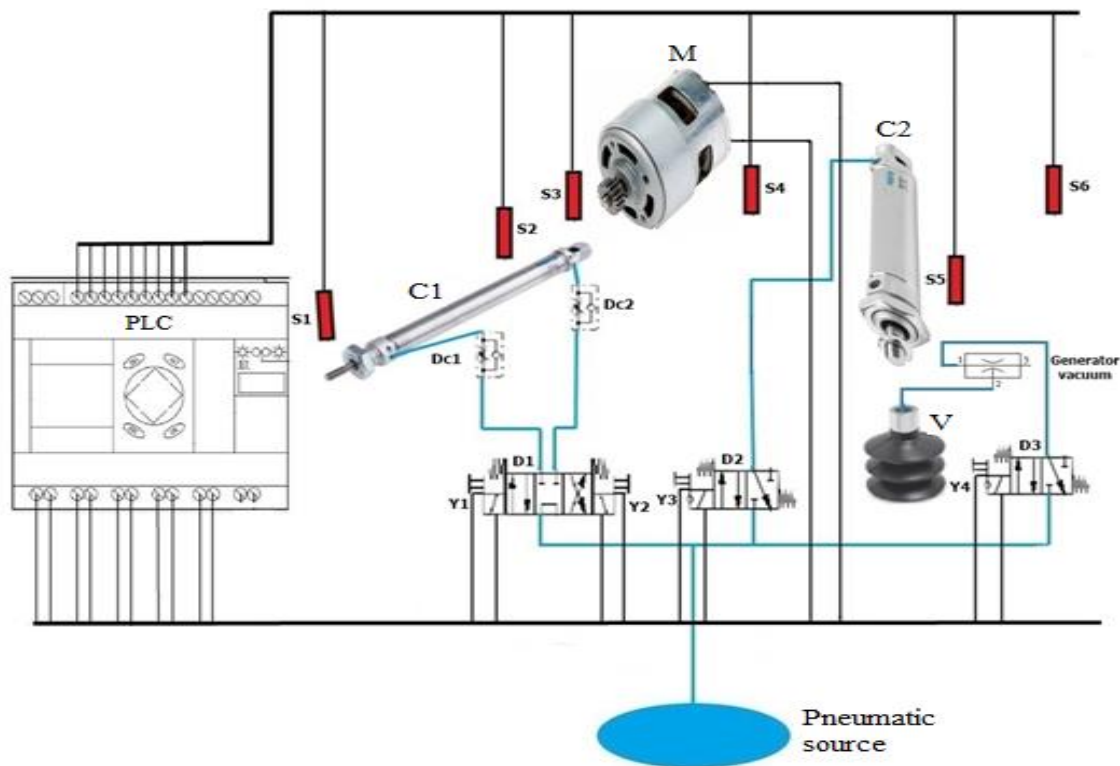


Fig. 3. Schematic diagram for the interconnection of the elements of the handling system

The meaning of the elements used in the scheme is as follows:

- PLC – programmable logic controller;
- D1...D3 – pneumatic valve;
- C1, C2 – pneumatic cylinders;
- M – electric motor;
- DC1...DC3 – one-way flow control valves;
- S1...S6 – sensors;
- Y1...Y4 – relays valve control;
- V – vacuum suction cup.

3. Structure of remote control system

3.1. Hardware structure

One of the most challenging problems in the design of control systems is the development of the system architecture, namely the choice of hardware components (e.g.: sensors, actuators, integrated circuits, microcontrollers) [9], [10].

To design the system, several tests have been carried out with various types of development boards, among which we can list: ESP8266 NodeMCU, Arduino UNO WIFI Rev2 and ESP-01 ESP8266.

Following the tests carried out, it has been proven that the best solution that lends itself to implementation is the use of the ESP8266 NodeMCU development system.

The main advantage of this system, compared to the two tested modules, is given by the multitude of analog and digital outputs that are arranged on the same board, by the size but also by the technical characteristics that will be presented next.

1) Development board ESP8266 NodeMCU

The NodeMCU ESP8266 development board comes with the ESP-12E module which contains the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at an adjustable clock frequency from 80 MHz to 160 MHz. The NodeMCU has 128 KB of RAM and 4 MB of Flash memory to store data and programs. Its high processing power with built-in Wi-Fi/Bluetooth and sleep mode operation functions make it ideal for IoT (Internet of Things) projects [11].

The NodeMCU can be powered using a Micro USB plug but also a VIN pin (external power pin).

