DOI: 10.2478/sbeef-2022-0013

IMPLEMENTATION OF AN AUTOMATED GREENHOUSE

C. STANESCU
Department of Electronics, Telecommunications and Energy Engineering, Valahia University of Targoviste, Romania
E-mail: cristian.stanescu.ie@valahia.ro

Abstract. ***The development of measuring devices such as sensors and transducers appeared as a necessity imposed by technical progress and the need to control certain processes in real time. The paper proposes the analysis and implementation of an automated greenhouse where the "heart" of the circuit is the Atmega328P microcontroller integrated on an Arduino Nano development board.***

Keywords: Atmega328P, Arduino Nano, DHT11, BMP180

1. INTRODUCTION

Initially, the measuring devices were created based on human senses, namely sight led to the development of optical sensors, smell led to the development of gas and humidity sensors, hearing to the appearance of acoustic or pressure sensors, tactile sense led to the creation of thermal sensors, pressure or capacitive, and taste resulted in the appearance of chemical composition sensors. Obviously, this was only the basis, because nowadays there are many more sensors that exceed the human capabilities to feel certain stimuli. In general, regardless of the field of use, the sensors are conditioned like this:

- To carry out the primary processing of information.
- To ensure high operational safety.
- To supply a high-precision signal, without disturbing the measured source.
- To allow the choice of the measurement range and the sensitivity adjustment.
- To admit the overload of the duration.
- To show adaptability to location.
- To ensure immunity to disturbances.
- To have a simple, rigid construction, resistance to shocks and the action of the environment in which it is placed, small mass and dimensions.
- Have simple In-Out links.
- Requires easy adjustment and maintenance [1-4].

Modern measuring systems have a wide range of applications. They can be found in the fields: industrial, scientific or laboratory, but they can also be used in home environments. However, the measurement of the signal requires conditioning before the data acquisition system can effectively and accurately take over the respective values. This part of the conditioning can be thought of as amplification, attenuation, filtering, electrical isolation, sampling, or multiplexing. Moreover, most transducers require stimuli, namely currents or voltages, the creation of measurement bridges, linearization, or amplification to ensure their best operation [5].

Microcontroller is an assembly of electronic structures intended for controlling a process or interactions with certain components or sensors. It works based on a code loaded in the flash memory. The first controllers were made purely analogue, using discrete components or electromechanical components. Obviously, the sizes were on point, and the consumption was quite significant. Comparatively, today most electronic components of the microcontroller type come in the form of an IC (Integrated Circuit), which refers to all the components of very small sizes, encapsulated in a garment, usually of plastic or of a solid material to ensures the protection of circuits and components and their possible cooling [6-7].

2. DEVICES USED IN IMPLEMENTATION OF THE PROJECT

There are two major parts that make up the platform: the development board, which we use when we physically build the project, and the software (Arduino IDE) which is the code that makes the circuit work. Thus, we write the code in the form of a "sketch", upload it to the platform, and it knows how to work properly [8].

Not so long ago, working on the hardware side of electronics meant building projects from scratch, using "n" components with strange names (transistor, capacitor, coil, etc.). Each circuit was specialized on a single method, and changing it involved cutting wires, adding new components, soldering on PCBs, etc. With the digital evolution of technology, all of this can be replaced very easily with a few lines of code through software. We can change the interaction of certain components just by changing a few words instead of removing or putting components then pasting etc.

The list of used components can be seen in the table below (Table 1).

Table 1. Components used in the implementation of the project

Part	Value	Type
Arduino NANO v3	ATmega328p	-
4x Potentiometer	10kΩ, 200V	Linear, 15mm
Capacitor	100uF, 50V	Electrolyte
LCD 2004A	4 lines x 20 characters	Green
Module I2C LCD	-	I2C
Photoresistor	-	Analogue
DHT11	-	Digital
BMP180	-	I2C

DS3231	-	I2C
Sensor USol	-	Analogue
4x Resistances	10kΩ	Carbon film 1/4W
8x Resistances	220Ω	Carbon film 1/4W
3x Transistor	2N2907	PNP
8x LED UV	40mW, 20mA,3.2V	UV 395-400nm
Fan	0.3A, 5V	8cm x 8cm
The submersible pump	3-6V	80-120L/H
Power supply	1.55A, 5V	Samsung charger

The ATmega328P is a chip made by Atmel, part of the megaAVR family. It has a modified 8-bit RISC Harvard architecture [9].

The general features of the Arduino Nano platform are (Figure 1): ATmega168 (v2.x and older) or ATmega328 microcontroller; operating voltage of 5V; the supply voltage is between 7-12V; 14 digital inputs/outputs of which 6 have the possibility to work PWM and 8 analogue inputs; the maximum current on the input/output pin: 40 mA; 16MHz clock frequency or 8MHz with internal clock [10].

