Co-funded by the
Creative Europe Programme
of the European Union
Project 2020-1-TR01- KA201-094533

STEM
E-CONTENT
3D VIRTUAL REALTY

The Key To Global Life,
Digital Change Of Nature

Total Duration: 2 or 3 hours

Student's Age: 14-18 Years

Application Area:
- Climate change
- physics
- electronics

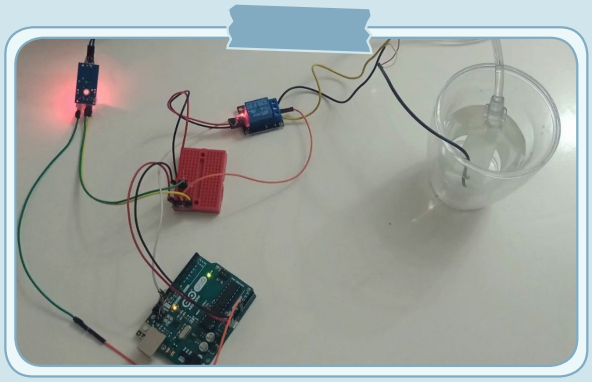
Keywords: Temperature, data analysis, climate, Arduino, technology

G1 - Temperature Monitoring System with Arduino UNO

Module
- Global Warming

E2 - English Version

Materials:
- **Paper (A3)**
- **(Colored) pencils**
- **Laptop**
- **Arduino (www.arduino.cc)**
- **Breadboard**
- **Temperature sensor (LM35 or alternative)**
- **Jumper wires**

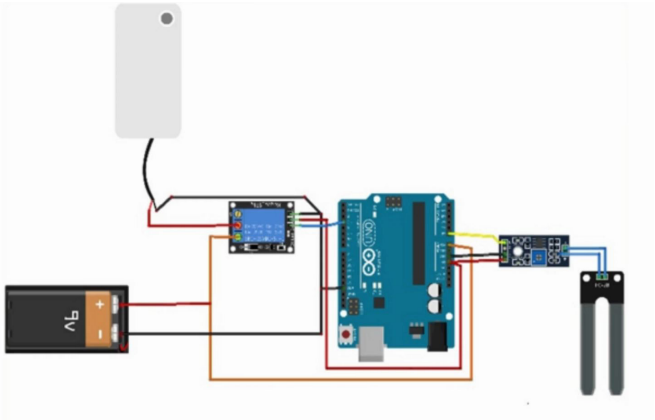


Notes:
- Size of each group: 3-4 students.
- Fase 1: Make it in a virtual environment first.
  - o Use Tinkercad (circuits):
  - o Sketch a neat, detailed sketch of the setup: include temperature sensor, Arduino and breadboard.
  - o Connect everything together, using different colors of 'wire' and use the correct color of wire for positive and negative pole.
  - o Make the program on the circuit in Tinkercad and test it first online.
- Fase 2: Make it real:
  - o Find all the components and connect everything.
  - o Do not connect the Arduino to the PC yet! The teacher checks it first.

Introduction

We will build an affordable, portable DIY installation monitoring temperature and light intensity. We use an Arduino UNO to collect data, and link it to a computer to display the data in Excel. To perform this activity the teacher needs to have basic knowledge in the subject of Arduino, Arduino IDE and basic use of Excel.

Weather and climate are of large importance in today's society. Increasing computer power enables us to run weather models at ever higher resolutions, but this creates a need for more local weather data. Imagine a country where every school monitors weather, we can get very accurate ground-level data (Picture 1).



Picture 1. Tempurtare Monitoring System

Not only are measurements important, you need to visualise what you measure.

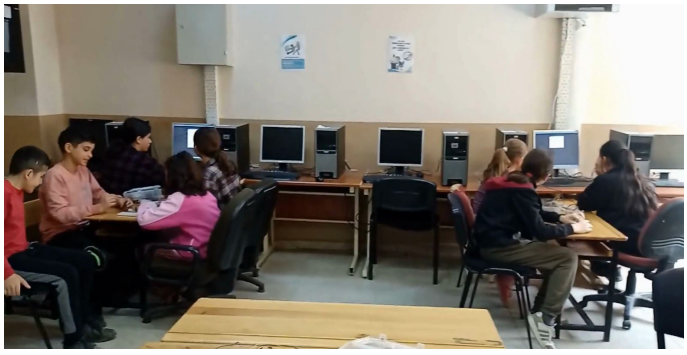
We will build an affordable, portable DIY installation monitoring temperature and light intensity. We use an Arduino to collect data, and link it to a computer to display the data in Excel.