ESP8266 NodeMCU development board is based on a wireless ESP8266 microcontroller with Wi-Fi 802.11 compatible IDE-Arduino development environment. The structure of this board is based on a standard Arduino hardware design with similar proportions to the Arduino Uno and Leonardo. The low price of the device has made it very popular, being included in modules, but also in development boards for IoT applications. The ESP8266 contains a 32-bit processor clocked at 80 MHz, with 32kB instruction RAM and 80kB user RAM.

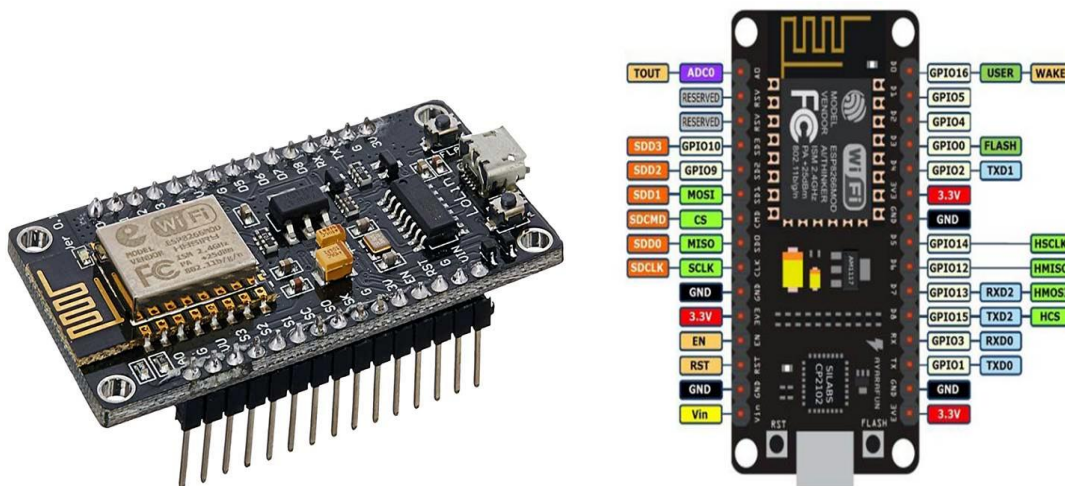


Fig. 4. ESP8266 NodeMCU development board

2) Expansion module

For the control of the component elements that make up the hardware part of the handling system, the solution of building an expansion board consisting of the supply part and the control part has been adopted (Fig. 5). This board has been designed using the Targhet 3001 program [12].

The module is useful for controlling several devices operating at different voltages (12V, 24V) using the NodeMCU ESP8266 development board.

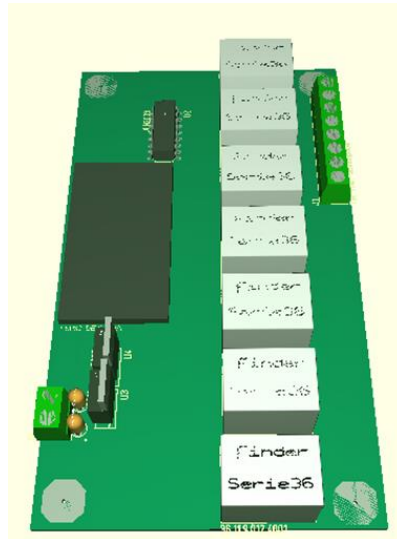


Fig. 5. Expansion module

3.2. Software structure

For the software part, the Blynk application has been used, an application that can be installed on almost any smartphone or tablet [13].

This is to create a user interface and provide a way to control the manipulation system.

The Blynk app has been designed for the Internet of Things. It can remotely control hardware, display sensor data, store data and visualize it.

There are three major components in the platform (Fig. 6):

- *Blynk application* - allows the creation of project interfaces using various widgets that it provides.
- *Blynk Server* - responsible for all communications between smartphone and hardware. One can use any Blynk cloud or run a local Blynk private server. It is open-source and can easily manage thousands of devices.
- *Blynk libraries* - for all popular hardware platforms - allow communication with the server and process all incoming and outgoing commands.

