MODULE IV

Lesson Plan 3

Summary

Title

Exploring Wind Power: Build Your Own Anemometer

Duration	45 minutes
Student age range	10-14

Summary of the topic

Wind energy is generated from the wind and is a clean and sustainable energy source. It is currently one of the planet's largest producers of clean energy. However, although it has many advantages, such as a low carbon footprint and no emission of particles into the atmosphere, one of its most significant disadvantages is its dependence on the climate. The movement of the blades of a windmill or wind turbine generates energy. These blades move when pushed by the wind, so the presence of wind is necessary to generate electricity. In addition, the higher the wind speed, the greater the wind energy generation. In this session, we will build an anemometer that measures wind speed. Students will be able to know the wind speed using a specific sensor and an Arduino UNO board.

Learning outcomes

Purpose/Learning Objectives

- Teach students a relatively accurate way of measuring wind speed
- Interpretation of the result to determine the viability of wind energy generation projects
- Understand the importance of wind speed in our society

Transversal objectives

- Know other ways, apart from the empirical, to measure magnitudes
- Acquire basic knowledge of electronics and programming

Connection with Sustainable Development Goals

This topic is aligned with SDG 4, “Quality education”, specifically with Target 4.7 “By 2030, ensure that all learners acquire the knowledge and skills needed to promote sustainable development, including, among others, through education for sustainable development and sustainable lifestyles, human rights, gender equality, promotion of a culture of peace and non-violence, global citizenship and appreciation of cultural diversity and culture's

contribution to sustainable development". Building an anemometer is an educational activity that promotes practical learning and interest in science and meteorology.
In addition, it is related to SDG 7 "Ensure access to affordable, reliable, sustainable and modern energy for all". The anemometer, used to measure wind speed, plays a crucial role in the viability of wind energy generation, inspiring and motivating us towards sustainable development.

Implementation

Facilitation

- Ensure that all resources and materials are ready before starting the class.
- Electronic devices will be used. Due to their delicacy and possible electrical charge, pay special attention when handling them. Always disconnect them from the power grid or other equipment when handling them.
- When the wind speed is high, the anemometer will rotate at a high speed. Always keep the sensor at a safe distance to avoid hitting people or other objects.
- Make sure to use, whenever possible, recyclable or recycled materials.

Resources / Materials needed

- Computer with Arduino IDE installed
- Arduino UNO R3 development board
- USB type B cable
- Power adapter with 9-volt male barrel connector
- Anemometer for Arduino UNO R3 with voltage output from 0 to 5 volts

Ideas to continue

With the ability to know the wind speed at any point, an activity could be carried out in the vicinity of the session location, looking for those points where the installation of a windmill would be most feasible. It is important to note that this is a theoretical exercise, as not all wind turbine installation regulations will likely be met. To do this, it would be necessary to analyse the wind speed at different times of the day at various points. For example, studying three locations that will be visited three times a day. Teams could be formed with three anemometers to make the measurement more reliable, allowing the wind speed to be measured at three points simultaneously. The ideal location would be the one with the highest average speed.
They can also learn about the Beaufort Scale, an empirical measure of wind speed based primarily on sea state, waves, and wind strength.