Considerations

- Size of each group: 3-4 students.
- Fase 1: Make it in a virtual environment first
  Use Tinkercad (circuits):
  Sketch a neat, detailed sketch of the setup: include temperature sensor, Arduino and breadboard.
  Connect everything together, using different colors of 'wire' and use the correct color of wire for positive and negative pole.
  Make the program on the circuit in Tinkercad and test it first online.
- Fase 2: Make it real:
  Find all the components and connect everything.
  Do not connect the Arduino to the PC yet! The teacher checks it first.

Aim of the Activity

- Make it easier to measure temperature at regular intervals
- Use an analysis tool like Excel that is widely known to collect and visualize data
- Learn to work with sensors in general to collect data:
- Understand the relationship between measured voltage and the physical quantity to be measured
- Learn the concept of calibrating a sensor, learn to work with a technical data sheet
- Learn to program in C++
- Get acquainted with linear functions



Picture 2. Aim of the activity

Activity Process

Before Activity

- Explain the assignment: background, aim, time frame for each part
- Divide the class in groups 2-4 students, each group at their own table. Each group has a laptop, paper and pencils.

Picture 3. Mix with a spoon

Let's Start

1 Circuit Design (30 Minutes)

Draw the circuit on a page.
You can use Tinkercad Circuits to draw it online instead. What you need is (Picture 3, and 4).

- an Arduino
- a breadboard
- the temperature sensor (LM35 or TMP36)

Picture 4. Materials

2 Connect it!

- Position the temperature sensor on the breadboard, be sure it is put the right way that the pins are not short circuited.
- The sensor needs Voltage to operate. It can use the 5V pin of the Arduino. Check the figure for the correct pin lay-out. Connect pin 1 to the 5V pin of the Arduino. Connect the GND to the GND of the Arduino.
- The Arduino has to 'listen' to the readings of the sensor. This is analog input, you have to use one of the analog input pins of the Arduino. Connect the middle pin of the sensor to the A0.

3 Programming: display the output of the sensor

We start reading what your Arduino receives from the sensor (the INPUT) and display it in the Serial Monitor. You do this by programming in C++. We'll start with a simple program that reads the sensor reading every second, and prints it to the computer's console (Picture 5).

- To read the voltage on pin A0, use analogRead(A0).
- You store the voltage in a variable called 'value'. In this programming language (C++) you always have to say what kind of information you want to store in a variable when you create the variable. Creating the variable is called "declaring" and that happens in line 1 of the program. In this case, the variable will contain a whole number, called "integer". Hence the int value at the very top of the code.

```
1  int value;
2
3  void setup() {
4    Serial.begin(9600);
5  }
6
7  void loop() {
8    value = analogRead(A0);
9    Serial.println(value);
10   delay(1000);
11 }
```

Picture 5. C++ codes

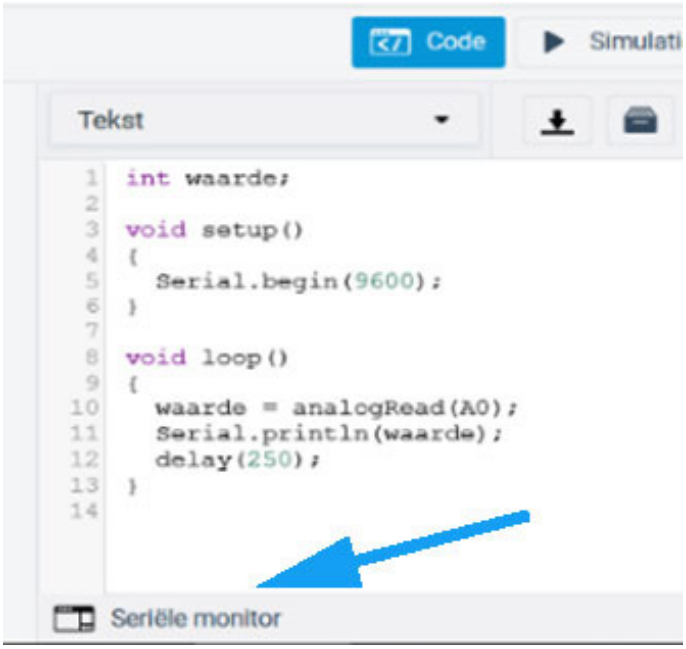
Table 1. Other types of information in C++

int	Whole number (0, 1, 2, -3, 19839, -78, …)
float	Decimal number (0.13, 713.24, -3.0, ….)
char	One single character ('a', 'B', 'c', 'D', '0' '*', 'µ', …)
String	text ("Hello, world!", "this is a test", "123", …)