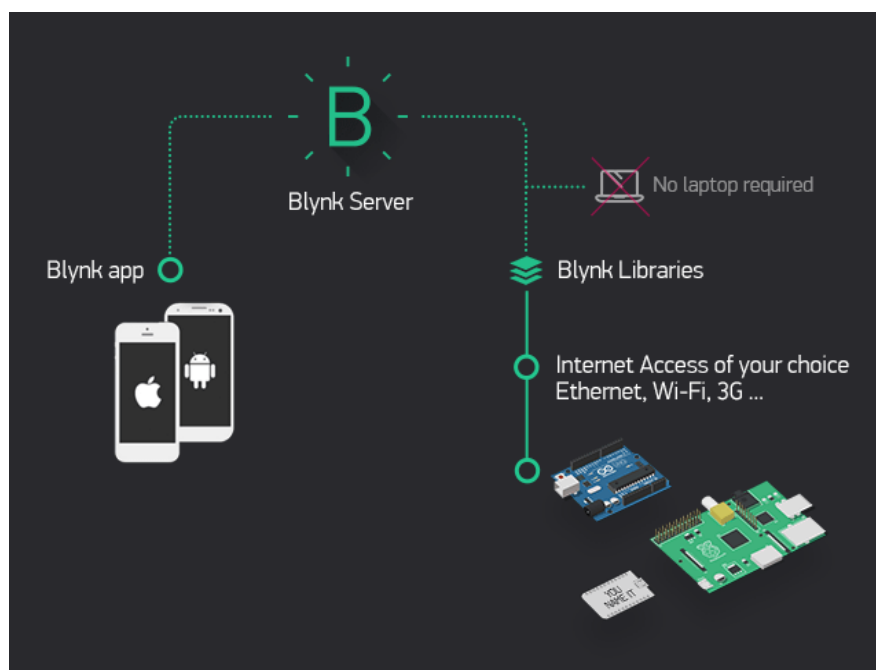


Fig. 6. Structure of the Blynk application

Every time a button is pressed in the Blynk app, the message reaches the Blynk Space Cloud, where it finds its way to the hardware, working the same way in the opposite direction, all happening in an extremely short time.

To create the virtual control panel in the Blynk application, buttons are created using widgets.

Widgets are pre-designed pieces of the graphical user interface. Each widget performs a specific input/output function when communicating with hardware or the end user.

Two types of widgets have been used for the chosen application:

Button widget

It works in push or switch modes. Allows sending any numeric value to clicks and button release events. By default, the button uses values 0/1 (LOW/HIGH). The button sends 1 (HIGH) when pressed and 0 (LOW) when released.

It is also possible to change the state of the button from the hardware side. For example, turn on the button assigned to virtual pin V1:

```
Blynk . virtualWrite ( V1 , HIGH );
```

Can change the hardware button labels with:

```
Blynk.setProperty(V1, "onLabel", "ON");
```

```
Blynk.setProperty(V1, „offLabel”, „OFF”)
```

or change the colour:

```
///#D3435C - Blynk RED
```

```
Blynk . setProperty ( V1 , „culoare” , „#D3435C” );
```

Button status can be obtained from the server if the hardware has been disconnected with Blynk Sync:

```
BLYNK_CONNECTED() {
    Blynk.syncVirtual(V1);
}
BLYNK_WRITE(V1) {
    int buttonState = param.asInt();
}
```

Led widget

A simple indicator (similar to an LED).

- if the value is equal to the DataStream min setting, the indicator is OFF
- if the value is equal to the DataStream max setting, the indicator is ON
- intermediate values are used for brightness control (such as PWM)

```
// DataStream is configured to 0..255 range
```

```
// LED light setting is to 50%
```

```
Blynk.virtualWrite(V1, 127)
```

It can change the color of the LED with:

```
///#D3435C - Blynk RED
```

```
Blynk.setProperty(V1, „culoare”, „#D3435C”);
```

The outputs have been configured as custom widgets (windows) for each movement in the form of virtual buttons for manipulator movements, and for the system status a led has been added for the status of the presence of the piece on the output stock.

The virtual control panel in the Blynk application is illustrated in Fig. 7.

After completing the previous steps, proceed to programming the NodeMCU Esp8266 development board.

On the development board there is a program written in the native Arduino language. It has the role of processing the information received wirelessly and commanding the movement of the manipulator.

This programming is done using the Arduino IDE program [14].

