Journal of Electrical Engineering, Electronics, Control and Computer Science – JEEECCS, Volume 9, Issue 34, pages 9-12, 2023

Integrating Arduino-Based Sensing with LabView Visualisation for Accessible Weather Monitoring

Bogdan-Adrian Enache, Cosmin-Karl Banica
Department of Measurements, Apparatus and Static Converters
Electrical Engineering Faculty, University POLITEHNICA of Bucharest
bogdan.enache2207@upb.ro

Geroge Bogdan Ana
Management Department
SC Business Consulting House SRL
Bucharest, Romania

***Abstract* – This paper presents the development of a weather monitoring system based on an Arduino Board and integrated sensors. This system collects real-time data, particularly temperature and humidity, and sends it to a computer running LabView for visualisation and processing. This novel approach combines the benefits of both worlds, adding the flexibility of an embedded system with the LabView capabilities of data processing and visualisation. By exploring a unique combination of accessible technologies, this paper contributes to a growing body of literature emphasising the democratisation of scientific tools, empowering communities, educators, and researchers to engage in cutting-edge climate analysis and environmental stewardship.**

Keywords-component: weather data, Arduino Uno, LabView, integrated sensors

I. INTRODUCTION

Weather patterns and climate conditions significantly influence the quality of life and human activity [1]. The prevailing means of meteorological data collection relies on sophisticated weather stations that often come with a high cost. This leads to a lack of comprehensive monitoring in some regions, particularly in countries with more pronounced financial constraints and logistical inconveniences [2]. Comprehensive acquisition of meteorological data is vital for the accuracy of weather forecasting [1]. Insufficient data on various meteorological parameters can result in the dissemination of inaccurate weather forecasts, adversely affecting planning and decision-making across multiple sectors, such as agriculture, transportation, and emergency management.

Modern weather forecasting employs computer-calculated models that depend on a rich influx of meteorological data [3]. These models integrate information from several weather stations strategically installed in various locations, thus enabling forecasts over extensive areas.

Environmental sensors are critical in delivering near real-time data for monitoring weather and other environmental phenomena. This task cannot be fulfilled solely through remote satellite methods [4]. With the ongoing expansion of open-source ecological sensor systems, various components and resources have become increasingly accessible. For example, ALog BottleLogger by Northern Widget LLC is an open field sensor data acquisition system, though it does not encompass telemetry features [5]. In contrast, MayFly, an Arduino-based system developed by EnviroDIY, offers telemetry options such as Xbee, LoRa, and 4G, yet its sensor capabilities are confined mainly to water applications [6]. The ThingsBoard, an open IoT rapid development platform, caters to the management and scaling of sensor projects but relies on Raspberry Pi hardware, a computer system not optimised for low-power, minimal-peripheral field sensor applications [7]. A distinct solution, known as WeatherChimes, emerges as a low-cost (with pricing details provided), low-power alternative capable of accommodating various sensors across various applications. Compared to conventional commercial IoT weather stations, WeatherChimes is up to 70% more economical [8-10]. Its lightweight design and compact size also enhance portability and ease of installation.

On the other hand, other enthusiast/hobbyist solutions are also based on Arduino boards and custom-designed graphical user interfaces (GUI), but all are developed especially for a single project, and an adaptation implies severe code modifications [11], [12].

The present research explores an alternative approach to weather data acquisition that leverages accessible technologies such as Arduino boards and LabVIEW, offering a cost-effective and scalable solution. Integrating Arduino microcontrollers with LabVIEW provides a unique blend of simplicity, flexibility, and precision. Arduino, known for its ease of programming and adaptability with various sensors, offers an ideal platform for weather data collection. On the other hand, LabVIEW's graphical programming interface provides an intuitive platform for data analysis and visualisation, making it an attractive tool for developing easily customisable GUIs [13]. By bridging the flexibility of embedded systems with advanced data processing capabilities, this work expands the meteorological toolset, enhances the precision of forecasts, and facilitates broader engagement in environmental monitoring and stewardship.

The rest of the paper is organised as follows: Section 2 comprehensively delineates the hardware and software configuration, while Section 3 presents and analyses the results. The article ends with a conclusion chapter and future developments.

II. Hardware and Software Setup

The block diagram of the proposed system used to monitor the weather conditions is presented in Fig. 1.

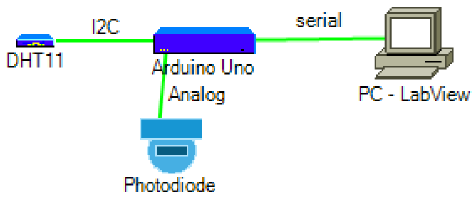


Figure 1. Block diagram of the proposed system

This system measures the temperature, relative humidity (RH) and a specific location's illuminance level.

