Remote Laboratory for Servomotor Studying

[1]Vîrlan Petru, [1]Todos Petru, [2]Adăscăliței Adrian

[1]Tehnical University of Moldova
E-mails: petru.virlan@feie.utm.md, petru.todos@adm.utm.md

[2]Tehnical University „Gh. Asachi" Iași, Romania
E-mail: adrian.adascalitei@utiasi.ro

ABSTRACT

This paper presents a way of creating a laboratory work on controlling a remote servomotor. Using LabView, a real experiment is performed, the equipment being remotely controlled, and viewing with a camcorder. The purpose of this paper is to demonstrate the possibility of carrying out laboratory works for students requesting e-learning.

Keywords: virtual, instrument, engineer, servomotor, LabView, remote, laboratory, study

1. INTRODUCTION

An engineer is a person with a technical - theoretical and practical training, obtained in a higher education institution practicing engineering.

Unlike scientists who study the nature and phenomena of nature to establish principles, axioms and theorems, engineers apply the theoretical principles in mathematics and physics to create a concrete product, such as an example, a converter or a mechanism [1].

The laboratory is the subject of the largest financial expenditures in the technical educational institutions, while the rapid development of the technologies and techniques of the contemporary period make it even more difficult to modernize the laboratories. However, because we have the information technologies that provide us with virtual tools to create virtual laboratories that are no more down-to-earth than real laboratories. At the same time, investments are considerably low.

Remote lab - this kind of lab, as shown in Figure 4, is a software application that can be accessed online via the browser, but it allows for a real experiment, the equipment being routed remotely, and visualization done with a video camera [2].

2. REMOTE LAB EQUIPMENT FOR SERVICE CONTROL

To create a "Remote" lab, we will use the LabView application, an Arduino board, communication equipment, a servomotor and a potentiometer. We will also create a control and data acquisition

interface to provide distance learning students with the lab by visualizing real physical equipment through a video camera.

Below we present each hardware component of the lab with a small description of it.

The LabView application

LabVIEW is a graphical programming environment based on the G language (graphical language), designed specifically to build applications to control and acquire data, analyze them and present the results. We mention some of the most important features of this environment: - Automatically resolves most of the problems associated with hardware resource management and operating system communication, and in this way the user can focus on the concrete problem he has to solve and not on the operation of the computer . For example, almost all of the nodes (functions), as well as the graphs, adapt to any type of data, whether they are simple real values or data structures;

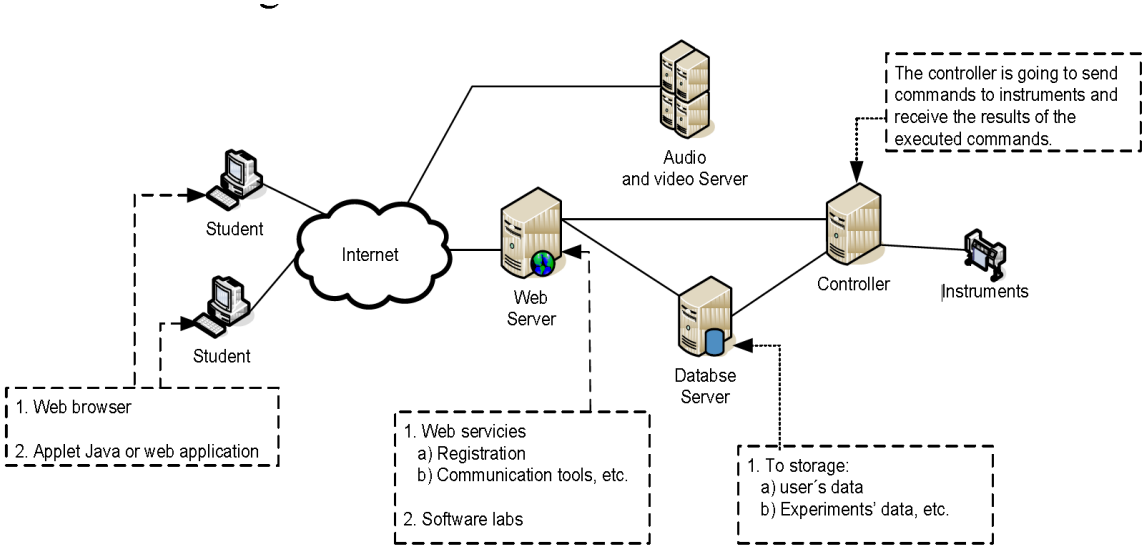

Figure. 1. Remote lab [2]