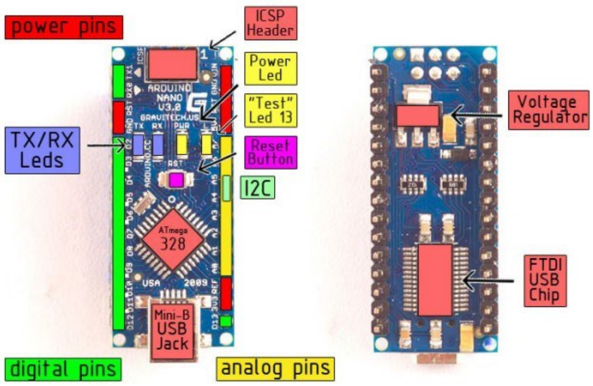

Figure 1. Arduino NANO

DHT11 (Figure 2) is the sensor that provides us with the values of temperature and humidity in the air. It has 3 pins, namely 5V, GND and S. The last one is the data path to the microcontroller. The data is of a digital type. The values it can measure are humidity between 20%-90%, with an accuracy of ±5% and temperature that has values between 0 and 50°C with an accuracy of ±2°C. The electrical characteristics are: 3-5V power supply, the measured consumption is between 0.5mA and 2.5mA, the standby current is between 100µA and 150µA and the readings are done at one second [11-12].

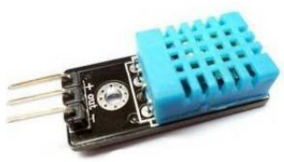

Figure 2. DHT11 sensor

The BMP180 (Figure 3) is the module that can give us both the temperature (but lower quality than the DHT11) and atmospheric pressure. This has 4 pins (3.3V, GND, SDA, and SCL). It is connected to pins A4 and A5 on the I2C bus of the microcontroller and a specific address is chosen to avoid confusion of values. Measurement accuracy is decent. It can read atmospheric pressure values between 300 and 1100hPa with an accuracy of ±0.12hPa. Operates at a low voltage of 3.3V with a low consumption of 3µA, with atmospheric pressure readings occurring every 5ms [13].

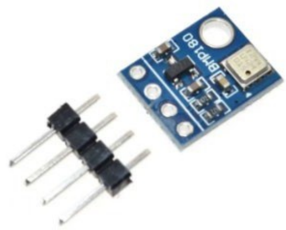

Figure 3. BMP180 sensor

The DS3231 (RTC = Real Time Clock) (Figure 4) is a high-accuracy timekeeping module. The connection interface is through two pins, SDA and SCL, together with 5V and GND. It is connected on the I2C bus. It has a precision of ±2ppm from 0°C to +40°C, and of ±3.5ppm from -40°C to +85°C. It has a 3.3V backup battery in case the power supply is interrupted, thus, the time-related data is saved [14].

Figure 4. DS3231 module

The soil moisture sensor (Figure 5) consists of two blades that send a weak current to each other. Depending on the soil moisture, it returns a value dependent on the resistance imposed by the material it passes through. Wet soil has a low resistance; thus, we get a high value and for a dry soil, the resistance is high, and we will receive low values. It has 3 pins, it can be powered from either 3.3V or 5V, GND and S from the values it sends to the microcontroller. If it is connected digitally, it returns either 0 for dry or 1 for wet, if it is connected analogue, it will give values from 0 to 1023 proportional to the soil moisture [15].

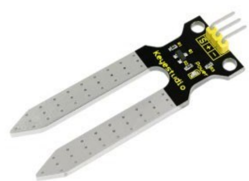

Figure 5. Soil moisture sensor

LCD2004 (Figure 6) comes equipped with an I2C module for easy connection. It is based on a chip that converts bits into the specific information of the bus. It connects to a 5V power supply and has a consumption of approximately 250 mA. The display is done on 4 lines each having 20 characters [16].

Figure 6. LCD2004 and I2c module

The pump (Figure 7) is used to be able to water the plants in the greenhouse. It can work between 3-6V, consuming between 100 and 200 mA, having a flow of 80-120L/hour [16-17].

Figure 7. Submersible pump

3. THE SIMULATION OF THE PROPOSED SYSTEM

As modelling and simulation program I used Proteus 8. It is part of Design Suite Proteus, a set of functions and programs used mostly for electronic circuit design.

The Proteus 8 software is generally made for electronic components, not for simulating projects for microcontrollers. However, it supplies a varied range of functions to simulate various circuits [18].