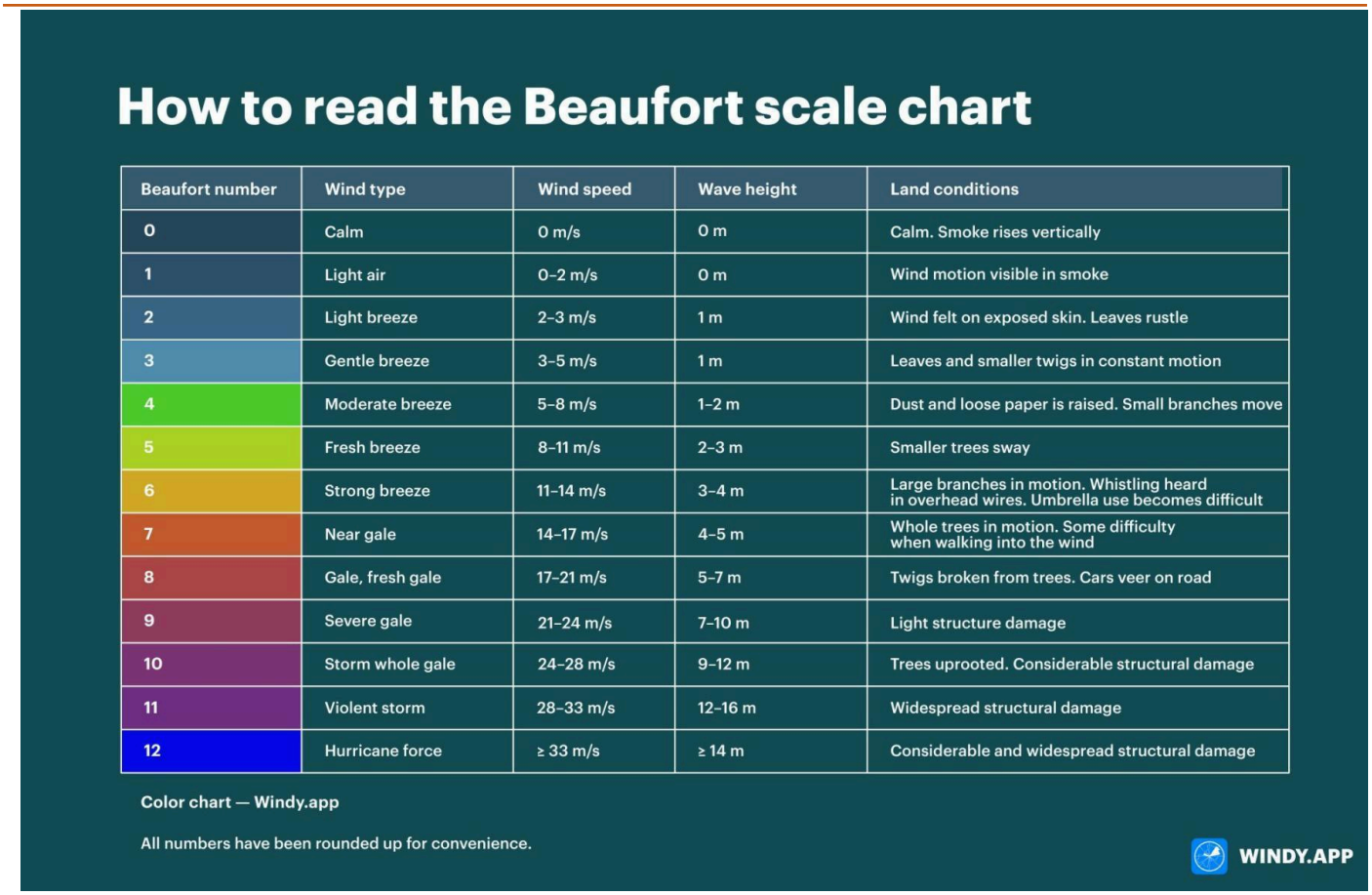


Figure 4.3.1 - Beaufort scale. Source: Windy. App. Link: https://windy.app/blog/wind-speed-beaufort-scale.html

This scale was developed around 1805, when wind speeds could not be measured as they are today. An exciting activity would be to test the scale with an anemometer and see its accuracy. It is important to note that the speeds in the table (and those indicated in weather forecasts) are measured at 10 meters above the ground. At 2 meters, the speed could be 50 to 70% of the above speeds. You can find more information about this scale at the following link: https://www.rmets.org/metmatters/beaufort-wind-scale.

