Marijana Pavlov-Kagadejev[*]

DIFFERENT WAYS FOR VISUALIZATION OF DATA OBTAINED FROM AN ARDUINO UNO MICROCONTROLLER BOARD[**]

Abstract

Arduino microcontroller platforms are frequently in use in the last few years. The reason is their simplicity, low cost, and availability of a large number of ready-made solutions on the Internet. The Arduino Integrated Development Environment itself has a Serial Monitor and Serial Plotter tools for data presentation. The Serial Monitor displays the textual and numeric values of the data obtained through the serial port. By using the Serial Plotter, measurement results from the Arduino serial port can be seen in the form of Real-Time diagrams. The deficiency of such a way of data visualization is that it does not provide many opportunities for change and adaptation to the user needs. This paper presents examples of several different programming environments which provide the good opportunities for presentation the results generated by the Arduino boards. Some hardware modules used to display data from the Arduino boards are also described in this paper.

Keywords: Arduino, visualization, measurement, sensor, graph, Real-Time

1 INTRODUCTION

The Arduino-based microcontroller boards are very popular in the last few years due to their low cost, simplicity, and functionality. It is very often used to measure various parameters, as a research tool, or for entertainment [1, 2, 3]. There are a number of low-cost sensors, which can be easily connected to Arduino boards. The obtained results can be archived and later used for various purposes. There are different versions of the Arduino boards like the Arduino Uno, Arduino Mega, Arduino Leonardo, etc. The all are affordable, but due to their price, the most common is the Arduino Uno microcontroller board. The Arduino Uno is easy to operate and there are a number of ready-made software solutions on the Internet that works with this platform.

For the users, who only need to monitor the current value of measured parameters, without archiving data, and with no need for better visualization, it is enough to connect some sensors to the Arduino board, and load the appropriate program (which is easy to find online and modified if needed) using the Arduino IDE (Integrated Development Environment). In such cases, by the use of the IDE tool Serial Monitor, and sending the measured values to the serial port, the results of measurements will appear in the format defined by the program. Another Arduino IDE tool, the Serial Diagram, provides an on-screen presentation of measurements results in the form of time diagrams. Such a way of presenting data is very limited and hardly usable for further

research and analysis. To overcome this problem, different programming environments and hardware modules are used.

In this paper, different ways of data visualization will be shown on a simple example of temperature and humidity measurements using the Arduino Uno microcontroller and DHT22 sensor. Measured data were displayed using the programs created in different programming environments (Processing, Node-red, LabVIEW, and Visual Studio C #) and hardware modules (LCD, 7-segment display, LED, and OLED display). The aim of this paper is to facilitate the selection of programming environments for working with an Arduino board and to give their advantages and disadvantages. Also, the paper aims to display various hardware modules, which are suitable for the self display of data without a PC. This is especially suitable for portable measuring devices development.

2 ARDUINO UNO AND DHT22 SENSOR CHARACTERISTICS

a) b)

Figure 1 *a) Arduino Uno microcontroller board, and b) DHT22 sensor*

The Arduino Uno (Figure 1a) is a microcontroller board with the 14 digital input/output pins (6 can be used as PWM outputs), 6 analog inputs, 16 MHz ceramic resonator (CSTCE16M0V53-R0), USB connection, power jack, ICSP header and reset button. It is based on the ATmega328P microcontroller. This board contains the USB port for simply connection to a computer, or powers it with an AC-to-DC adapter or battery to get started. The Arduino Uno board is the first in a series of USB Arduino boards, and reference model for the Arduino platform. The main characteristics of the Arduino Uno microcontroller are given in the Table 1 [4].