The components used were: pot-hg (4x 50k potentiometers), led-blue (symbol for UV LEDs, has a 220Ω series resistance), led-green (water pump symbol, has a 220Ω series resistance), led-red (symbol for fan, has 220Ω series resistance), ldr (photoresistor), jhd-2x16-i2c (2-line 16-character lcd display), hih-5030 (soil moisture sensor), ds3231 (clock module), dht11 (humidity and air temperature sensor), bmp180 (atmospheric pressure sensor), 2n2907 (3x transistors for controlling symbols), resistor (10kΩ for ldr) and Arduino Nano v2 (microcontroller) (Figure 8).

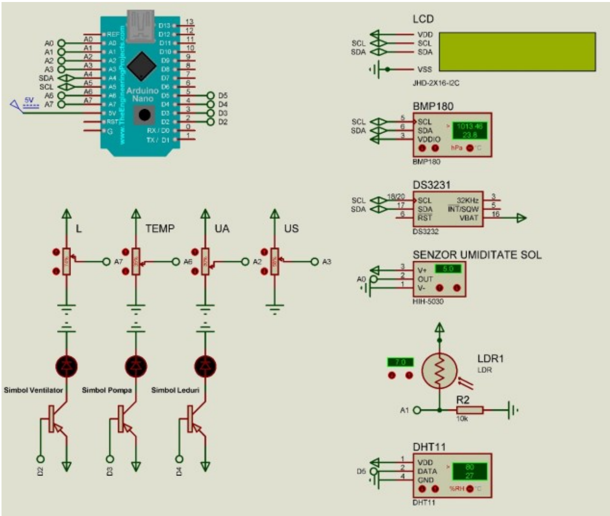

Figure 8. Circuit schematic

As can be seen in Figure 8, we can set the values of each sensor and each potentiometer to later simulate the correct operation (except for the clock module which must be programmed from the code). To "translate" the values received from the sensors, I changed from their properties. For example: the soil moisture sensor we changed the part values (Figure 9) from 0-100 to 0-10.

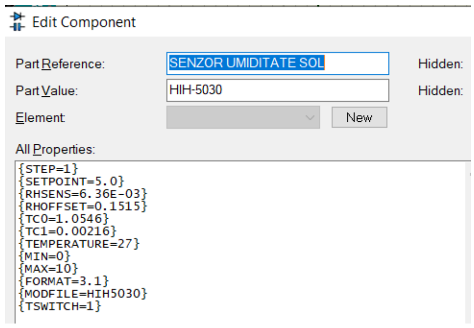

Figure 9. Change of soil moisture sensor values

The next step was creating the code to get the circuit working.

In Figure 10 we can see the "Export compiled Binary" option that allows us to transform the code from the IDE into hex code and hex code with bootloader to load it into Proteus for simulation. In Figure 11, the way to upload the hex file is shown, "Upload hex file" is the file previously exported from the IDE, which is usually found in the same place as the sketch.

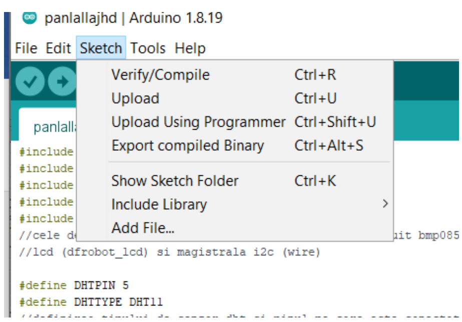

Figure 10. Exporting the code in hex form

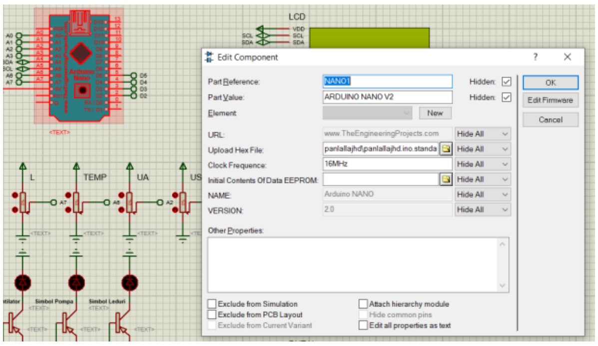

Figure 11. Adding the code for the simulation

Figure 12 shows the first page of the LCD display, which has the time, date, and the value of the atmospheric pressure in millibars.