- graphical language is more compact, the chart contained in a window contains more information than text, and is easier to read and understand, and drawing a chart is faster than writing an equivalent text;

- in graphical language, parallelism is natural, so writing programs that perform parallel data processing is just as simple as sequential processing;

- allows networking across multiple computers via TCP / IP and UDP, with many functions for working in local area networks or the Internet;

- contains many pre-fabricated applications in various fields, along with the corresponding G code, which can be used directly, can be taken as a teaching example or can be modified by the user to best meet the specific work needs.

All LabVIEW solutions work by default on multithreading without requiring additional programming. In this way, the time needed for the development of applications can be significantly reduced. LabVIEW is available for a wide range of operating systems: Windows 2000 / NT / XP / Me / 9x, Mac OS, Sun Solaris, Linux [3].

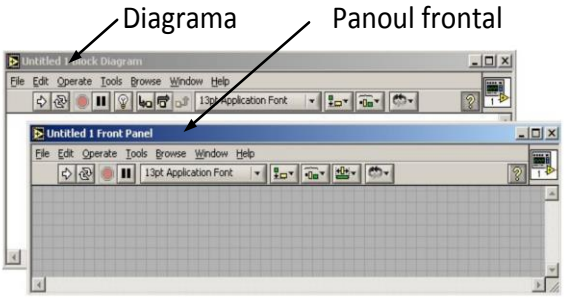

Figure 2. LabVIEW panel and diagram [3]

The Front Panel is the graphical user interface, the window the user will see when they access the application. Using panel elements, the application receives the input data and then displays the output data that resulted from the run.

Block Diagram is the window where the programmer describes the algorithm after which the application will perform the calculations and reasoning required to process the information. In most cases, after the developer has made an application and delivered it to a user, the latter no longer has access to the chart.

Plate arduino Uno

Arduino is a very simple microcontroller platform to use. Arduino can be used to develop interactive applications. In fact, the information is taken from a wide range of input elements (sensors and switches), processed inside the microcontroller and transmitted to an equally wide range of output elements: LEDs, motors, actuators, etc.

The advantages of Arduino over these microcontroller-based systems are:

- reduced acquisition costs;
- can be used on any operating system (Linux, Windows or MacOS). Most development boards are limited to the Windows operating system.
- a simple and easy to learn programming environment.

Arduino Uno is a development board based on the ATmega328P microcontroller, with 6 analog inputs, 14 digital inputs / output pins (6 of which can be used as PWM outputs), a 20 MHz quartz oscillator, a USB connection, a power jack, and a reset button.

Significance of pins:

- VCC - the positive pole of the source (+);
- GND - mass (-);
- PB 0-7 - the 8 input / output pins of port B;
- PC 0-5 - the 6 input / output ports of port C;
- PD 0-7 - the 8 input / output pins of port D;

- ADC 0-5 - input pins that provide digital analog conversion.

Analog inputs are used to read analog signals from temperature sensors, light sensors, pressure sensors, humidity, etc. An analog input pin can measure a current or voltage signal between 0-5 V. Digital Inputs / Outputs: Allows you to read the state of an input / output element or control elements that have two states: closed ie 0 (LOW values) or open 1 (HIGH); The Pulse Width Modulation (PWM) Pulse Width Modulation (PWM) Pulse Width Modulation (PWM) Pulse Width Modulation (PWM) Pulse Width Modulation Pulse Width Modulation Pulse Width Modulation Pulse Width Modulation The USB port has two roles: to supply the Arduino platform and to provide system data. Power supply of the Arduino platform can be made from an AC-DC power supply with recommended voltage between 7-12 V through the power socket.