Table 1 *Technical characteristics of the Arduino Uno microcontroller*

Microcontroller	ATmega328
Operating Voltage	5V DC
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) (0.5 KB used by bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

The DHT22 is a digital sensor module (Figure 1b) that consists of a thermistor, for temperature measurement, and capacitive sensor, for determining the relative air humidity. The DHT22 working temperature is from -40°C to + 80°C, and the relative air humidity range is from 0 to 100%. Tempe-rature and humidity measurement accuracy are 0.5° C and 2%, respectively [5]. The DHT22 sensor module with 3 pins has been used in this research. The first pin (shown in Figure 2) is for ground (GND), the second for power supply (VCC), and the third for data transfer (DATA) [6].
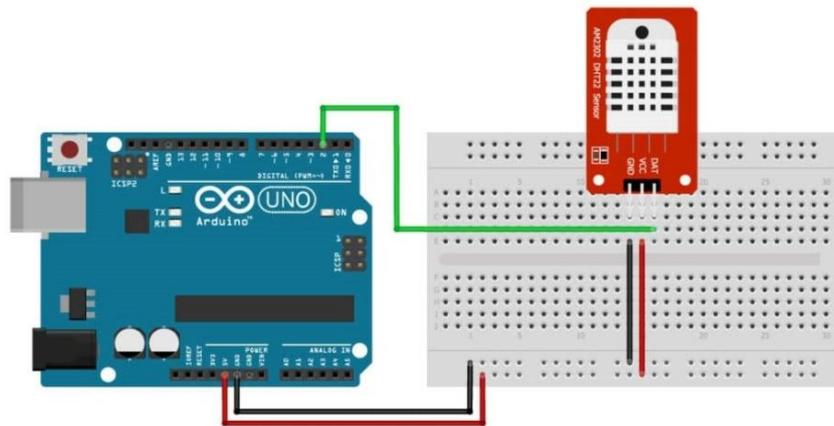

Figure 2 *Wiring DHT22 sensor module to Arduino Uno board*

For the DHT22 sensor module pro-gramming, the DHT sensor library has to be installed by the Arduino Library Man-ager. Also, an adequate code has to be written and uploaded on the Arduino Uno. After that, the measured values of temper-ature and relative air humidity will appear on the Adriano Serial Monitor and Serial Diagram. On the Serial Monitor, the measured values appear in the format de-fined by the code, and time diagrams of the measured parameters appear on the Serial Diagram (Figure 3).
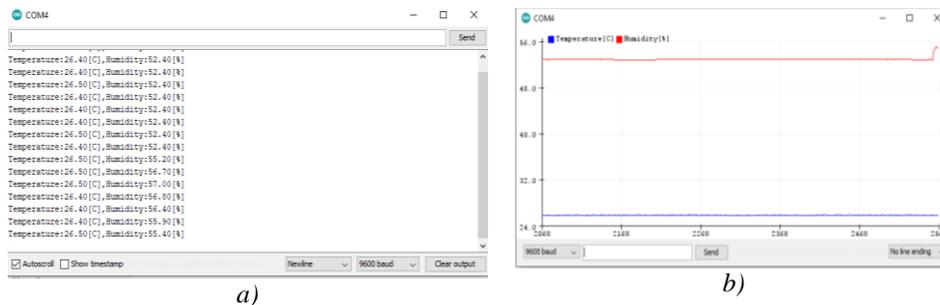

a)

b)

Figure 3 *Data displayed on a) Serial Monitor and b) Serial Plotter on Arduino IDE*

3 SOTFWARE TOOLS FOR DATA PRESENTATION

In order to improve the presentation of data obtained by an Arduino board, a large number of different programming environments may be used. These environments are very different, both in terms of programming, capabilities, and price. Some of them are free, while some are very expensive. In this paper, we will present data measured with Arduino Uno and DHT22 sensor using the programs created in the Processing, Node-red, LabVIEW, and Visual Studio C # programming environments.

3.1 Processing 3

The Processing 3 is an open-source environment, a flexible software sketchbook and language for learning how to code within the context of visual arts [7]. It has been promoting software literacy in visual arts and visual literacy in technology since 2001. Processing is used for learning and prototyping.