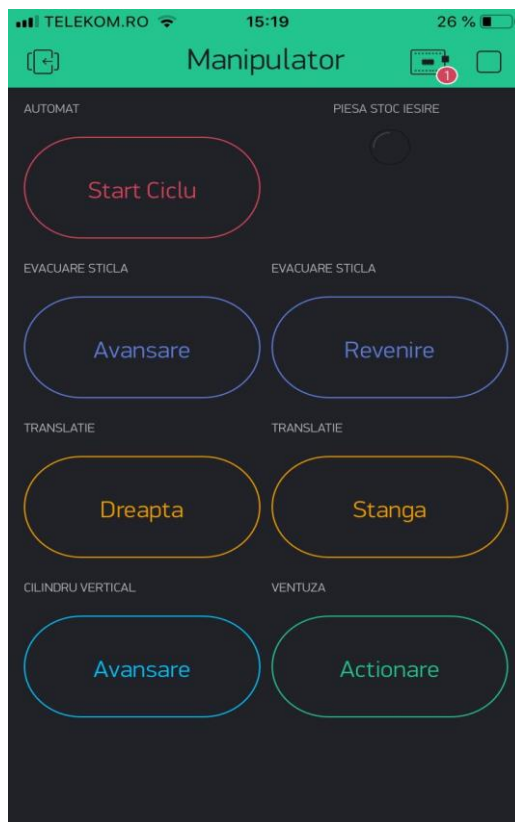


Fig. 7. Virtual control panel in the Blynk app

4. Achievement and testing of the automatic handling system

The achieved automatic handling system (Fig. 8) can work in the following ways:

- local manual mode;
- local automatic mode;
- remote manual mode;
- automatic remote mode.

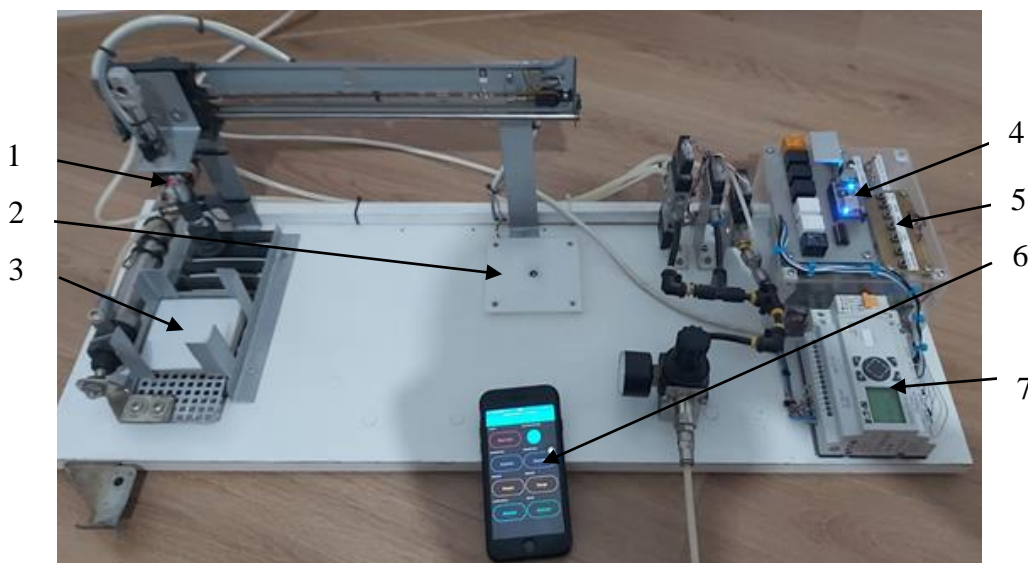


Fig. 8. Automatic handling system achieved: 1- manipulator; 2- output stock; 3- input stock; 4- NodeMCU Esp8266 development board; 5- push button board; 6- smartphone; 7- PLC

Experimenting with the operation mode of the remote monitoring and control system has been physically carried out on the experimental model made, through a series of tests. They captured all the stages of the process of handling fragile objects through the Blynk application, which aims to monitor and control the system remotely.