A. *Hardware setup*

The system's main parts are the Arduino Uno board, a DHT 11 humidity and temperature integrated sensor and a photodiode for determining the illumination level.

The DHT11 is a digital transducer for measuring the temperature and humidity of the ambient air. This sensor employs a capacitive humidity sensor in conjunction with a thermistor and an integrated 8-bit microcontroller. It produces an I2C digital output assuring an ±0.5°C and ±1% RH absolute accuracy [3].

The photodiode is a light-sensitive semiconductor diode. It produces current based on the level of illumination. It has an analogue output and a 10% absolute accuracy, which is very high, but in this case, it is used only to determine if there is day or night.

The Arduino board is used as an integrator for all the systems' components. It collects the different format data and sends it to the LabView computer for processing and visualisation.

B. *Software setup*

Arduino offers a programming environment (Arduino IDE) characterised by its simplicity and accessibility, featuring a syntax that resonates with established programming languages such as C and C++. These characteristics make Arduino an appealing choice for a diverse audience, including hobbyists, students, and amateur innovators. However, developing academic or professional applications that require GUIs is a real challenge [12].

To overcome this issue, a cross-platform software solution was developed. It is based on MakerHub Linx firmware, which is now part of the LabView Hobbyist Toolkit [14]. Linx provides a suite of user-friendly LabVIEW Virtual Instruments (VIs) that facilitate interaction with prevalent embedded platforms, including but not limited to Arduino, chipKIT, and myRIO. There are VIs especially developed for an extensive array of sensors that enable rapid data acquisition and transmission to personal computers. At the same time, the peripheral VIs provide comprehensive access to device functionalities such as digital I/O, analogue I/O, SPI, I2C, UART, PWM, and additional features. For this project, a more generic approach was used, considering the different types of outputs used by the sensors. This approach is based on custom commands, which are snippets of code used by the Arduino to communicate with the Linx Listenres. The developed function for weather data acquisition is presented below.

```
int WeatherDaq(unsigned char numInputBytes,
unsigned char* input, unsigned char*
numResponseBytes, unsigned char* response)
{
  int value = analogRead(0);
  response[0] = (int)  dht.readTemperature();
 response[1] = (int) dht.readHumidity();
 response[2] = map(value,0,1023,0,255);
 *numResponseBytes = 3;
 return 0;
 delay(500);
}
```

The functions use four parameters: *unsigned char*, *unsigned char**, *unsigned char**, *unsigned char**, which establish the communication with the Linx Listener. The first two parameters specify the *numInputBytes* is the number of bytes in the input array (from LabVIEW) and the data bytes sent using the Custom Command VI. The last two parameters set the size of the data packet sent to LabView and the response array.

The function's response should be an int value, and all the values returned to LabView should be integers, too. Because this feature is fixed, it causes a loss in accuracy, especially for the temperature, which now has a ±1°C. Another aspect that should be taken into consideration is that the 10-bit Analog-to-Digital (ADC) converter used by Arduino returns a value between 0 to 1023, which should be converted to an int for transmission, and to avoid severe loss of accuracy, the *map()* function is used to transpose the ADC response to the 0 - 255 range.

The cross-platform nature of the project is given by the synergic use of two mature technologies, Arduino IDE and LabView. So, the custom command function developed in Arduino should have a counterparty in LabView.

The block diagram of the developed system is presented in Fig. 2.

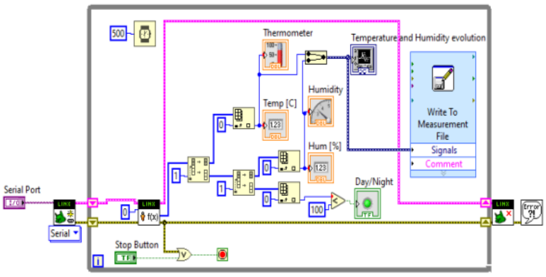


Figure 2. LabView block diagram

From the Linx library, there are only three Vis used. The first is *Open. vi*, which establishes a serial connection to the Arduino board. It requires the used COM port, the speed and parity of the connection. After enabling this connection, the data is fed to the program's loop, where *CustomCommand.vi* acts as a listener. This VI requires, as input the data stream from the serial connection, the number of the custom command function from Arduino code (0 in this case) and provides the array containing the temperature, humidity and illuminance level as output. The *SplitArray.vi* is used to extract each set of data. The *Close. vi* is placed outside the loop and terminates the serial connection to the Arduino board when an error is caught or the user presses the Stop button. The while loop of the program is run every 500 ms, which is set by the *Wait. vi*. This wait value should match the *delay()* value from the Arduino code for stability reasons.