In terms of structure, the Processing IDE is similar to the Arduino. It has setup function and draw function like the Arduino has a setup and loop function. The Processing IDE can communicate with the Arduino IDE via serial communication. In this way, we can send data from the Arduino to the Processing IDE, and also from the Processing IDE to the Arduino. The Processing IDE layout is shown in Figure 4.
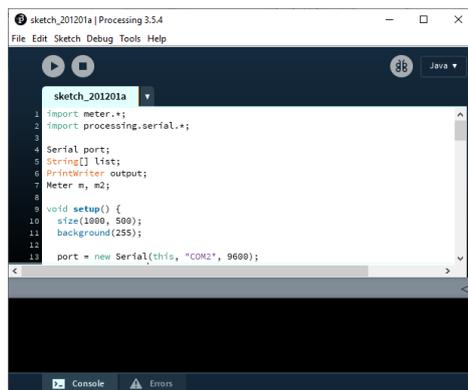

Figure 4 *Processing code*

The Arduino board communicates with the PC using the serial port. Processing sketch can be used to read data from the Arduino board via serial communication by importing a Serial library (Figure 5). This library is already built-in Processing.
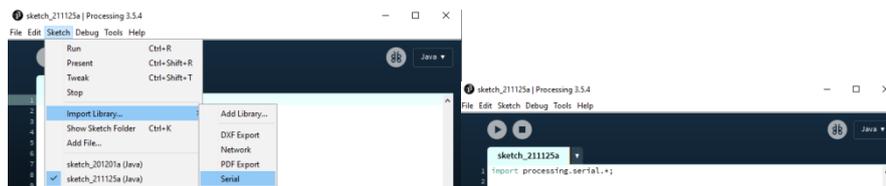

Figure 5 *Importing Serial Library in Processing sketch*

The methods used by the serial communication protocol are described below:

To Import Serial library: `import processing.serial.*;`

To create an object from Serial class: `Serial myPort;`

To list all available serial ports: `println(Serial.list());`

To define port number connected to Arduino board:

`String portName = Serial.list()[0];`

To open the used port and define the Boud Rate:

`myPort = newSerial(this,portName,9600);`

Figure 6 presents an example of displaying data from athe Arduino Uno board and DHT22 sensor module by a program written in the Processing IDE.
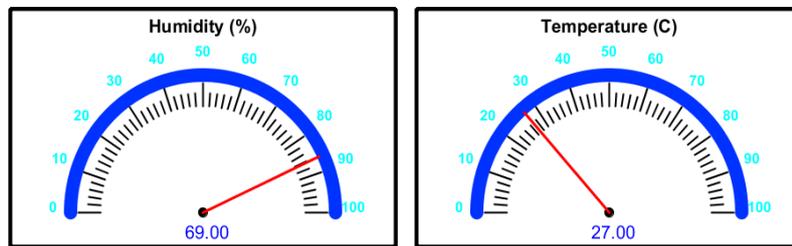

Figure 6 *Arduino data visualization by using Processing script*

3.2 Node-red

The Node-RED is a programming tool that connects hardware devices, APIs, and online services in new ways [8]. It has a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be deployed to the runtime in a single click. The JavaScript functions can be created within the editor using a rich text editor. Useful functions, templates, or flows for re-use can be saved using the built-in library.

The lightweight runtime is built on the Node.js. This makes it ideal to run on a low-cost hardware such as the Raspberry Pi as well as in the cloud. There are over 225,000 modules in the Node package repository. That makes it easy to extend the range of palette nodes to add the new capabilities.

The Node-RED does not have preinstalled nodes for the Arduino, Serial port, and Dashboard. So, these nodes have to be installed with the following commands on the Command Prompt:

`npm install node-red -node-arduino`

`npm install node-red -node-serialport`

`npm install node-red -dashboard`

The Serial port node (Figure 7a) is used to enable the Node-RED serial communication with the Arduino board. It has to be dropped from the Nodes palette into the flow section. By double click on the node, a pop window will appear. The correct number of COM port has to be chosen. The other parameters like the Boud Rate have to be defined in this window (Figure 7b).
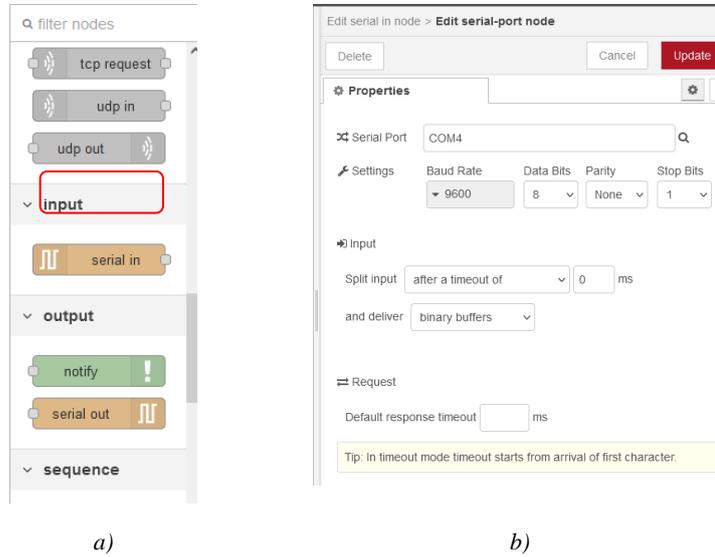
a)   *b)*

