CONSIDERATIONS REGARDING ATMOSPHERIC AND ENVIRONMENTAL DATA ACQUISITION WITH QUADCOPTER UAV

Vasile PRISACARIU, Răzvan MĂNESCU

"Henri Coandă" Air Force Academy, Brașov, Romania (prisacariu.vasile@afahc.ro, razvan.manescu@afahc.ro)

DOI: 10.19062/2247-3173.2025.26.21

Abstract: *The specific applications of atmospheric and environmental data acquisition are found in the following areas: air quality monitoring, climate and meteorological studies, extreme events (volcanic eruptions, fires, and floods), biodiversity, waste and soil pollution, precision agriculture (crop monitoring).*

The paper proposes an applicative approach regarding the monitoring and acquisition of atmospheric and environmental data with the help of open source hardware tools (arduino platform) on board a customized UAV, as a low-cost functional model, with educational and research use.

Keywords: *atmospheric monitoring, UAV, Arduino, temperature sensor, smoke sensor.*

Acronims:

ppm	parts per million	csv	comma-separated values
IDE	Integrated Development Environment	SPI	Serial Peripheral Interface

1. INTRODUCTION

The use of unmanned aircraft for the acquisition of atmospheric and environmental data offers specific approaches to the versatility of these types of aerial vectors together with the technological advantages of the miniaturization of dedicated sensors depending on the specific national regulations applied to the categories of UAV used.

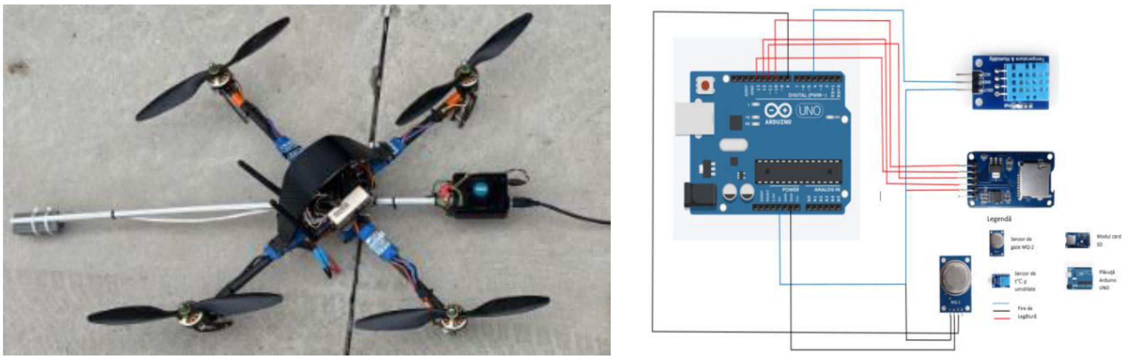
The specific applications of atmospheric and environmental data acquisition are found in the following areas: air quality monitoring, climate and meteorological studies, extreme events (volcanic eruptions, fires, floods), biodiversity, waste and soil pollution, precision agriculture (crop monitoring). [1-5]

The paper proposes an applicative approach to the monitoring and acquisition of atmospheric and environmental data using open source hardware tools (Arduino platform) on board a customized drone, as a low-cost functional model, with educational and research use. The theoretical and applicative aspects developed in the article include a series of references about the tools used in the application stage and obviously an applied approach with the presentation of concrete experimental data on the sampled atmospheric and environmental parameters.