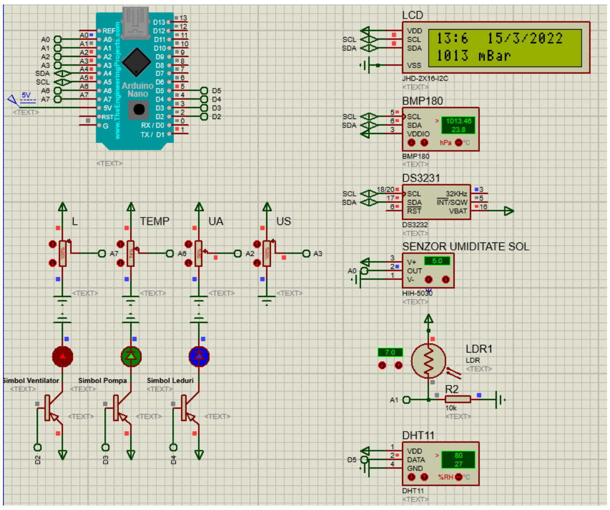

Figure 12. The simulation in the first case

In Figure 13, the LCD displays a soil moisture of 5/10 (5 is the same value returned by the sensor) and it is also noted that the pump symbol is active. It follows that it should start and water the plants until the soil moisture reaches the value of 10, and the displayed temperature is 27^0C/42^0C (the value of 27^0C is the same as that returned by dht11), the fan should be stopped, but in Figure 14 it is found that the humidity is 2% more than the desired level, so the fan is turned on. Also in Figure 14, the value of the light is 8/10, so the symbol for the LEDs must be lit.

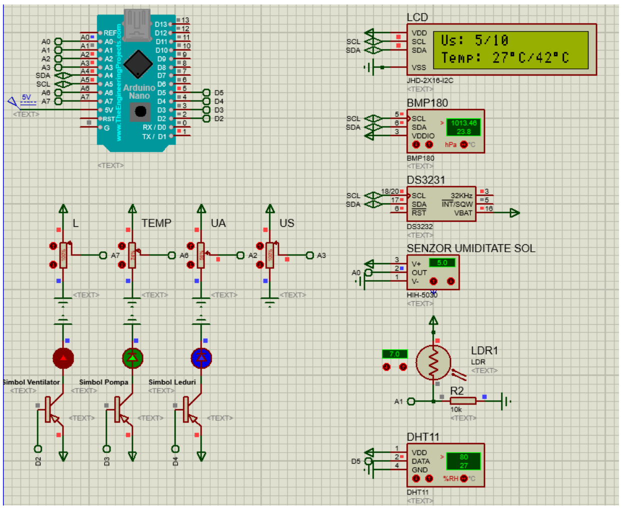

Figure 13. The simulation in the second case

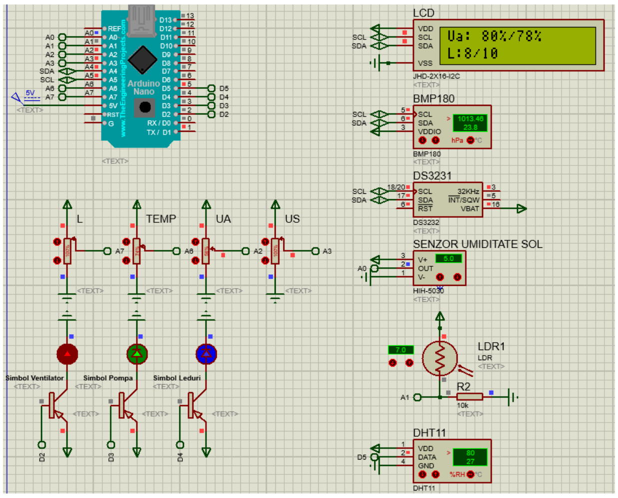

Figure 14. The simulation in the third case

In Figure 15, we want the pump to stop and the LEDs and the fan to remain on, so we change the US potentiometer to 20%, and the soil moisture sensor still is at the value 5 (Us: 5/2, the current value is greater than the value desired, so the pump is stopped).

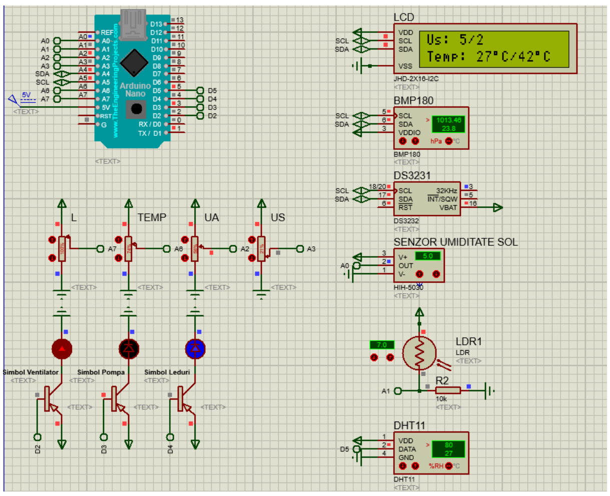

Figure 15. The simulation in the fourth case

In Figure 16 we want all functions to be disabled. We set the value of the potentiometer L to 50% (the desired light has the value 5), we set the value of the air humidity to 45% (from dht11) then we change the value of the potentiometer UA to 72% (86% humidity).