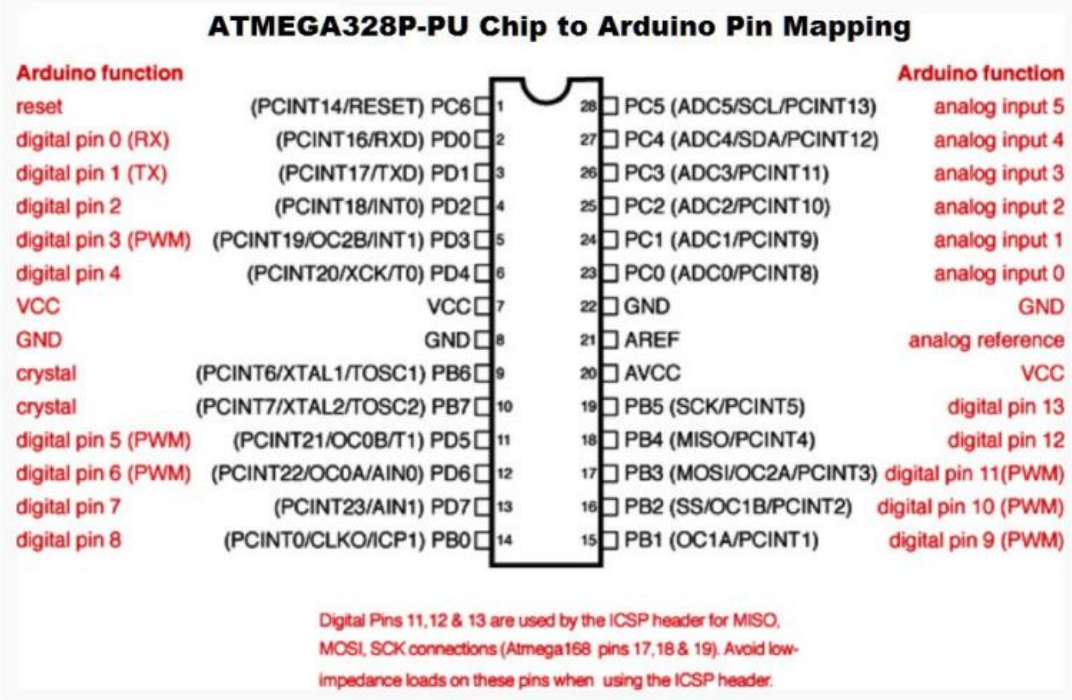

Figure 3. Atmega328P microcontroller pins [4]

Requirements for C-program training would seem difficult for beginners, but due to its structure it has a wide range of possibilities.

Several training rules will be enough to be memorized:

- // (unilateral comment) is often used in the content of the program to more explicitly describe one of the program rows. All that is written after the double bar and to the end of the given row will be ignored by the compiler;

- / * * / (multilateral comment) this structure is used if there is a need to explicitly describe the content of several rows in the program, all that is written between these symbols will be ignored by the compiler;
- {} is used to determine the start and end of the control block (used for functions and cycles);
- **;** each command must end with this symbol (the lack of this symbol makes the compilation impossible).

variables:

- int (integer numbers) stores numbers in memory using 2 bits or 16 bits and can include any integer in the range -32768 ..... 32767;
- **long** is used when int is not enough, occupies memory from 4 bits up to 32 bits and contains integer numbers in the range -147483684 .... 2147483647;
- **boolean** (true / false), occupies only one bit of memory;
- **float** (decimal numbers) is used to enter the decimal numbers;
- **char** (symbol) holds a symbol using the ASCII encoding system.

Mathematical Operations:

- **=** (assigns) makes an assignment (example: x = 10 * 2, assigns it to variable x number 20);
- **%** (as from the division) 12% 10 result the result 2;
- **+** (assembly);
- **-** (difference);
- ***** (Îmulțire);
- **/** (division).