2. RESOURCES

2.1.Hardware

For the acquisition of atmospheric and environmental data we used a quadcopter to which a DHT11 temperature and humidity sensor and an MQ-2 gas sensor were attached. These were integrated into the unmanned aerial platform using an Arduino UNO microcontroller, see Fig. 1. [6-8]



a b

FIG.1 Resources, a. multicopter, b. microcontroler ARDUINO Uno

To initialize the experimental model, I used the Arduino IDE to program the sensor array to deliver data in an easy-to-read and process manner. Humidity is measured in percentage, air temperature and dew point temperature in Celsius degrees, and gases are measured in parts per million (ppm). The Arduino microcontroller is responsible for communicating with the sensors through various protocols, processing the acquired data, and, in my case, storing it on the SD card in csv format. [9]

2.2.Software

The software implementation of the monitoring system was done in the Arduino-specific language, based on C/C++. This code can be divided into several sections: initialization and configuration of components, acquisition of data from sensors, processing and calculation of derived parameters, data storage and display.

a. Initializing and setup elements

This section consists of configuring the libraries used, declaring the pins used in communicating with the hardware components, and initializing global variables. Without the use of libraries, it would not have been possible to run the code below:

```
#include <DHT.h>         // Biblioteca pentru senzorul DHT11
#include <SPI.h>         // Biblioteca pentru comunicarea SPI (necesară pentru SD card)
#include <SD.h>          // Biblioteca pentru modulul SD card
#include <math.h>        // Biblioteca pentru funcții matematice (necesară pentru calculul punctului de rouă)
```

The DHT library facilitates the microcontroller's communication with the temperature and humidity sensor, and the SPI and SD libraries are essential for the optimal functioning of the SD card. The math library provides the mathematical functions necessary for calculating the dew point. We used preprocessing directives to define the pins used in communicating with the hardware components.

This approach allows for easy modification of the hardware configuration without much code modification.

```
#define DHT_PIN 2          // Pinul digital pentru senzorul DHT11
#define DHT_TYPE DHT11     // Tipul senzorului DHT (DHT11)
#define MQ2_PIN A0         // Pinul analog pentru senzorul MQ2
#define SD_CS_PIN 10       // Pinul CS pentru modulul SD card
```

The setup( ) function is important because it provides a robust solution for implementing the SD card, which includes configuring pin 10 as an output and setting it *high*.

```
pinMode(10, OUTPUT);
digitalWrite(10, HIGH);
```

This solves a well-known SD card initialization problem. At the same time, the SD card initialization has been done at a reduced speed to increase the stability of SPI communication.

```
if (!SD.begin(SD_CS_PIN, SPI_HALF_SPEED)) {
  Serial.println("Eroare la inițializarea cardului SD!");
  Serial.println("Programul va continua fără înregistrarea datelor pe card SD.");
  cardSDFunctional = false;
} else {
  Serial.println("Card SD inițializat cu succes.");
  cardSDFunctional = true;
```

An advantage of this code is that it can be used to read data even if an SD card is not used.

b. Data acquisition from sensors

Acquisition of data from sensors is done in the loop() function which is also the main loop of the program. In order not to overload the system and to record data at a certain period, we implemented a timing mechanism based on the millis() function:

```
unsigned long timpCurent = millis();

if (timpCurent - timpAnterior >= interval) {
  timpAnterior = timpCurent;
```

Thus, data recording is done once every 5 seconds. With the help of the call of the readhumidity() and readtemperature() functions, data acquisition from the DHT11 sensor is performed:

```
umiditate = dht.readHumidity();
temperatura = dht.readTemperature();
```

At the same time, we also introduced checks to detect reading errors of the DHT11 sensor.

```
if (isnan(umiditate) || isnan(temperatura)) {
  Serial.println("Eroare la citirea senzorului DHT11!");
  return;
}
```

Information acquisition from the MQ-2 sensor is achieved by reading the value from the corresponding pin.

```
valoareMQ2 = analogRead(MQ2_PIN);
```

3. PARAMETER PROCESSING

3.1. Processing and calculation of parameter

The calculation of the dew point temperature is performed by the function calculaeazaPunctRoua() based on a mathematical formula that uses the temperature measured in degrees Celsius but also the humidity recorded in percentage. The dew point temperature represents the temperature at which water vapor condenses.

```
float calculeazaPunctRoua(float celsius, float umiditate) {
  // Implementarea metodei dewPointFast - mai rapidă și suficient de precisă pentru majoritatea aplicațiilor
  float a = 17.271;
  float b = 237.7;
  float temp = (a * celsius) / (b + celsius) + log(umiditate * 0.01);
  float Td = (b * temp) / (a - temp);
  return Td;
}
```

This implementation uses the *dewPointFast method* which is an efficient and highly optimized method, being faster than other methods of calculating the dew point temperature.