Source / Materials and class preparation

Wind is a moving mass of air caused by pressure differences on the Earth's surface. Air always moves from points of high pressure to areas of low pressure. This pressure difference can be caused by differences in the heating of the Earth by the sun. You can learn more about wind in the following sources (https://www.rmets.org/metmatters/what-wind and https://education.nationalgeographic.org/resource/wind/), and show the following video to your students: https://www.youtube.com/watch?v=SqbTrbxWT1o. There are many ways to measure wind speed, and the instrument used for this task is called an anemometer. You can find more information about anemometers: https://www.mrclab.com/how-to-use-cup-anemometer). One of the most widespread anemometers is the cup anemometer, which has three or more cup-shaped elements attached to an axis. The cups act as an obstacle to the wind and rotate at a higher speed the higher the wind speed. You can check the wind speed and direction anywhere in the world on the following website: https://www.windy.com/.

Wind speed is not just a number on a weather report. It is a variable that can make certain activities successful, enjoyable, and safe or a failure, complicated and unsafe. Whether windsurfing, skydiving, or paragliding, the right wind conditions are essential for a rewarding, fun, and safe experience. Even in professions such as window cleaning, construction work, and high-altitude maintenance, a strong wind can turn a routine task into a severe hazard. Measuring wind speed and establishing safety thresholds is the key to preventing these accidents.
In turn, wind speed and other meteorological variables are key to making weather predictions and understanding phenomena such as tornadoes, hurricanes, storms, etc.
It is also essential to know it when determining the viability of wind farms, facilities that take advantage of the force of the wind to generate electricity. Wind farms' generation is determined based on the area's wind speed.

The primary and essential elements of this session are:

- Arduino UNO R3
- Anemometer with voltage output from 0 to 5 Volts
- 9V power adapter with barrel connector

Although this step-by-step guide is carried out with specific models of the three elements, the recommended anemometer is the following: https://www.dfrobot.com/product-1114.html; this session could be carried out using other models with other characteristics. However, although the guide will still be valid, modifying specific steps, especially those related to assembly and programming, may be necessary.
The Arduino UNO R3 development board, seen in Figure 4.3.2, is a programmable device ideal for beginning to acquire programming and electronics knowledge. It is a microcontroller-based board with the essential components of a computer: a processor or CPU (the brain of the computer), memory, and input/output peripherals (to communicate with the outside world). The Arduino UNO R3 board is a component that, although simple, has numerous functionalities that allow you to carry out any domestic project: automatically turning lights on and off based on ambient light, an automatic watering system for plants, and almost anything you can think of.