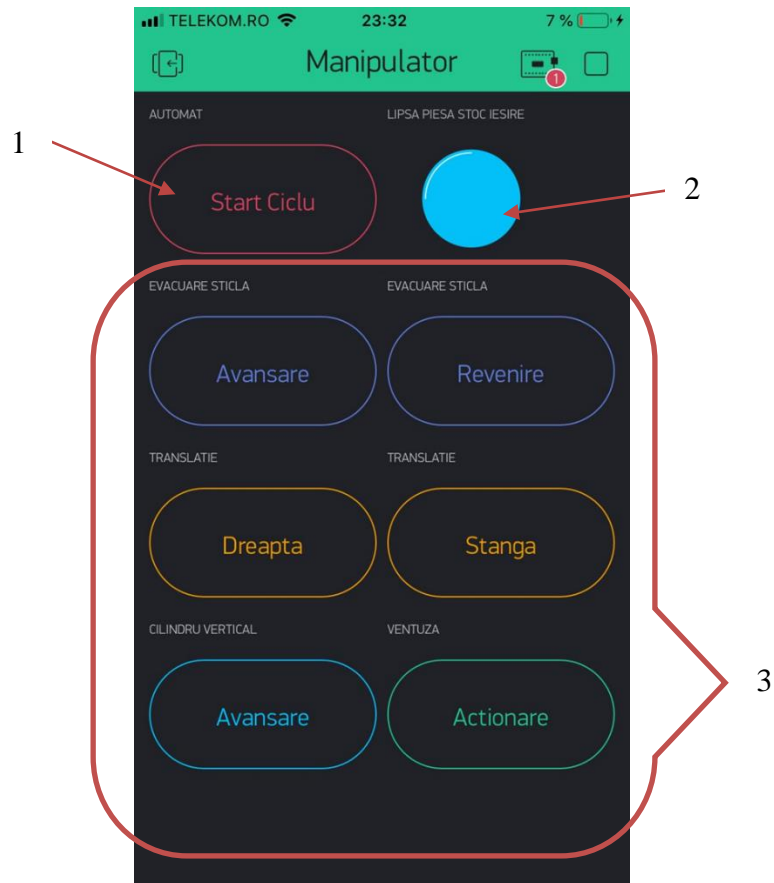


Fig. 9. Virtual control panel of remote control system: 1- start button of automat mode; 2- LED status of output piece stock; 3- buttons of manual remote control

Figure 9 shows the screen of the application during operation. The blue colour of LED 2 shows the lack of piece in the output stock, which means that button 1 can be activated to start the automatic cycle.

5. Conclusions

The main goal of the research presented in the paper has been to create a wireless module that has been attached to an automated fragile object handling system to be able to remotely monitor and control the handling of various fragile objects using a vacuum suction cup.

Taking into account the composition of the wireless module, its programming has been carried out using the Blynk application.

The design consisted of the use of widgets from the Blynk application through which all stages related to the handling of fragile objects can be monitored and controlled.

For the remote monitoring and control of the fragile object handling system, the following steps have been taken: designing the control system for a pneumatic manipulator and choosing the component elements, making the handling system, and using widgets in the Blynk app to program the wireless module.

Experimentation with remote monitoring and control of the automated fragile object handling system has led to the conclusion that the system is operating correctly according to the established operating protocol.

References

- [1] Alboteanu, L. “Automatic System for Handling Fragile Objects.” *Hidraulica Magazine*, no. 4 (December 2020): 26-32.
- [2] FMUSER International Group INC. “Diferite tipuri de comunicații fără fir cu aplicații”/ “Various types of wireless communication with applications.” May 29, 2020. Accessed February 20, 2021. <https://ro.fmuser.net/>.
- [3] ***. “About flexible production systems and complex automation.” *Plant Engineering*, XIII Year, no. 11 (118) (2013): 42-45.
- [4] Hucknall, D. J. *Vacuum Technology and Applications*. Oxford, Butterworth-Heinemann, 2013.
- [5] Alboteanu, L., Gh. Manolea, and F. Ravigan. “Automatic sorting and handling station actuated by pneumatic drive.” *Annals of the University of Craiova, Electrical Engineering Series*, no. 1 (2018): 1-8.
- [6] Alboteanu, L. “Automatic Processing Station Actuated by Pneumatic Drive.” *Hidraulica Magazine*, no. 1 (March 2019): 16-22.
- [7] Alboteanu, L. “Modeling an Automatic Processing Station Using Fluidsim Software.” *Hidraulica Magazine*, no. 3 (September 2017): 27-30.
- [8] Alboteanu, I. L. “Pneumatic Tracking System for Photovoltaic Panel.” *Hidraulica Magazine*, no. 1 (March 2015): 32-39.
- [9] Gorton, Ian. *Essential Software Architecture*. Springer Publishing House, 2006.
- [10] Carnegie Mellon University, Software Engineering Institute. “What Is Your Definition of Software Architecture?” January 22, 2017. <http://www.sei.cmu.edu/architecture/definitions.html>.
- [11] ElectronicWings. “Introduction to NodeMCU.” <https://www.electronicwings.com/nodemcu/introduction-to-nodemcu>.
- [12] ***. Target 3001 Software. <http://server.ibfriedrich.com/>.
- [13] ***. Blynk software. <https://blynk.io/>.
- [14] ***. Arduino software. <https://www.arduino.cc/en/software>.