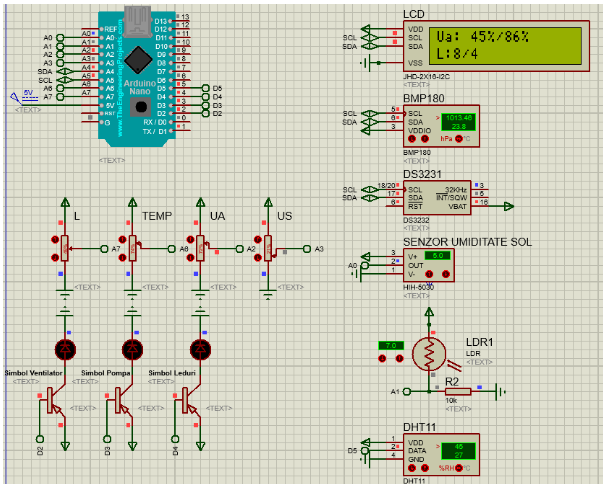

Figure 16. The simulation in the fifth case

4. IMPLEMENTING THE PROPOSED SYSTEM

The physical implementation was the most exciting part of this project. Starting the creation process, I realized that I would need a plate on which to place the greenhouse, a control panel that would show the specific values, positioned next to the greenhouse and obviously, a water tank for the pump (Figure 17).

Figure 17. The project's physical implementation

Control panel is assembled in a storage box. Its interior (Figure 18) consists of: the breadboard on which the Arduino Nano is installed, the BMP180 sensor, the DS3231 clock module, the control transistors for the pump, fan and LED, the power supply, and the connecting wires.

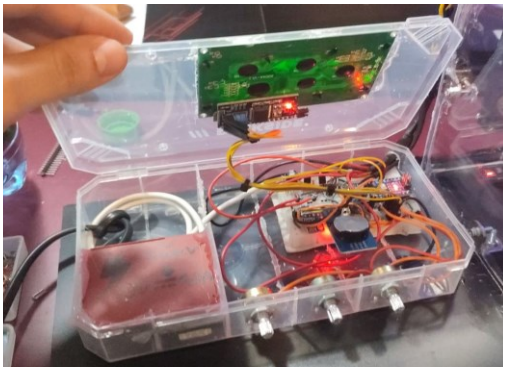

Figure 18. Control panel interior

The water tank is represented by a liquid storage container. The pump was found on the bottom of it.

The fan is placed at the back of the box, attached to it by screws. The face on which it is positioned has multiple holes in a circular shape to let air in when the fan is running, but to minimize contact with the outside air to support the greenhouse effect.

The LED panel is placed on the roof and consists of 8 UV LEDs.

In general, the project aims at a slight automation of the plant growth process. Two small containers can be placed in the premises, but only one can be checked for soil moisture (it is assumed that both containers are kept in the same environment and that they receive the same "treatment"). However, as the watering system has two nozzles, both plants can be watered equally. The temperature and humidity sensor are placed near the roof.

The clock module and the atmospheric pressure sensor supply information that cannot be changed, however, the rest of the sensors (the photoresistor, the soil moisture sensor and the dht11 sensor) have conditions that influence the automation (Figure 19).

Figure 19. The values of the sensors displayed on LCD no.1

On the LCD screen you can see several values (Figure 19). The date, time and atmospheric pressure are on the right, while the current and threshold values are on the left, for example if we have:

„*x Us/ y Us*",

Us = soil moisture,
x = is the current value provided by the sensor,
y = the threshold value that we can change with the potentiometers.

The same applies to temperature (^{0}C), air humidity (Ua) and light (L). The degrees and the humidity are the values received exactly from the sensor, while the photoresistor and the soil moisture sensor have another way of calculation. They return an analogue value between 0 and 1023 depending on the intensity of the effect: for a strong light, the photoresistor will give a high value, while for a weak light it will give a low value, the same with the soil moisture sensor, if the ground is wet, the current passes much more easily from one lamella to the other, so it provides a high analogue value, if the ground is dry, the current passes more difficult, and the value will be a small one.