Using a simplified conversion formula, the system converts the analog value read from the MQ2 sensor into ppm units.

```
ppm = valoareMQ2 * 10.0;  // Conversie simplă pentru demonstrație
```

3.2. Storage and display data

The system provides two methods for displaying acquired data: displaying on the Arduino IDE application interface (Serial Monitor) and storing on the SD card. [9]

Displaying data on the serial interface allows for real-time monitoring.

```
// Afișarea datelor pe Serial Monitor
Serial.print("Temperatura: ");
Serial.print(temperatura);
Serial.print(" °C, Umiditate: ");
Serial.print(umiditate);
Serial.print(" %, PPM: ");
Serial.print(ppm);
Serial.print(", Punct de Rouă: ");
Serial.print(punctRoua);
Serial.println(" °C");
```

Data is stored on the SD card only after it has been successfully initialized, using the variable 'cardSDFunctional' to control this action. With this code, the data is saved in CSV format, meaning the values are separated by commas, which facilitates easy data entry into Excel or other special statistical analysis software.

The code implementation includes several optimizations worth mentioning, such as: using the timing mechanism based on 'millis()' instead of the 'delay()' function allows for a more efficient operation of the system in terms of energy consumption. At the same time, organizing the code into different functions to perform certain operations increases modularity and allows for very easy system changes.

```
// Salvarea datelor pe cardul SD doar dacă acesta funcționează
if (cardSDFunctional) {
  File dataFile = SD.open(numeFisier, FILE_WRITE);

  if (dataFile) {
    // Scriere date în format CSV
    dataFile.print(temperatura);
    dataFile.print(",");
    dataFile.print(umiditate);
    dataFile.print(",");
    dataFile.print(ppm);
    dataFile.print(",");
    dataFile.println(punctRoua);
    dataFile.close();

    Serial.println("Date salvate pe cardul SD.");
  } else {
    Serial.println("Eroare la deschiderea fișierului pentru scriere!");
  }
}
```

4. EVALUATION OF THE EXPERIMENTAL MODEL

4.1 Cost and mass of the atmospheric sensor system

The atmospheric sensor system consisting of an Arduino UNO and the DHT-11 and MQ2 sensors is an economical and efficient choice because the sensors used have a low cost. At the same time, they are recommended for applications that do not require a very high degree of accuracy. [10-15]

Table1. Costs and weight of sensors in the atmospheric monitoring module

Elements	Cost (ron)	Mass (grams)
Arduino UNO	133,24	9
DHT-11 sensor	8,36	2
MQ2 sensor	10,92	8
SD card module	7,14	8
Alluminium stick	50	150
Arduino case	20	40
Senzor case	0,2	25
TOTAL	**229,86**	**242**

4.2. Atmospheric monitoring module power consumption

Estimating the power consumption of this system is essential to know which battery to use and the autonomy it will have. Since this sensor module receives power from the same battery that powers the drone, a battery with a sufficiently large capacity must be chosen so as to meet the needs of each device.

Table 2. Sensor module power

Elements	Power (mA)
Arduino UNO	25-50 (Max)
DHT-11 sensor	2,5 (Max)
MQ2 sensor	180 (Max)
SD card module	20-100 (Max)
TOTAL	**332,5 (Max)**

5. CONCLUSIONS AND FUTURE STUDY

Atmospheric and environmental monitoring through the acquisition of specific data involves the use of calibrated sensors and measurement scenarios adapted to the purpose of experimental tasks. Given the logistical preparation carried out, a series of future stages can be designed as follows: sensor calibration, definition of work scenarios and measurement stages, stages that are the subject of future scientific work.