Figure 7 *Serial port node object properties in the Node-RED programming environment*

Figure 8 shows an application that displays data from the Arduino Uno board and DHT22 sensor module. The flow dashboard is shown on the left, and the values and graphs of temperature and relative air humidity, obtained from the sensor, are shown on the right side.
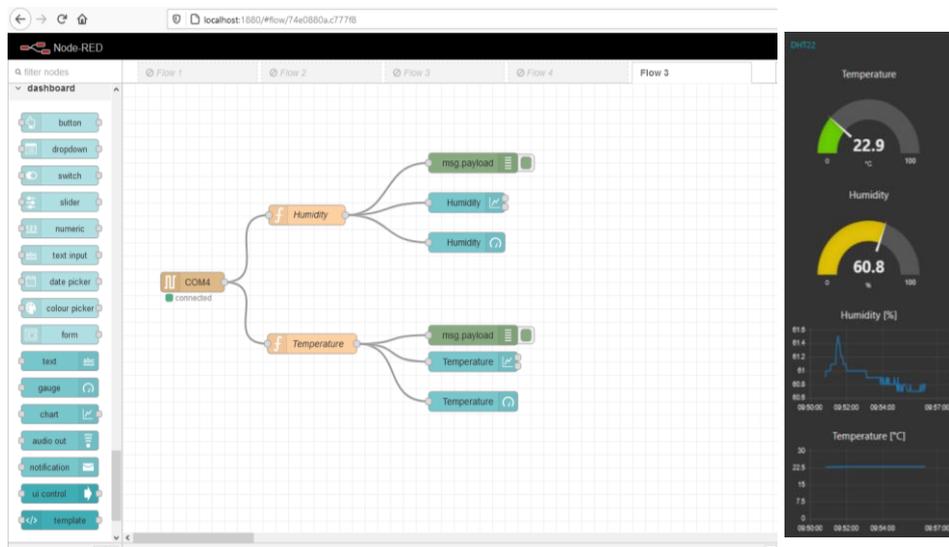

Figure 8 *Node-RED application for data visualization from Arduino board*

3.3 LabVIEW

The LabVIEW is a software dedicated to the measurement applications that require test, measurement, and control with rapid access to the hardware and data insights [9]. It offers a graphical programming approach that helps to visualize every aspect of an application, including hardware configuration, measurement data, and debugging. This makes it simple to integrate measurement hardware from any vendor, represent complex logic on the diagram, develop data analysis algorithms, and design custom engineering user interfaces.

With LabVIEW and adequate hardware, it is easy to build a custom measurement solution to visualize and analyze real-world signals and to make data-driven decisions. Flexible test applications that control multiple instruments can be created with the LabVIEW. Industrial equipment and smart machines can be built faster using the LabVIEW. The LabVIEW can be used for teaching students in the classroom or lab, too.

In our example, the Virtual Instrument Software Architecture (VISA) is used to connect the Arduino board with the LabVIEW. The required object has to be selected from the Block Diagram window as follows: Data Communication > Protocols > Serial (as shown in Figure 9).
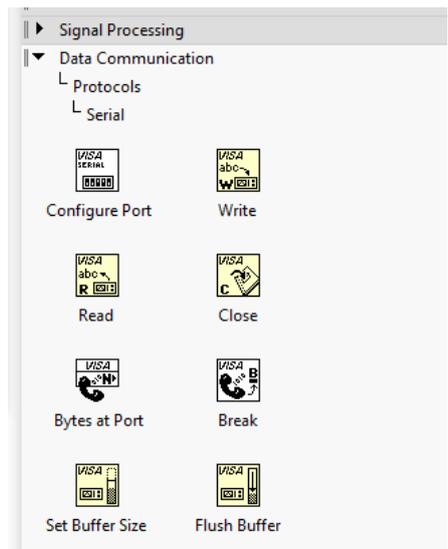

Figure 9 *Virtual Instrument Software Architecture objects window*

A VISA Configure Serial Port object allows users to choose which serial port to use and to set up the Baud Rate. The default Baud Rate is 9600.