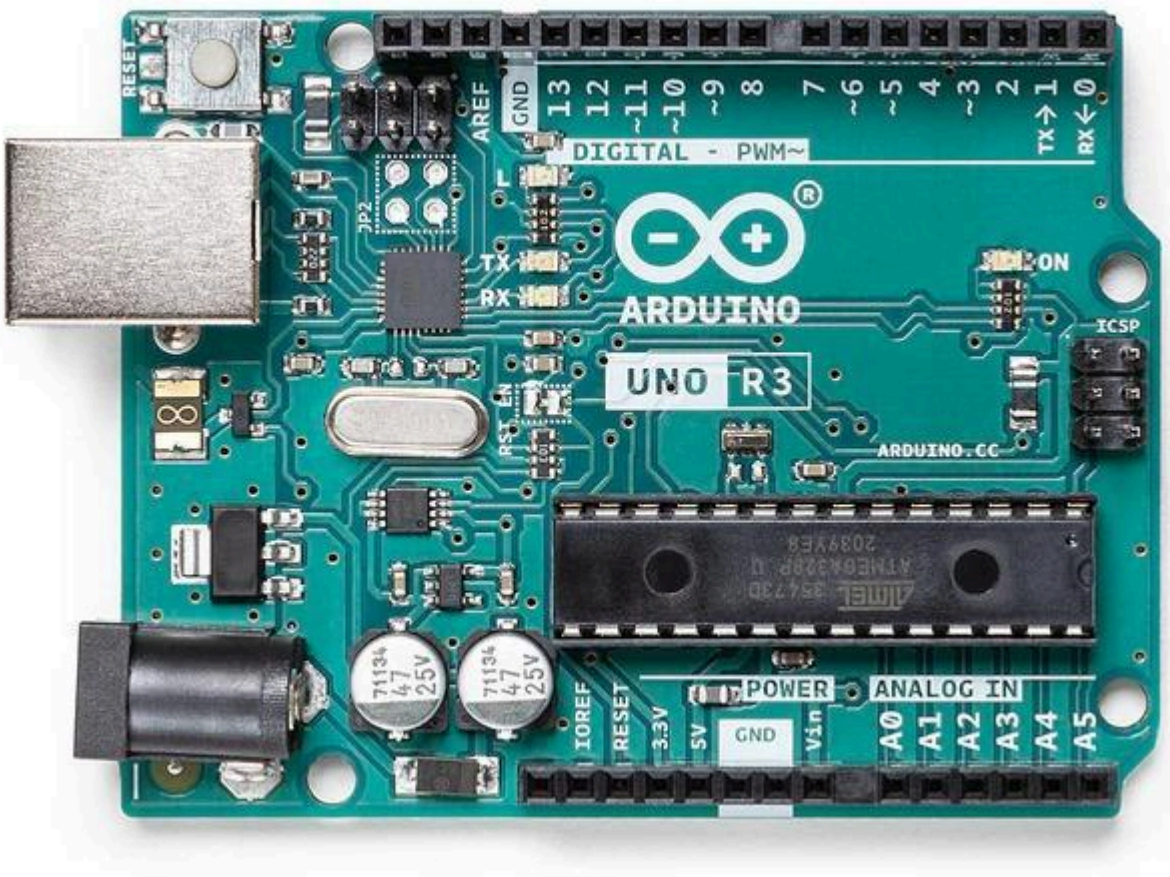


*Figure 4.3.2 – Arduino UNO R3 development board. Source: Arduino Link: https://store.arduino.cc/en-es/products/arduino-uno-rev3?_gl=1*1osninv*_gcl_au*NTg3MjU0NDI5LjE3Mjc3OTY4NTI.*FPAU*NTg3MjU0NDI5LjE3Mjc3OTY4NTI.*_ga*MTI4NDQzODQxMC4xNzI3Nzk2ODUx*_ga_NEXN8H46L5*MTcyNzc5Njg1MC4xLjEuMTcyNzc5Njg1OS4wLjAuMTE2NTA4MTcxMA..*_fplc*WEVJSlNnQmlWc0VMWjFEVDZ6TVolMkJIV3ZiWmJkT0Z5TlliWVVqbkZmbm9HJTJCelFLRiUyRlUwQzc4R0xiUVhlUktOQXIybTNRRWslMkZsbnFQZ3BwRCUyQm1kTFg3QVB1UyUyRmhhRGtDVW9kTmZhNW50JTJCQVFCZ2R6Y1pNM3YzM3J0RTdkZVElM0QlM0Q.*

One of the easiest ways to program this development board is through the Arduino IDE, which allows you to develop and upload code written in the Arduino programming language to the board. The Arduino programming language is based on C and C++. The structure of an Arduino code is divided into two functions: setup and loop. Figure 4.3.3 shows an Arduino code's basic structure, automatically generated when creating the file.

```
1  void setup() {
2
3  }
4
5  void loop() {
6
7  }
```

Figure 4.3.3 – Basic structure of an Arduino code

The first function, *setup* (https://docs.arduino.cc/language-reference/en/structure/sketch/setup/), is only executed once at the time of starting the program. It is used to initialise variables, configure the boards' inputs/outputs, start using specific libraries, etc. All the code (the

instructions we want to execute) contained in the function is executed only once at the beginning.
The other function, *loop* (https://docs.arduino.cc/language-reference/en/structure/sketch/loop/), is executed continuously while the development board is active. Its content is repeated cyclically, one after another. Normally, the code to carry out the desired task is placed in this space.
You can develop the code from the Arduino IDE, compile it (check that there are no syntactical errors or any other type of error), and upload it to the development board. The corresponding buttons in the upper left area of the environment allow you to perform these two actions. Figure 4.3.4 highlights these two buttons in the Arduino IDE.