Symbols for logical comparison:

- **==** (equal);
- **!=** (different);
- **<** (smaller);
- **>** (bigger);

By using the programming rules, the program is being developed, which will command the servomotors according to the prescribed algorithm.

Running the "arduino" program from the development environment you just downloaded to the previous step (see the screenshot below). Arduino connects to the PC via a serial port. The first step you have to make is to determine this port. The easiest way is to connect the board, wait about 30 seconds - 1 minute to make sure it was detected by your PC and then open the "Tools -> Serial Port" menu. You should see one or more entries. Memorize them (or write them on a sheet of paper / make a screenshot). Disconnect the Arduino card from the USB port (removes the cable from the PC). Opens the "Tools -> Serial Port" menu again. That port that has disappeared is the port associated with the Arduino board. Reconnect the Arduino board to the PC, wait for it to be recognized by the PC, and then select the port from the "Tools -> Serial Port" menu. The next step is to select the type of plaque you work with. From the "Tools -> Board" menu, select the type of board you are working with (Arduino Uno, Leonardo, Mega, etc.) [4].

3. LIST OF THE ARDUINO PROGRAM FOR DIRECT START AND REVERS OF SERVOMOTOR

The control of the actuator without the LabView application can only be done by programming the Arduino board, this procedure is presented below.

Servo motors allow precise positioning and can be controlled using the Servo Arduino library.

A linear regulator can be used to create a 5V secondary source.

Coding of comments is essential to facilitate debugging and sharing.

Program

```
#include            //Servo library
 Servo servo_test;    //initialize a servo object for the connected servo
int angle = 0;
void setup()
{
  servo_test.attach(9);   // attach the signal pin of servo to pin9 of arduino
}
void loop()
{
  for(angle = 0; angle < 180; angle += 1)  // command to move from 0 degrees to 180 degrees
  {
    servo_test.write(angle);      //command to rotate the servo to the specified angle
    delay(15);
  }
  delay(1000);
  for(angle = 180; angle>=1; angle-=5)    // command to move from 180 degrees to 0 degrees
  {
    servo_test.write(angle);        //command to rotate the servo to the specified angle
    delay(5);
  }
    delay(1000);
}
```

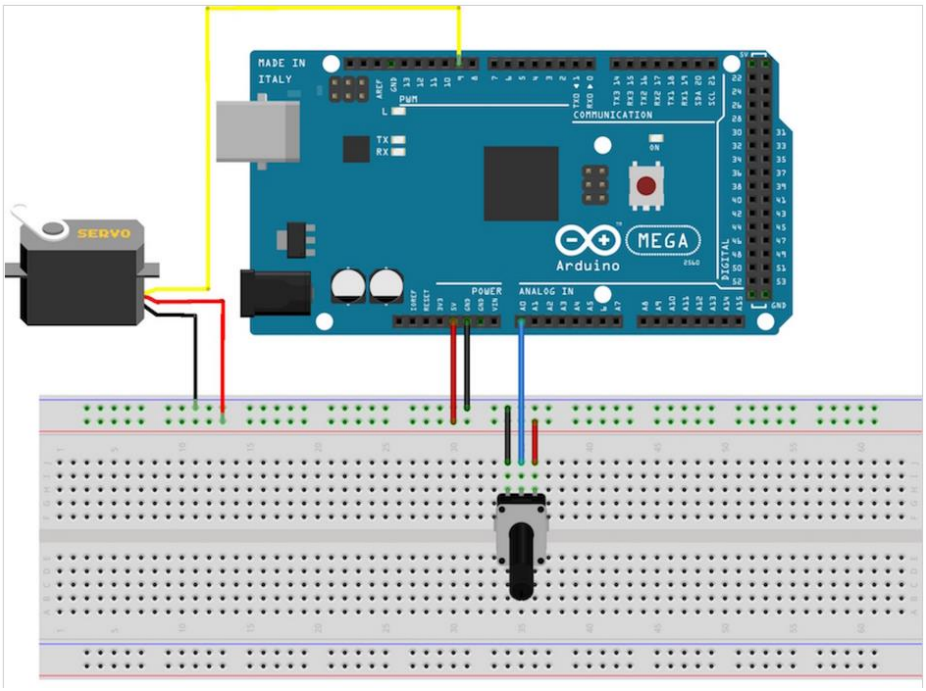

Figure 4. Electrical connection diagram [6]

4. CREATING THE REMOTE LAB FOR DIRECT RUNNING AND REVERSE OF THE SERVOMOTOR

The LabView application allows you to control the remote actuator. So there is no need for the listing of the Arduino program.