An example of a LabVIEW application is shown in Figure 10. Figure 10a shows the Front Panel, and Figure 10b a the Block Diagram of this application. Controls, bar graphs, diagrams, and parameter values obtained from the Arduino Uno board and DHT22 sensor module are shown in the Front Panel. The Block Diagram shows the elements that make up the LabVIEW application.
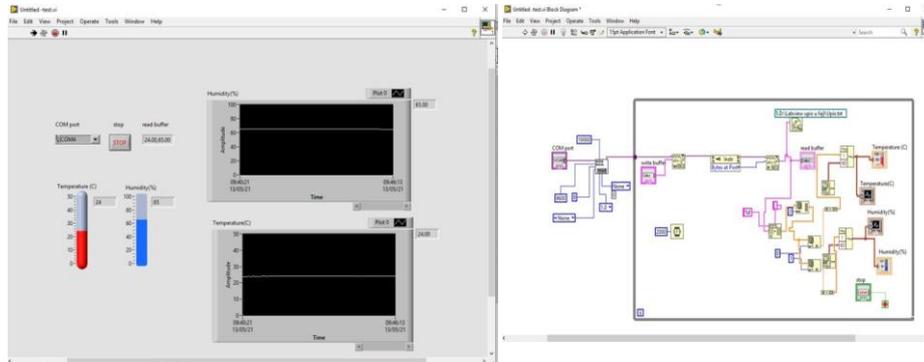
a)   *b)*

Figure 10 *LabView application: a) Front Panel, and b) Block Diagram*

3.4 Visual Studio IDE | C#

The Visual Studio IDE is a creative launching pad that can be used to edit, debug, and build code, and then publish an app. Such IDE can be used for many aspects of software development. Visual Studio includes compilers, code completion tools, graphical designers, and many more features to ease the software development process [10]. Visual Studio IDE| C# is shown in Figure 11.

The Visual Studio Code is a powerful source code editor which runs on a desktop and is available for Windows, macOS, and Linux. It comes with a built-in support for the JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).
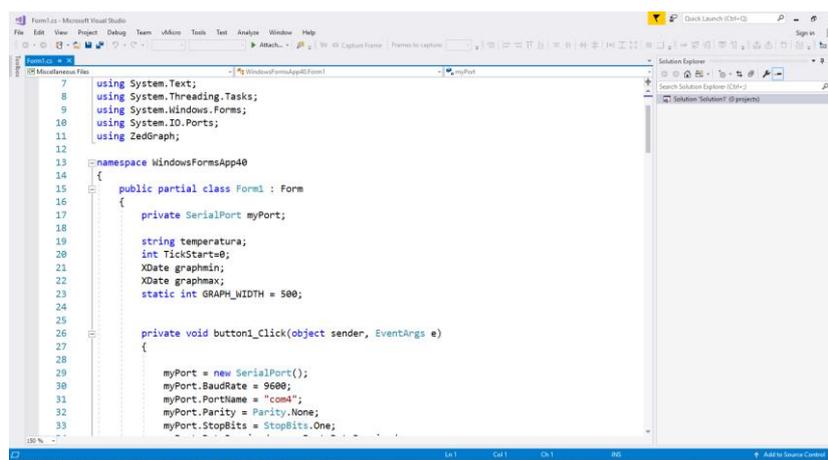

Figure 11 *Visual Studio IDE | C#*

The using statements at the top of code define links to the libraries (called namespace). Using System.IO.Ports has to be written to the top of code. After that, the SerialPort class can be used in the program for serial communication with the Arduino board. This class provides a framework for synchronous and event-driven I/O, access to the pin and break states and access to the serial driver properties [11].