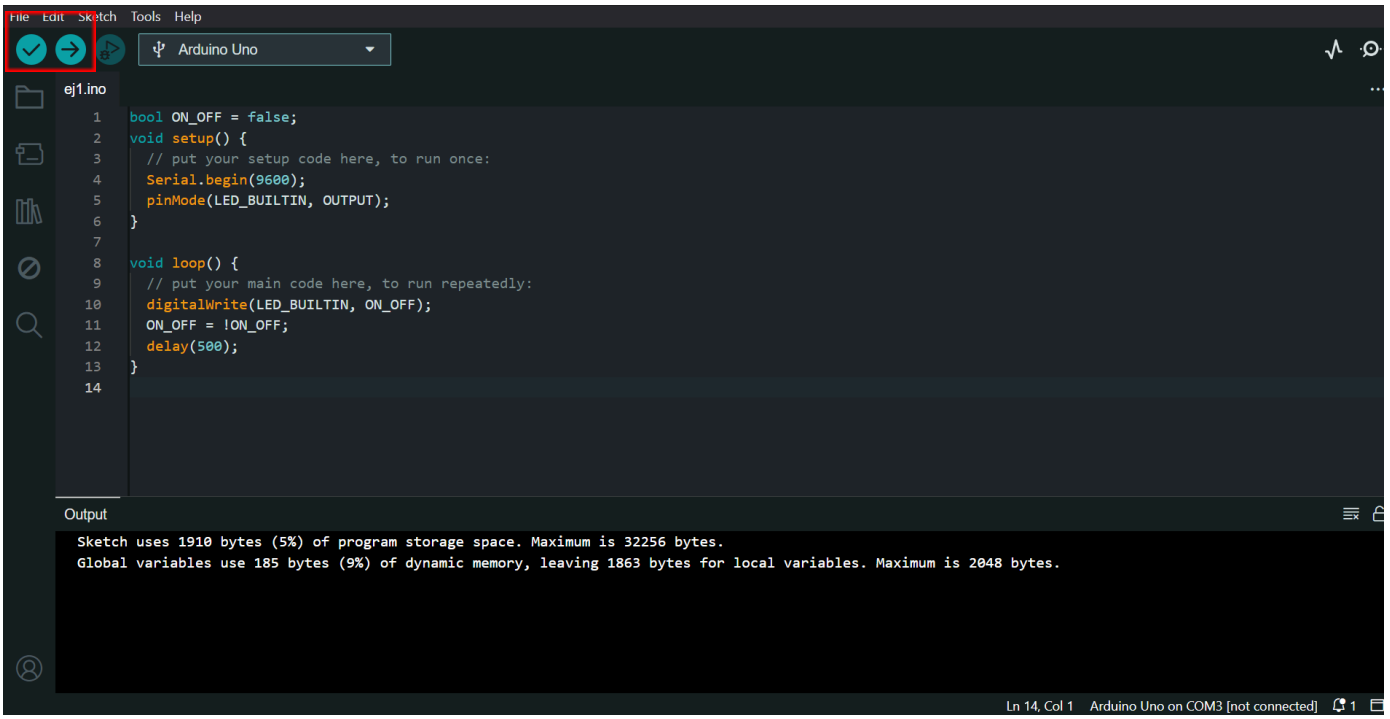


Figure 4.3.4 – Verify and upload code buttons.