Using the map() function, from the Arduino IDE, I transposed the values of the sensors from 0-1023 into values from 0-10, where 0 would represent low brightness, or a dry land and 10 would represent high brightness and wet land; I have also added a "manual" mode for the potentiometers to be able to disable the pump and the LEDs (also in Figure 19, the values of "-1" for Us and L represent the "manual" mode, they are disabled, until changing the threshold values by the user). As we read the values of analogue potentiometers (with the function AnalogRead(A0)) they are also from 0 to 1023, and with the help of the function map() are transposed in the same way as the values of the sensors (from 0 – 1023, to -1 – 10).

An example of the operation of the project can be seen in figures 20 and 21. From Figure 20 the following current values appear: 20^{0}C, 58% air humidity, 6 soil humidity and 7 brightness. The desired values are 25^{0}C, 65% air humidity, 5 soil humidity and 10 brightness. Thus, the temperature being lower than the desired value, the fan is off, the air humidity is lower than the desired one, again, the fan will be off, the soil humidity is higher than the desired one, therefore the pump will not start until it dropped below 5, and lastly, the brightness is lower than the imposed value, so the LEDs must be lit. The results can be seen in Figure 21, the fan and the pump are turned off and the LEDs are lit.

Figure 20. The values of the sensors displayed on LCD no. 2

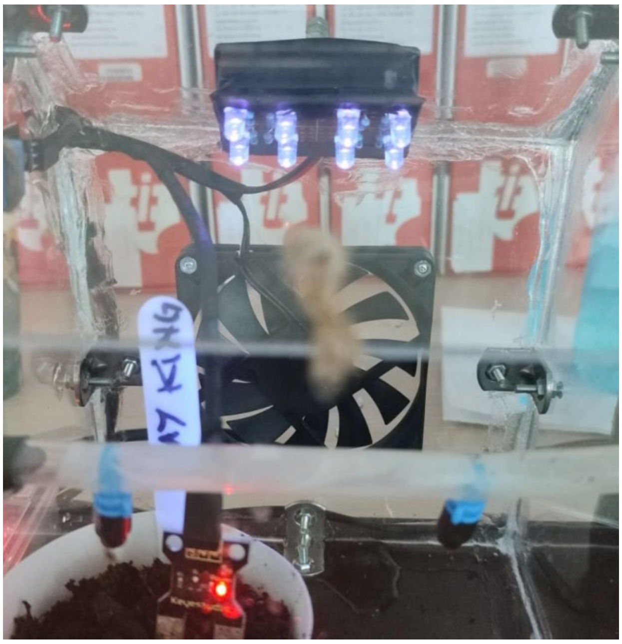

Figure 21. State of the fan and LEDs no. 1

In the second example, from Figure 22 we have the current values: 21^{0}C, 58% air humidity, 6 soil humidity and 6 brightness. Threshold values set are 25^{0}C, 40% air humidity, 5 soil humidity and 5 brightness. Thus, the fan must start (the desired air humidity is lower than the current one), the pump is stopped (the soil humidity is higher than the desired one) and the LEDs are turned off (the current brightness is higher than the desired one), things that can be seen Figure 23.

Figure 22. The values of the sensors displayed on LCD no. 3

Figure 23. State of the fan and LEDs no. 2

In Figure 24, you can see the data provided by the sensors and the state of the pump, the fan, and the LEDs. The screenshot was taken in the Serial Monitor panel of the Arduino IDE with the greenhouse connected to the computer via USB. The results were displayed at one-second intervals. According to the decision block, we have a temperature higher than the desired one (24.2^0C/20^0C) so the fan must be turned on. Soil moisture is only 3/5 so the pump must be turned on to balance the moisture. Also, the current brightness is lower than the desired one, so the LEDs are on.

```
Temperatura:24.20°C/20°C          VENTILATOR ON    8/4/2022
Umiditatea aerului:62.00%/93%     VENTILATOR ON    16:52
Umiditatea solului:3.00/5         POMPA ON
Luminozitate:6/10                 LEDURI ON        983mbar

Temperatura:24.30°C/20°C          VENTILATOR ON    8/4/2022
Umiditatea aerului:62.00%/93%     VENTILATOR ON    16:52
Umiditatea solului:3.00/5         POMPA ON
Luminozitate:6/10                 LEDURI ON        983mbar
```