The main advantage of this structure is that it doesn't require any modification if the Arduino sensors are changed. It only displays the three values coming from the Arduino custom command function, making it versatile and easily extendable.

In contrast to other methodologies found in existing literature [15], [16], the approach delineated in this study minimizes reliance on the existing Virtual Instruments from the Linx library, utilizing primarily the communication modules. Instead, it emphasizes an extensive use of Arduino code, thereby benefiting from an expansive selection of libraries designed for various sensors and the support of an extensive enthusiast community.

III. Results and Discussions

The proposed system is presented in Fig. 3.

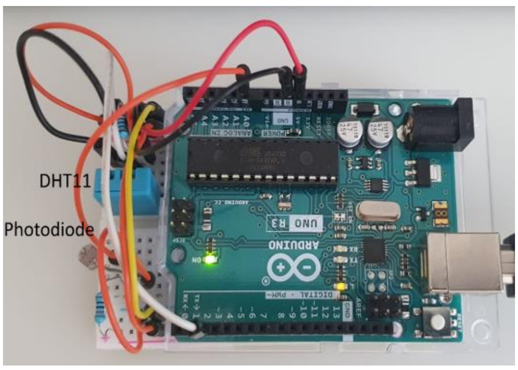


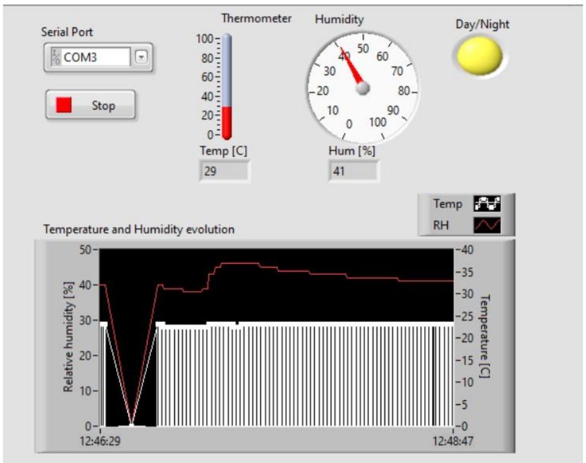
Figure 3. Proposed system

This system was used to acquire data from several moments of the day and for several periods. The longest online time was six hours, but based on the observed data, there is no problem with keeping the system active for several days.

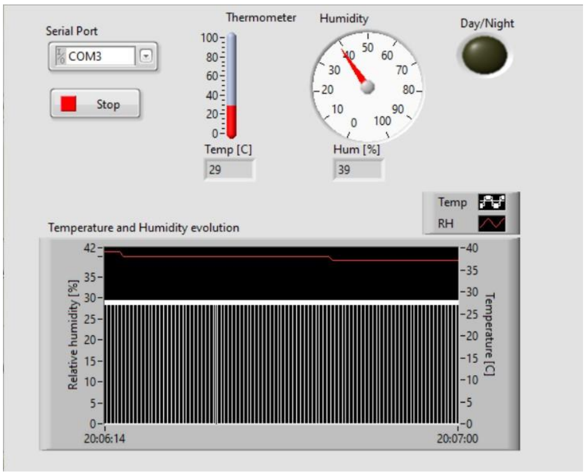
All the recorded data is stored in a .csv text file for comparison and analysis.

To validate the proposed architecture, several recordings were performed for different periods of the day and on other days.

Fig. 4 a) and b) present the most representative data.



a) System recordings from the day period



b) System recordings from the night period

Figure 4. The front panel of the proposed system

To facilitate the observation of the recorded values, the relative humidity was plotted using a solid red line, while the temperature was represented using a bar graph.

All the tests were performed in a semi-enclosed area, where air drifts had a relatively low influence. With all this in the day recordings, there are some variations in the relative humidity caused by the air movement.

A snippet of one of the .csv files recorded is presented in Fig. 5.

Time	Temperature	Humidity
0	0	0
0.474732	0	0
0.973703	0	0
1.474277	0	0
1.974384	0	0
2.501801	27	41
2.97689	27	41
3.477726	27	41
3.977092	27	41
4.475577	27	41
5.001192	27	41

Figure 5. Test recordings

These .cvs files are recorded one for each day and are time-stamped with the time when the system was started. Based on this information, the time of each measurement can be deduced from the time column.