In the following link (https://docs.arduino.cc/language-reference), you will find the official Arduino help for the language functions.
In this session, we will use integer and floating point variables, the serial port, and analog input reading. Integer variables (https://docs.arduino.cc/language-reference/en/variables/data-types/int/) can store whole numbers from -32,768 to 32,767. For example, an integer variable can store 8, 19, 26, or -1. However, it cannot store 0.5 because it is a decimal number. To save that number, we must use floating point variables (https://docs.arduino.cc/language-reference/en/variables/data-types/float/) that store numbers with a decimal point.
Another function that we will use is analogRead (https://docs.arduino.cc/language-reference/en/functions/analog-io/analogRead/) which reads an analog value, that is, a continuous value, and converts it to a digital value, that is, discrete; which is interpretable by the development board. In the case of the Arduino UNO, the analog input range is from 0 to 5V, converting it to an integer between 0 and 1023.
Finally, we will use the serial port functions (https://docs.arduino.cc/language-reference/en/functions/communication/Serial/) for

communication with the computer. It will be enough to configure said communication and specify the message to be sent.

The anemometer is the other essential element of the session. It is a sensor capable of measuring a physical magnitude (in this case, the wind speed) and returning that information in an electrical signal. Although there are various possibilities in which this electrical signal can be presented, we will focus on the form that we will find in this session: voltage. The anemometer can transform the wind speed into electrical tension, measured in voltage. The operation of the anemometer is simple; it has magnets mechanically anchored to the axis on which the cups are, and when the cups move due to the wind, the magnets also move, generating tension in the conductor due to Faraday's law of electromagnetic induction. This signal, previously conditioned, is sent to the output of the anemometer. The recommended anemometer for the session is the following: https://www.dfrobot.com/product-1114.html.

In the case at hand, the anemometer will generate a voltage between 0V and 5V at its output, depending on the wind speed. For example, the anemometer recommended for this session will return a value of 5V when the wind speed is 30 m/s, a value of 2.5V when the speed is 15 m/s, and 0V when the wind is moving at 0 m/s. Figure 4.3.5 shows the correlation line between sensor voltage (X-axis) and wind speed (Y-axis).

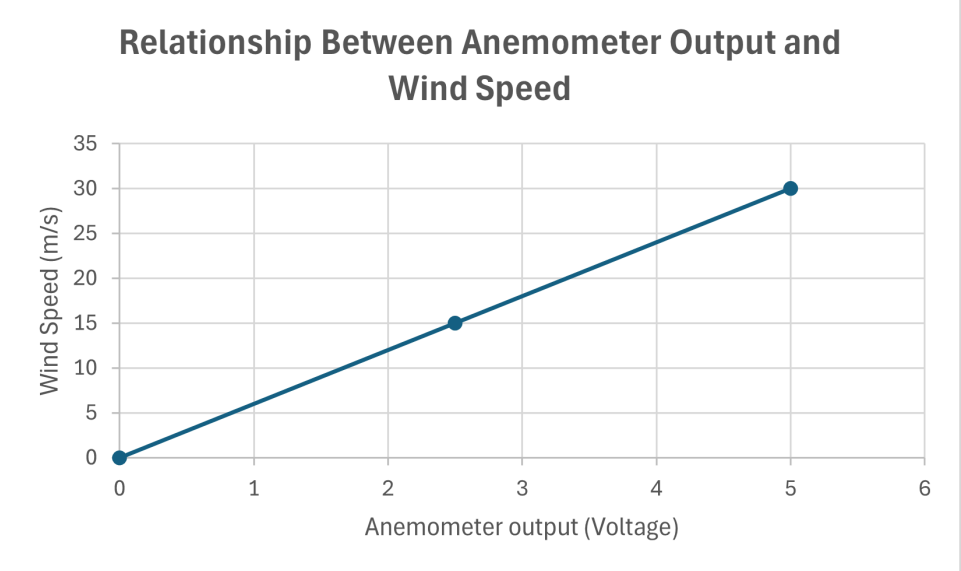
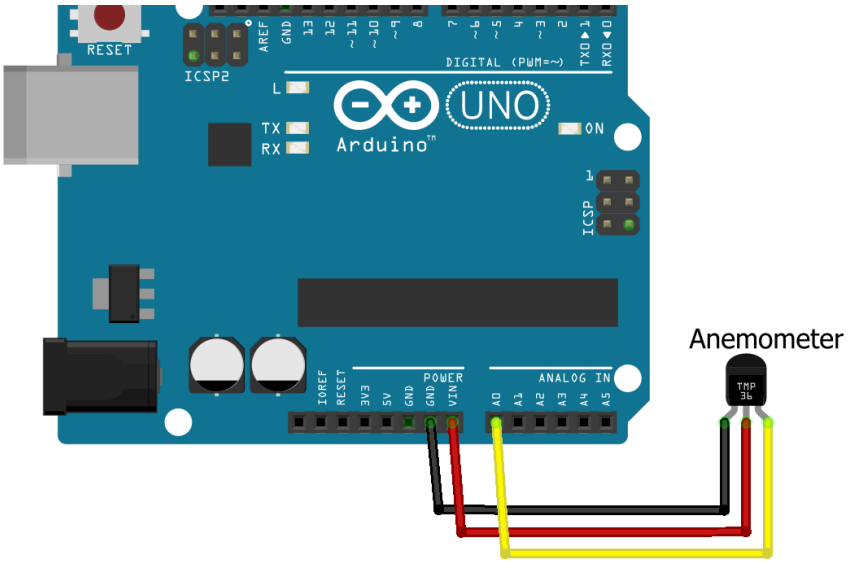