Figure 12 shows how the SerialPort tool can be added to the Windows Form Application. From Toolbox, the Serial Port tool has to be dropped to Form. Properties of this tool will appear by the right click on it.
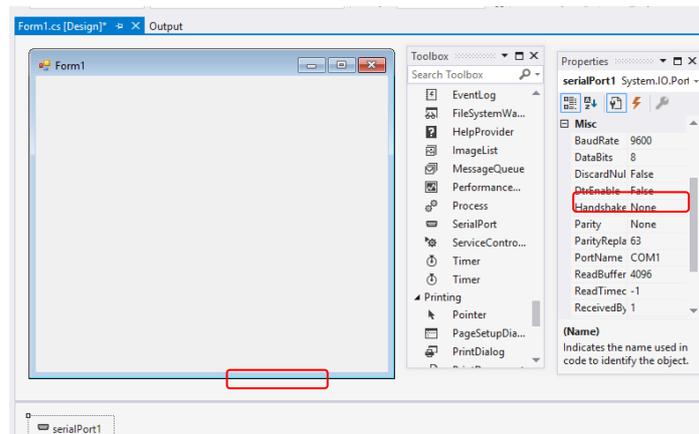

Figure 12 *SeralPort tool with serial port object properties in C#*

The Visual Studio programming IDE is available for Windows and Mac OS. An application for data visualization, obtained from the Arduino board, is developed in the Visual Studio C#. Real-time diagrams and numerical values, obtained from the Arduino Uno board and DHT22 sensor board, are shown in Figure 13.
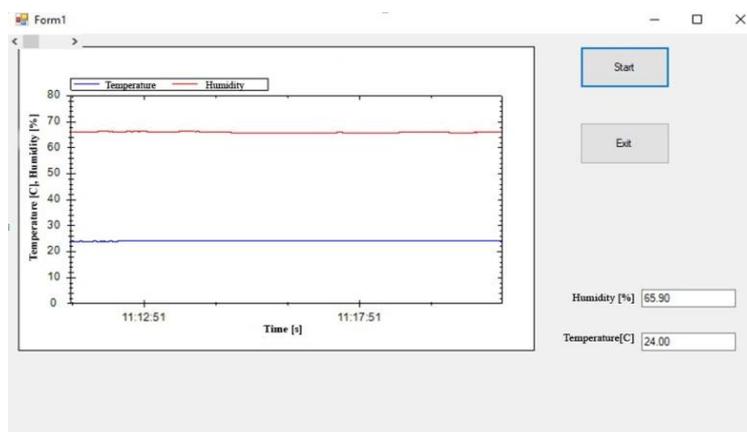

Figure 13 *Visual Studio C# application for data visualization from Arduino board*

4 HARDWARE MODULES FOR DATA DISPLAYING FROM THE ARDUINO BOARDS

There are different types of hardware modules that are used to display data from an Arduino microcontroller. Some of them are the 7 segments, LED, OLED, and LCD displays (Figure 10). The advantage of this way of displaying data is that a PC is not necessary after code has been uploaded to the Arduino.

The measured values are shown on display. This way of displaying data is suitable for the portable measurement devices [12, 13].
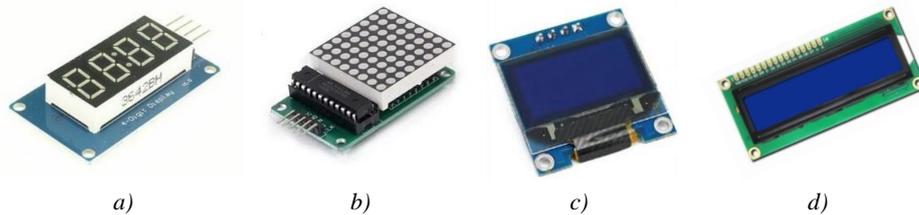