Figure 24. Sensor data and the state of components no.1

In Figure 25 we have the same values for temperature and air humidity, so the fan must be turned on. This time I put the LEDs and the pump on "manual" mode (value of -1), so they will never start (the minimum value the sensors can supply is 0) without user intervention. Therefore, the pump and the LEDs are turned off.

```
Temperatura:24.10°C/20°C          VENTILATOR ON    8/4/2022
Umiditatea aerului:61.00%/70%     VENTILATOR ON    16:53
Umiditatea solului:3.00/-1        POMPA OFF
Luminozitate:7/-1                 LEDURI OFF       983mbar

Temperatura:24.10°C/20°C          VENTILATOR ON    8/4/2022
Umiditatea aerului:61.00%/70%     VENTILATOR ON    16:53
Umiditatea solului:3.00/-1        POMPA OFF
Luminozitate:7/-1                 LEDURI OFF       983mbar
```

Figure 25. Sensor data and the state of components no.2

In Figure 26 we have 24^0C/30^0C, but the air humidity is at the value of 61%/40% so the fan is on. The soil moisture is switched to manual mode, so the pump will be stopped, and the brightness is 7/8 being lower than the desired value, therefore the LEDs are on.

```
Temperatura:24.00°C/30°C          VENTILATOR ON    8/4/2022
Umiditatea aerului:61.00%/40%     VENTILATOR ON    16:54
Umiditatea solului:3.00/-1        POMPA OFF
Luminozitate:7/8                  LEDURI ON        983mbar

Temperatura:24.00°C/30°C          VENTILATOR ON    8/4/2022
Umiditatea aerului:61.00%/40%     VENTILATOR ON    16:54
Umiditatea solului:3.00/-1        POMPA OFF
Luminozitate:7/8                  LEDURI ON        983mbar
```

Figure 26. Sensor data and the state of components no.3

In the last example from Serial Monitor, Figure 27, we have the temperature value 24^0C/35^0C and the air humidity 61%/75% and the fan is off. The soil moisture is lower than the desired value (5/8), so the pump is turned on until the value exceeds 8. The brightness is again lower than the desired value (7/8), so the LEDs are on.

```
Temperatura:24.00°C/35°C          VENTILATOR OFF   8/4/2022
Umiditatea aerului:61.00%/75%     VENTILATOR OFF   16:55
Umiditatea solului:5.00/8         POMPA ON
Luminozitate:7/8                  LEDURI ON        983mbar

Temperatura:24.00°C/35°C          VENTILATOR OFF   8/4/2022
Umiditatea aerului:61.00%/75%     VENTILATOR OFF   16:55
Umiditatea solului:6.00/8         POMPA ON
Luminozitate:7/8                  LEDURI ON        983mbar
```

Figure 27. Sensor data and the state of components no.4

To highlight the results, I decided to do a small study for 9 days between two lettuce varieties (May King and Helmut), having two copies in the greenhouse and two on the desk (both being at home).

In Figures 28-31, the evolution of plants is presented. Started taking pictures from the 3rd day, this being the first time when you can see clear differences. Table 2 supplies the data for analysis in a compact form.

Figure 28. The appearance of the plants on the 3rd day (greenhouse above, room below)

Figure 29. The appearance of the plants on the 4th day (greenhouse above, room below)

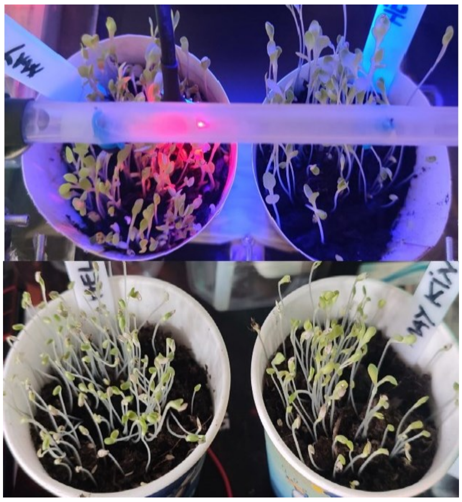

Figure 30. The appearance of the plants on the 6th day (greenhouse above, room below)

Figure 31. The appearance of the plants on the 9th day (greenhouse above, room below)

Table 2. Analysis of the experiment using lettuce from the varieties Helmut and May King.