In this case, the following steps are taken:

Step 1

We link the LabView application to the Arduino board.

First, we download the "arduino-1.8.4-windows" exe file from the "https://www.arduino.cc/en/ Main / Software" site. Then install it on your computer. At the same time, download and install the "VI Package Manager" software from the "http://www.ni.com/tutorial/ 12397 / en /" website, which contains the program file that provides the LabView interface with arduino (Figure 14).

Step 2

Load the program in Figure 14, in the memory of the Arduino UNO board. This ensures communication between Arduino and LabView.

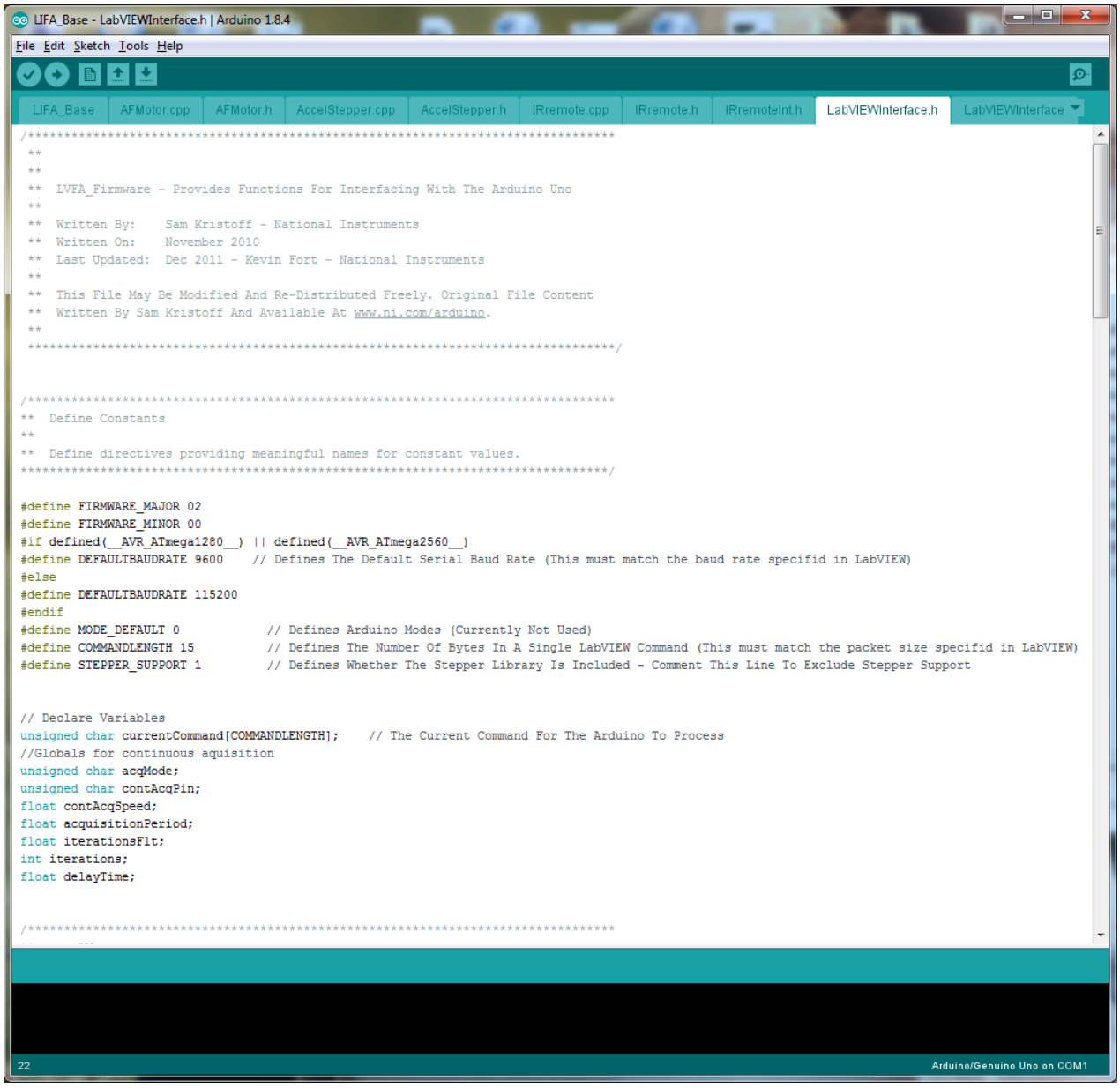

Fig. 5. The LabView interface with arduino.

Step 3

The front panel (Figure 15) which contains the control and acquisition part of the laboratory and is visible to the student, as well as the block diagram (Figure 16), which is designed to create control logic of the actuator, is not visible to the student.