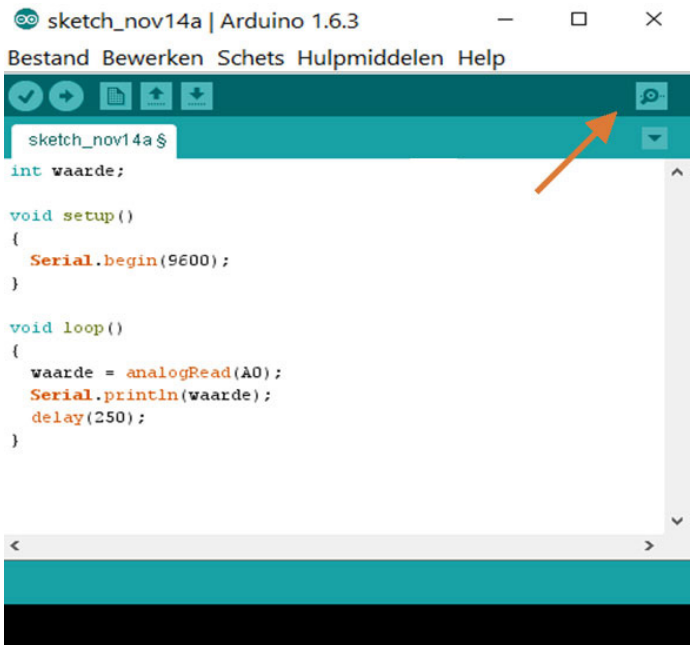
4 Communication between the Arduino and PC

- As you may remember, you "upload" a program to the Arduino, and the program runs on the Arduino, not the computer. It is the Arduino that reads the sensor value, but the computer that has to display this value on the screen. The Arduino must send this information first to the computer.
- To set up a connection between the Arduino and computer, use ***Serial.begin(9600)*** in the ***setup block*** (because you only need to set up the connection once, at the very beginning). The number between parentheses () tells the computer how many bits it will send per second. This is called the 'baud rate'. The higher this number, the more data is sent per second, but also the higher the chance of errors during communication (that a 0 accidentally becomes a 1 and vice versa). For some applications it is important to achieve a high speed, for example in 3D printing. 9600 is more than sufficient for the applications we use.
- After the Arduino has established communication with the computer and has read the value of the middle pin and stored it in a variable, it is time to send this value to the computer.
- ***Serial.println(value)*** causes a line to be printed on the computer (hence ***println***, translated as 'print line'). Put in parentheses what should be printed.

- The output can be found in the 'Serial monitor'. In TinkerCad you can find this by clicking on 'Serial monitor' at the bottom of the Code screen (Picture 6).
- In Arduino IDE you can bring it up by clicking on 'Serial monitor' in the 'Tools' menu. Or click on the icon (Picture 7).



Picture 6. Serial monitor in Tinkercad



Picture 7. Serial monitor in Arduino IDE

5 Programming: calibrating the sensor

- You receive a certain value from your sensor, but what does that value mean exactly? That is why it is important that you calibrate your sensor, so that you can correctly interpret a certain value. In this case: what does the value 523 mean? Is it "warm" or "cold"? This time, we have to look in the datasheet of the LM35/TMP36 sensor to find the specifications. If we find them, we know how to convert the reading to the correct temperature.

- Add the keyword 'datasheet'

- Search on the internet what are the specifications for the temperature sensor. Note down in the table below what you have found.

Condition	*Theoretical measured value*
Min temperature	
Max temperature	
Slope (mV/degree Celsius)	
Accuracy	
Vout at 150°C	
Vout at 0°C	
Vout at -20°C	

Now we have to find a formula to calculate our temperature. But how does analogRead() work?
Arduino always translates the measured voltage on the analog pin (0-5V) to a number (0-1023).

- 0V translates to 0
- 5V translates to 1023

So you have divide by 1023 and multiply by 5 again to go from value to voltage.

The next part depends on the sensor you use, so be careful to use the correct specifications! Our example below is valid only for LM35:

We know from the specifications in the datasheet that

- 0°C translates to 0V
- 150°C translates to 150mV = 0,150V
- 5000°C translates to 5000mV = 5V

So each 10mV (0.01V) is 1°C.
You have to multiply the measured voltage by 100 to get the °C

The formula to calculate the temperature from the value as a result from analogRead()
*Temp = (5.0 * value *100.0) / 1023.0*

Our program to measure and calculate the temperature with an LM35 sensor (Picture 8)

Be careful, decimal number are displayed with a point (.) not a comma (,)

```
int value = 0;
float temperature = 0;

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

void loop() {
  value=analogRead(A0);
  Serial.print("our sensor reads: ");
  Serial.println(value);
  temperature = 5.0 * value * 100.0 / 1023.0;
  Serial.print("this means in degrees Celsius: ");
  Serial.println(temperature);
  delay(1000);
}
```

Picture 8. DCalculate the temparture

For the smart students that finished faster:

Work it out for another sensor type like TMP36, TMP37 or the very commonly used DHT11 or DHT22.