ACKNOWLEDGMENT

The work presented in this paper is part of the project Innovative Smart Digital Platform (ISDP) SMIS code 2014+ 142654.

CONCLUSION

This paper detailed the development of a cross-platform based on Arduino and LabView. This combination offers a democratised scientific tool that is a testament to the shift towards making advanced technological research and environmental monitoring accessible to wider audiences. Such advancements promise to pave the way for enhanced community involvement in climate studies and foster a greater sense of environmental responsibility among educators, researchers, and the general public. Considering labView's advanced capabilities in data processing, future developments of the project will aim to incorporate machine learning algorithms. Such enhancements are intended to forecast future weather conditions by leveraging both historical and contemporary data, thereby augmenting the system's predictive capacities.

REFERENCES

[1] H. Thieblemont, F. Haghighat, R. Ooka, and A. Moreau, “Predictive control strategies based on weather forecast in buildings with energy storage system: A review of the state-of-the art,” *Energy Build.*, vol. 153, pp. 485–500, Oct. 2017, doi: 10.1016/j.enbuild.2017.08.010.

[2] M. Mancasi, R. Vatu, O. Ceaki, R. Porumb, and G. Seritan, “Evolution of smart buildings. A Romanian case,” in *2015 50th International Universities Power Engineering Conference (UPEC)*, IEEE, 2015, pp. 1–4.

[3] M. Kusriyanto and A. A. Putra, “Weather Station Design Using IoT Platform Based On Arduino Mega,” in *2018 International Symposium on Electronics and Smart Devices (ISESD)*, Bandung: IEEE, Oct. 2018, pp. 1–4. doi: 10.1109/ISESD.2018.8605456.

[4] O. C. Novac, I.-I. Mureșan, M. C. Novac, M. Oproescu, C. Gordan, and G. Bujdosó, “Development of a weather application in android and IOS operating systems,” in *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, 2019, pp. 1–5.

[5] “Droughts & Floods Assessment and Monitoring using Remote sensing and GIS.” http://www.wamis.org/agm/pubs/agm8/Paper-14.pdf

[6] “EnviroDIY.” https://github.com/EnviroDIY

[7] “ThingsBoard Open-Source IoT Platform.” https://thingsboard.io

[8] W. Woo, W. Richards, J. Selker, and C. Udell, “WeatherChimes: An Open IoT Weather Station and Data Sonification System,” *HardwareX*, vol. 13, p. e00402, Mar. 2023, doi: 10.1016/j.ohx.2023.e00402.

[9] S. T. Nicolescu *et al.*, “The Optimization of a Real-Time Indoor Air Quality Monitoring System,” in *2022 E-Health and Bioengineering Conference (EHB)*, IEEE, 2022, pp. 1–4.

[10] H. Ucgun and Z. K. Kaplan, “Arduino based weather forecasting station,” in *2017 International Conference on Computer Science and Engineering (UBMK)*, Antalya, Turkey: IEEE, Oct. 2017, pp. 972–977. doi: 10.1109/UBMK.2017.8093397.

[11] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, and S. Vappangi, “A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations,” *Comput. Sci. Rev.*, vol. 40, p. 100364, May 2021, doi: 10.1016/j.cosrev.2021.100364.

[12] G.-C. SERIȚAN, B.-A. Enache, I. Vilciu, S.-D. Grigorescu, and V. Mladenov, “COMPARISON STUDY OF TOP DEVELOPMENT BOARDS IN THE CONTEXT OF IOT,” *Rev. Roum. Sci. Tech. ÉLECTROTECHNIQUE ÉNERGÉTIQUE*, vol. 67, no. 4, pp. 483–486, 2022.

[13] L. S. Csokmai, C. M. Novac, O. C. Novac, G. Bujdosó, M. Oproescu, and M. Codrean, “Comparative study about data speed acquisition and recording in a MySQL database of LabVIEW, MATLAB and Python programming languages,” in *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, 2021, pp. 1–4.

[14] “LINX”, https://www.labviewmakerhub.com/doku.php?id=libraries:linx:start

[15] Marsh, Jackson, Christy Dunlap, Stephen Pierson, and Han Hu. "Introducing LabVIEW and Arduino as Data Acquisition System Alternatives." ,*2022 ASEE Midwest Section Conference*.

[16] El Hammoumi, A., Saad Motahhir, A. Chalh, A. El Ghzizal, and A. Derouich. "Real-time virtual instrumentation of Arduino and LabVIEW based PV panel characteristics." In *IOP Conference Series: Earth and Environmental Science*, vol. 161, no. 1, p. 012019. IOP Publishing, 2018.