Figure 4.3.5 – Correlation between anemometer output voltage and actual wind speed

To sum up, the anemometer will generate a voltage at its output based on the wind speed. The Arduino UNO development board will be responsible for reading this voltage and performing the appropriate operations to convert the voltage to wind speed.

Outline of the topic

Time	Step-by-step instructions
5min	Unpack the anemometer carefully. Please take advantage of the opportunity to learn about its electrical characteristics and operation.
10min	Assemble the circuit. It will consist of a cup anemometer, which measures speed; an Arduino UNO R3, which calculates and displays wind speed; and a power adapter, which powers the devices. First, the black wire (*GND*) of the anemometer will be connected to any *GND* pin on the Arduino board. Then, the yellow wire of the anemometer (*Voltage Signal*) will be connected to pin *A0* on the Arduino board. Then, the red wire (*VCC*) of the anemometer must be connected to the pin *VIN* on the Arduino board. Figure 4.3.6 shows the electrical assembly.  *Figure 4.3.6 – Electrical assembly of the anemometer* Finally, the network adapter is connected to the Jack input on the Arduino board.
10min	We connect the Arduino board to the computer using the USB type B cable and open the Arduino IDE. We will start programming the Arduino. We declare and give a value, depending on the anemometer used, to the maximum possible speed and the voltage that indicates it (according to the anemometer). We initialise the serial port to establish communication with the computer. A speed of 9600 baud will be used. Figure 4.3.7 reflects the instructions for including this in the code. `float maximum_sensor_voltage = 5.0;` `float maximum_wind_speed = 30.0;` `void setup() {` `  Serial.begin(9600);` `}` *Figure 4.3.7 – Initializing variables and serial communication*