6 Capture the data in Excel

We can already print the values via the Serial monitor. Now we see how to stream this data to Excel instead. This might come in handy.

There is no change to the input or your circuit. All we need to do is enable and configure an add-in in Excel. This captures the data that is sent to the Serial monitor and displays these values in a number of cells configured for this purpose.

The add-in we need is called 'Microsoft Data Streamer for Excel' and is present by default in the latest versions of Microsoft Excel. You still have to enable it.

Click on:

- ***'File' >> 'Options' >> "Add-ins"***
- ***click on the arrow next to "Manage: Excel Add-ins" >> 'COM Add-ins' >> 'Start'***
- ***Check the box next to 'Microsoft Data Streamer for Excel'***
- ***OK'***

Now, a new tab will appear in the ribbon, on the far right, called "Data Streamer." Now open the "Data Streamer" tab. Click on 'Connect device', and choose 'Arduino Uno (COMX)'.

A new workbook is created. To stream data, click on 'Start data'. Each second time and temperature are now printed here. The time is added automatically by Excel. In the 'Settings' worksheet you can adjust, among other things, how many rows of data should be visible (max. 500). Now set the number of data rows to 50.

Modify the program in that way that you only print de values, not the text. *Now print both values on the same line, separated by a comma.*

- ***Serial.print(value);***
- ***Serial.print(",");***
- ***Serial.println(temperature);***

This prints one line consisting of the temperature, a comma, and the humidity. if you don't use the comma in between, Excel would treat this as one big value instead of two separate ones. Note that the data sent to the Serial monitor is formatted in the same way.

7 Graphs

Make two line graphs, one for the measured value and one for the temperature. If everything went well, these graphs should be updated live.

8 Save

You can also save data for later use. To do this, click on the button "record data". After you stop recording, you will be asked where you want to save the file. The format of this file is '.csv', short for 'comma separated values'. You can read this format in a lot of data related programs afterwards, including Excel.

9 Now tinker it, expand it

- Use a second sensor, e.g. an LDR, soil moisture sensor, … Add the sensor values that you read to the Excel. For the soil moisture sensor, calculate the relative humidity.
- Put everything in a graphic visualisation

Assesment

Evaluation

- The design of students can be displayed within the school. Different products can be created by diversifying waste materials used.

Goals	Must be Improved (1)	Medium (2)	Good (3)	Very Good (4)
(Re)formulating a problem so that it can be solved by a computer or other tool.	(.....)	(.....)	(.....)	(.....)
Active participation	(.....)	(.....)	(.....)	(.....)
Creating a set of instructions to reach a goal from a starting point (= algorithm) succeeds independently.	(.....)	(.....)	(.....)	(.....)
Selecting and implementing useful information from specified source succeeds independently.	(.....)	(.....)	(.....)	(.....)
Critical thinking	(.....)	(.....)	(.....)	(.....)
Total				

Links

- Colorfulworld. (2018). How to easily make a penguin, ice and igloo for a small amount of money / DIY. https://www.youtube.com/watch?v=Es-rCeIq6YU
- Cristofari, R., Liu, X., Bonadonna, F., Cherel, Y., Pistorius, P., Le Maho, Y., . . . Trucchi, E. (2018). Climate-driven range shifts of the king penguin in a fragmented ecosystem. Nature Climate Change, 8(3), 245-251.
- ScienceBuddies. (2020). Ocean Currents: Modeling the 'Global Conveyor Belt' in Your Kitchen. Retrieved 1010.2022 from https://www.sciencebuddies.org/science-fair-projects/project-ideas/OceanSci_p012/ocean-sciences/ocean-currents-modeling-global-conveyor-belt
-
- https://www.sciencebuddies.org/stem-activities?s=global%20warming
- https://www.sciencebuddies.org/stem-activities/polar-ice-caps-melting
- https://www.sciencebuddies.org/science-fair-projects/project-ideas/OceanSci_p015/ocean-sciences/will-ice-melting-at-poles-cause-sea-levels-to-rise
- https://www.tinkercad.com/things/c3BkCJdQxeI
- https://www.tinkercad.com/things/9UeZJTri0zD
- https://www.youtube.com/watch?v=Gkw45JaEQio
- https://www.youtube.com/watch?v=ztQYbRwBboU
- https://science.howstuffworks.com/environmental/earth/oceanography/ocean-current.htm
- https://web.ics.purdue.edu/~braile/edumod/convect/convect.htm