| a) | b) | c) | d) |

Figure 10 *Different types of displays: a) 7 segment, b) LED, c) OLED, d) LCD*

The most popular way for displaying information from the Arduino boards is made using the LCD displays. The LCD displays can be divided into two generic types: characters and graphics. Character LCD is the cheapest and simplest display. It consists of several rows and columns. 1602 LCD Display Module has 2 rows and 16 columns and can display 32 characters. This display is simple and has full compatibility with the all Arduino microcontrollers and processor boards. The 16x2 Arduino LCD Keypad Shield is very popular. It includes and 6 push buttons. Figure 11 shows an example of measurement results presentation by using the 1602 LCD Keypad Shield Module connected on an Arduino Uno board.
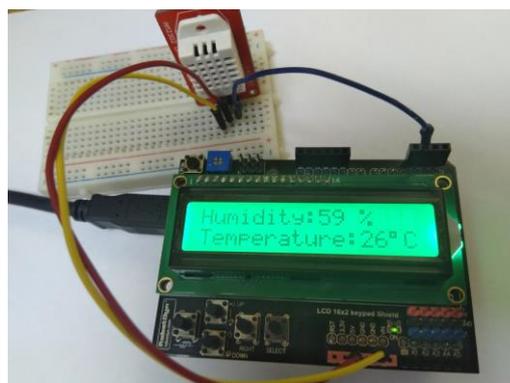

Figure 11 *Measurement results presentation by using the 1602 LCD Keypad Shield Module*

The LCD displays are frequently used for measurement the results presentation in portable measurement devices. An example of using the LCD2004 module (4x20 characters display) in a portable measuring device is given in Figure 12. In addition to the DHT22 sensor, this device has a sensor module for measuring the PM_{10} and $PM_{2.5}$ particle concentrations.

Figure 12 *Using the LCD display in a portable measurement device*

CONCLUSION

This paper presents the different ways for data visualization obtained with an Arduino board. Four different programming environments are used: Processing 3, Node-red, LabVIEW and Visual Studio C #. Processing 3 and Node-red are open-source free environments and there are numerous free examples on the Internet that can make their use easier for users who are not professional software developers. The LabVIEW environment provides a lot of possibilities, but it is necessary to know how this specific environment works, in order to develop the applications. In addition, the LabVIEW is an expensive package and is not affordable to everyone, after a trial period. The Visual Studio C # provides the biggest freedom for developers. However, the cost of the Professional and Enterprise version of this programming IDE is not negligible. For students and for non-commercial purposes, a Community version is available for free.

There are many other different programming tools that can be used to display data obtained by the Arduino boards that are not mentioned in the paper. All users will choose the visualization method that suits them the best, based on their personal requirements, needs, and possibilities. If the user priority is to read the measured values without a PC, then it is most convenient to use the hardware modules to display data from the Arduino boards. Such display modules are the low-cost and their implementation is not so complex for the common user of the Arduino IDE.

REFERENCES

[1] Bist´ak P., Arduino Support for Personalized Learning of Control Theory Basics, IFAC-PapersOnLine ELSEVIER Volume 52, Issue 27, 2019, Pages 217-221

[2] Louis L., Working principle of Arduino and Using it as a Tool for Study and Research, International Journal of Control, Automation, Communication and Systems (IJCACS), Vol.1, No.2, April 2016 DOI: 10.5121/ijcacs.2016.1203 21

[3] Organtini G., Arduino as a Tool for Physics Experiments, IOP Conf. Series: Journal of Physics: Conf. Series 1076 (2018)

[4] https://www.arduino.cc/en/Main/ArduinoBoardUnoSMD

[5] Bogdan M., How To Use The DHT22 Sensor For Measuring Temperature And Humidity With The Arduino Board , Acta Uiversitatis Cibiniensis – Technical Series Vol. LXVIII 2016

[6] https://www.makerguides.com/dht11-dht22-arduino-tutorial/

[7] www: https://processing.org

[8] www: https://nodered.org

[9] www: https://www.ni.com

[10] www: https://visualstudio.microsoft.com

[11] https://docs.microsoft.com/en-us/dotnet/api/system.io.ports?view=dotnet-plat-ext-6.0

[12] Tasić V., Cocić M., Radović B., Božilov A., Apostolovski-Trujić T., Assessment of the Impact of External Pollution by Suspended Particles on the Indoor Air Quality at the Technical Faculty in Bor, Copper, Vol.45., 2 (2020) 19-28 (in Serbian)

[13] Tasić V., Božilov A., Lazović I., Mirkov N., Measurement of CO2 concentrations in the Indoor Air Using the Low-Cost Sensors and Monitors, Copper, Vol.45., 2 (2020) 11-18 (in Serbian)