15min	Take the reading from pin A0. The output of the anemometer is connected to this pin, which will vary between 0 and 5 volts. The Arduino reading will return a value between 0 and 1023, with 0 being when the anemometer outputs 0V and 1023 when its output is 5 volts. This value is stored in an integer variable called the *sensor value*. Before passing it to speed, it is necessary to know the voltage to which it is equivalent, so the conversion is performed, and it is stored in a floating variable called *sensor voltage*. Finally, the conversion is performed between the voltage of the anemometer (stored in *sensor voltage*) and the wind speed (*wind speed*). Figure 4.3.8 shows the code to perform this entire process. `void loop() {` `  int sensorValue = analogRead(A0);` `  float sensorVoltage = sensorValue * (5.0/1023.0);` `  float windSpeed = sensorVoltage * maximum_wind_speed/maximum_sensor_voltage;` *Figure 4.3.8 – Anemometer reading and conversion to wind speed* For this, the variables *maximum_wind_speed* and *maximum_sensor_voltage* are used, which allow the conversion ranges to be easily modified to adapt the code to different sensors.
5min	Send the speed value to the computer to check its value through the computer's serial port (available from the Arduino IDE). First, a fixed text is sent: "The wind speed is: ". Then, the calculated speed is sent, and the speed units are indicated. Modify this last field to show the measurement in another unit. Figure 4.3.9 shows the message being sent via the serial port. `Serial.print("The wind speed is: ");` `Serial.print(windSpeed);` `Serial.println(" m/s.");` *Figure 4.3.9 – Sending wind speed to the computer*
5min	Finally, a delay should be set so that the wind speed calculation is periodic and not done continuously. A delay of 500 milliseconds (0.5 seconds) is a good value. Figure 4.3.10 shows the wait implemented in the code. `delay(500);` `}` *Figure 4.3.10 – 500ms delay between wind speed readings*

Formative assessment

Formative assessment will be carried out through classroom observations. Teachers observe students carrying out the activity, paying particular attention to their application of the acquired knowledge. Oral questions may also be asked of each work group to determine their theoretical knowledge of the subject, each student's ability to work in a team, and their degree of participation in the activity.

Feedback

Net Promoter Score (NPS) is a widely used tool to measure customer or student loyalty and satisfaction in an educational context. It measures the likelihood that students will recommend the subject or course to others.
Here's how to analyse and apply NPS:

NPS Question Development:

- Main question: "On a scale of 0 to 10, how likely are you to recommend this topic to other students?".
- Follow-up questions: "Why did you select that score?" and "How can we improve this topic?".

Collecting responses:

- Digital Surveys: Use online tools like Google Forms, SurveyMonkey, or Typeform to collect student responses.
- Aper surveys: At the end of the lesson, paper questionnaires are distributed for students to fill out. The Appendix shows a template for collecting responses using paper surveys.
- Interactive Applications: Interactive applications such as Quizziz, Mentimeter, or Kahoot! Collect responses in real time.

Classification of responses:

- Promoters (Score 9-10): Students who are very satisfied and likely to recommend the topic.
- Passives (score 7-8): Students are satisfied but not enthusiastic; they could easily change their minds.
- Detractors (Score 0-6): Students who are dissatisfied and unlikely to recommend the topic may offer constructive criticism.

NPS Calculation:

- Formula: NPS = % of promoters - % of detractors.
- Interpreting the score: NPS ranges from -100 to +100. A positive score (above 0) is generally good, and a score above 50 is excellent.

Analyse and use feedback:

- Review the comments to understand the reasons behind the scores.
- Identify patterns and specific areas of improvement among detractors and passives.
- Use positive feedback from promoters to reinforce the positive aspects of the topic.

Annexes

Programming code for the Arduino UNO R3 development board.

```
float maximum_sensor_voltage = 5.0;
float maximum_wind_speed = 30.0;


void setup() {
  Serial.begin(9600);
}


void loop() {
  int sensorValue = analogRead(A0);
  float sensorVoltage = sensorValue * (5.0/1023.0);
  float windSpeed = sensorVoltage * maximum_wind_speed/maximum_sensor_voltage;


  Serial.print("The wind speed is: ");
  Serial.print(windSpeed);
  Serial.println(" m/s.");


  delay(500);
}
```

QUESTIONNAIRE TO COLLECT FEEDBACK ON THE TOPIC

On a scale of 0 to 10, how likely would you recommend this topic to another student?

0	1	2	3	4	5	6	7	8	9	10

Not likely at all — Extremely likely

Why did you choose that score?

How could this topic be improved?