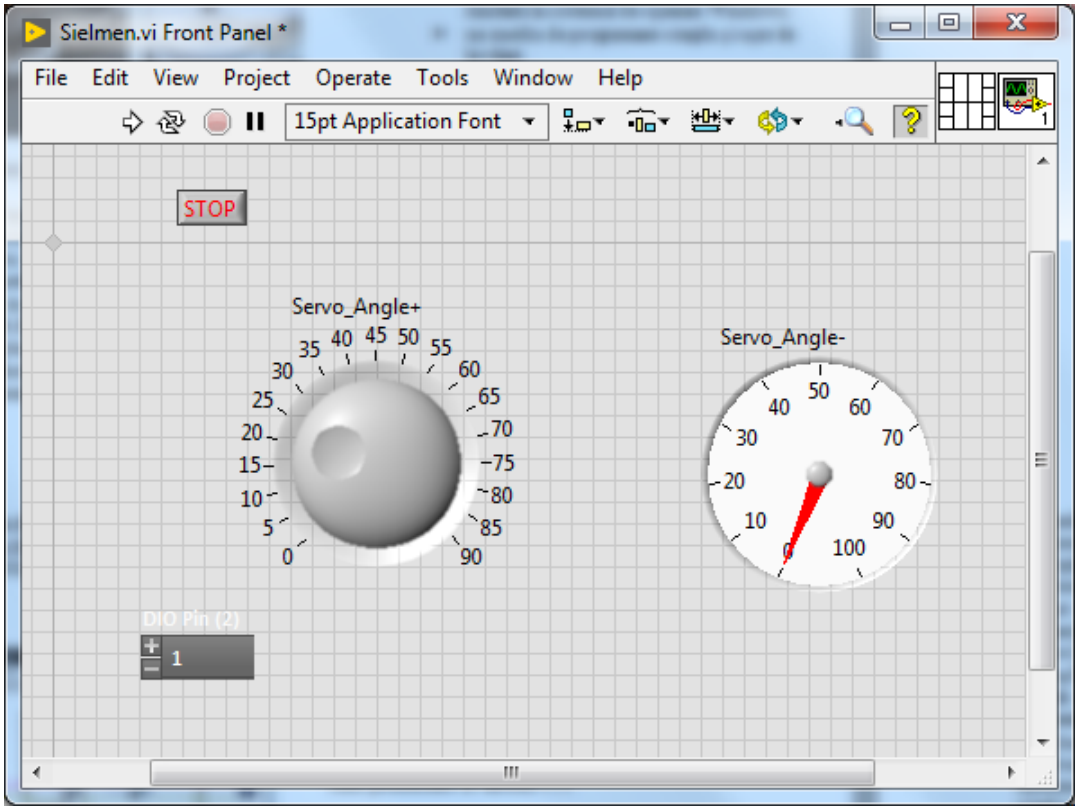

Figure 6. Front panel for servomotor control

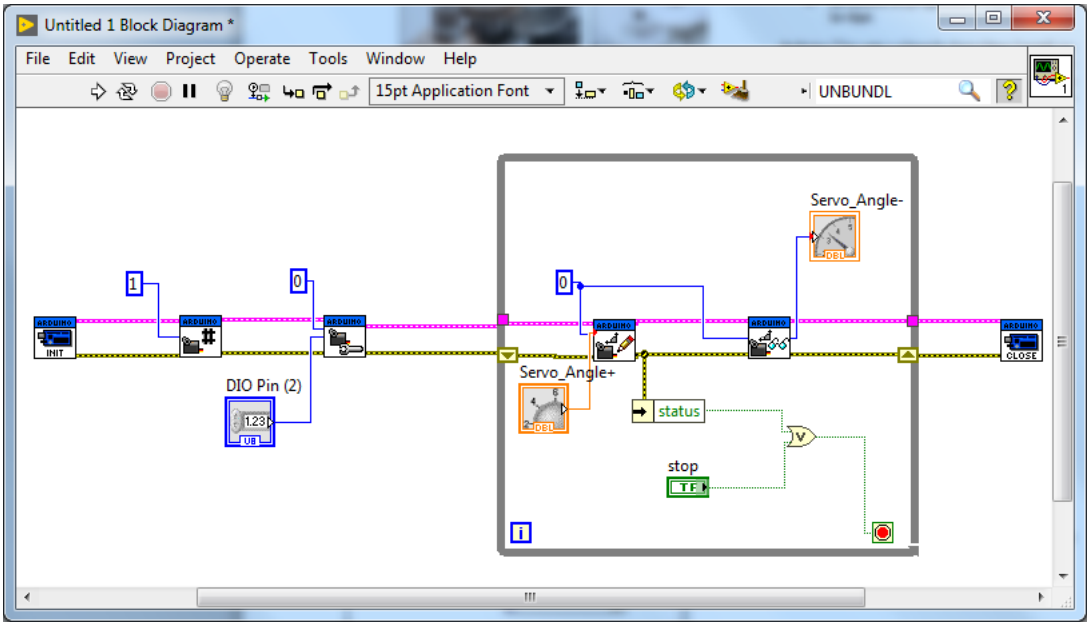

Figure 7. Block diagram for servomotor control

In the front panel for servomotor control, the student can change the direction of rotation of the actuator, the number of steps, the angle of rotation, etc. LabView also allows real physical parameters of the servomotor to be viewed, such as voltage, current, and so on.

In order to pay attention to the laboratory work, each student receives a special task for the study of the servomotor and the results obtained include them in the report and send them to the teacher on the Moodle e-learning platform.

The platform provides online communication between teacher and student. So the student can call on any question at any time. This possibility gives the student the maximum speed to elaborate and present the report of the laboratory work.

5. CONCLUSION

This paper aims to demonstrate the possibility of performing laboratory work at technical disciplines, with the student being at a distance.

Although the control and acquisition of data is virtual, the student can see through a video camera the real physical process that takes place in the laboratory room. Moreover, he can accomplish the work at any time, or may be invited to work with the up-to-date students.

LabView environment allows for any data acquisition and process control. And the arduino plate due to the low cost price makes communication with the physical part of the laboratory work. Therefore, this method of performing laboratory work seems to be the most optimal for the Republic of Moldova.

REFERENCES

1. http://ro.wikipedia.org/wiki/Inginer
2. http://www.researchgate.net/publication/224148635
3. http://www.dpue.energ.pub.ro/Laborator_Informatic/files/info/Laboratorul%206.pdf
4. http://elth.ucv.ro/student1/Cursuri/Bratu%20Cristian/MAP/004%20-%20Curs%20004%20-%20MAP%20-%20Arduino.pdf
5. http://onheli.blogspot.md/2010/02/inside-of-servo.html
6. https://www.allaboutcircuits.com/projects/servo-motor-control-with-an-arduino/
7. Evans Brian W. Arduino Programing Notebook, San Francisco, California, USA 2007