Educational unmanned aerial platforms constitute a mature technology that can instrument appropriate measurements in experimental scenarios with the delivery of data with medium and high confidence levels.

REFERENCES

[1] H. Sun, H. Yan, M. Hassanalian, J. Zhang, A. Abdelkefi, UAV Platforms for Data Acquisition and Intervention Practices in Forestry: Towards More Intelligent Applications. *Aerospace* **2023**, *10*, 317. https://doi.org/10.3390/aerospace10030317;

[2] Vishal, Manuj Sharma, Suresh Jain, Unmanned aerial vehicles and low-cost sensors for air quality monitoring: A comprehensive review of applications across diverse emission sources, Sustainable Cities and Society, Volume 127, 2025, 106409, ISSN 2210-6707, https://doi.org/10.1016/j.scs.2025.106409;

[3] Atmospheric monitoring, available at https://bst.aero/atmospheric-monitoring/, accessed at 12.04.2025;

[4] J. Greenfield and C. Thaxton, Data Acquisition and Logging System for a Drone-mounted Atmospheric Sensor Suite, *SoutheastCon 2022*, Mobile, AL, USA, 2022, pp. 178-183, doi: 10.1109/SoutheastCon48659.2022.9764113;
[5] S. S. Singh *et al*., Real-Time Monitoring of Atmospheric Air Pollutants using Sensor Integrated UAV, *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, Greater Noida, India, 2024, pp. 1-6, doi: 10.1109/ICEECT61758.2024.10739167;
[6] M. E. Irfan Bin Edi, N. Emileen Abd Rashid, N. N. Ismail and K. Cengiz, Low-Cost, Long-Range Unmanned Aerial Vehicle (UAV) Data Logger Using Long Range (LoRa) Module, *2022 IEEE Symposium on Wireless Technology & Applications (ISWTA)*, Kuala Lumpur, Malaysia, 2022, pp. 1-7, doi: 10.1109/ISWTA55313.2022.9942751;
[7] J. P. Grinias, J. T. Whitfield, E. D. Guetschow, and R.T. Kennedy, An Inexpensive, Open-Source USB Arduino Data Acquisition Device for Chemical Instrumentation, *Journal of Chemical Education* 2016 *93* (7), 1316-1319 **,** DOI: 10.1021/acs.jchemed.6b00262;
[8] B. Morawski & D. Głowacki & A. Głowacka, (2019). Low-Cost Data Acquisition Unit for Flight Tests. Fatigue of Aircraft Structures. 2019. 121-130. 10.2478/fas-2019-0012;
[9] ARDUINO IDE, https://www.arduino.cc/en/software/, accessed at 02.05.2025;
[10] Arduino Uno features, https://docs.arduino.cc/hardware/uno-rev3/, accessed at 02.05.2025;
[11] DHT-11 sensor, https://ardushop.ro/en/electronics/2306-dht11-digital-temperature-and-humidity-sensor-6427854010582.html, accessed at 08.05.2025;
[12] MQ-2 sensor, https://sigmanortec.ro/Senzor-gaz-MQ-2-p126093711, accessed at 08.05.2025;
[13] Modul micro SD, https://roboromania.ro/produs/modul-micro-sd-storage/, accessed at 08.05.2025;
[14] ARDUINO Uno, available at https://store.arduino.cc/products/arduino-uno-rev3?srsltid=AfmBOopaZBj_XkhVOQz9iJA4IF81fnmhRKXVEGklX8p51hq0DP3SMkaj, accessed at 08.05.2025;
[15] A. Saputro & M. Yantidewi, Analysis of Air Temperature and Humidity in Kedunggalar Against BMKG Data Based on DHT11 Sensor. Journal of Physics: Conference Series. 1805. 012045. 10.1088/1742-6596/1805/1/012045, 2021.