No. day	Greenhouse	House	Comments
01-Mar	Planted + ~6 hours in the greenhouse	Planted	The seeds were planted at around 3:00 p.m. Greenhouse conditions: 22°C, 80% Ua, 10 L, 8 Us Room conditions: 21°C, 55% Ua, low light L, manual Us
02-Mar	~12 hours in the afternoon	No action	Greenhouse conditions: 22°C, 80% Ua, 10 L, 8 Us Room conditions: 21°C, 55% Ua, low light L, manual Us
03-Mar	~12 hours of greenhouse	No action	All the seeds sprouted, both in the greenhouse and at home. Greenhouse conditions: 22°C, 80% Ua, 10 L, 8 Us Room conditions: 21°C, 55% Ua, low light L, manual Us
04-Mar	~12 hours of greenhouse	Watering	A difference between the size of the plants is seen. Those in the greenhouse are higher than those in the room. The factors that influenced the growth are: 30% more humidity and 1-2 degrees more than the room environment. Greenhouse conditions: 22°C, 80% Ua, 10 L, 8 Us Room conditions: 21°C, 55% Ua, low light L, manual Us
05-Mar	~12 hours of greenhouse	No action	The tendency of the plants to move towards the greenhouse leds /room light was seen. Greenhouse conditions: 22°C, 80% Ua, 10 L, 8 Us Room conditions: 21°C, 55% Ua, low light L, manual Us
06-Mar	~12 hours of greenhouse	No action	Plant height is similar. The ones in the greenhouse, however, have larger and richer leaves than those in the room. Greenhouse conditions: 22°C, 80% Ua, 10 L, 8 Us Room conditions: 21°C, 55% Ua, low light L, manual Us
07-Mar	~12 hours of greenhouse	Watering	Greenhouse conditions: 22°C, 80% Ua, 10 L, 8 Us Room conditions: 21°C, 55% Ua, low light L, manual Us
08-Mar	~12 hours of greenhouse	No action	Greenhouse conditions: 22°C, 80% Ua, 10 L, 8 Us Room conditions: 21°C, 55% Ua, low light L, manual Us
09-Mar	~12 hours of greenhouse	No action	Took out the samples from the greenhouse and compared them with those in the room.

5. CONCLUSIONS

Considering the study done, we can conclude based on the results, that the greenhouse prototype works and is an advantage compared to the normal growth of plants.

Like any system, it has several advantages and disadvantages. Starting from the positive side, the greenhouse has a low consumption and can be connected to a phone charger (5V, 1.5A) and implicitly from the socket (~220V, 0.25A). In terms of size, I would consider it portable, it can be moved easily and does not take up much space. In terms of implementation, I could say that it is a medium-level project, it requires knowledge in the field but also creativity. When it comes to using the prototype, I think it's quite easy to use even by a novice: considering that there are four values that we can set with the potentiometers, we just must rotate them and decide the threshold values on which we will use, then visiting the plants is easy, access being through the roof.

In the sense of the disadvantages, the space of the greenhouse is quite small, it offers the possibility to watch only two samples, then the material used is a thin one, so the differences in temperatures are quite small. Also, the water tank does not have an alert signal when it is almost empty, so it requires increased attention in case it runs out of liquid.

Being a light automation, human monitoring is still needed to set data and check the liquid in the tank.

It is obvious that this project can be changed to offer better results or more efficient automation. The following could be done:
- Replacing the Arduino Nano microcontroller with another microcontroller that can connect to the Internet. In this way, we could watch and change the values of the greenhouse if it is connected to the Internet, through a server, on the web.
- Added an alert for the fluid in the tank in case it reaches a low level.
- The addition of several temperature sensors to be able to watch the lower area (close to the ground) and the upper area (close to the roof). This way we could get steps of values (because the warm air tends to be at the top of the rooms).
- Improvement of the ventilation system. A part could also be added that heats or cools the air according to the user's wishes, not just airing the room.
- Implementation of a humidity change system. We can introduce a humidifier and a dehumidifier to work according to the threshold values decided by the user. But these, together with the other possibilities of improvements will lead to the need for space to be able to incorporate all the functionalities, thus it is possible to lose both portability and consumption.

6. REFERENCES

[1] Silva C.W., Sensor systems. Fundamentals and applications, CRC Press, 2016.
[2] Rupitsch S.J., Piezoelectric sensor and actuators: fundamentals and applications, Springer, 2019.
[3] Sensor and actuators A: Physical, volume 297, Science Direct, 2019.
[4] Alegria F.A.C., Sensors and actuators, World Scientific, 2021.
[5] Innovative field sensors and actuators: from modelling to measurements, MDPI, Journals Sensors, 2020.
[6] Dawoud D.S, Dawoud P., Microcontroller and smart home networks, River Publishers, 2020.
[7] Apostol F.G., Predusca G., Advanced fan controller using Atmega 1284P, The Scientific Bulletin of Electrical Engineering Faculty (SBEEF), Issue 2, pp.43-47, 2018.
[8] Alves C., Arduino reference: syntax, concepts, and examples, Independently Published, 2021.
[9] https://en.wikipedia.org/wiki/ATmega328
[10] www.roboromania.ro
[11] Nagendra Dangi, Monitoring environmental parameters: humidity and temperature using Arduino based microcontroller and sensor, Helsinki Metropolia University of Applied Sciences, Thesis, 14 November 2017.
[12] Constantin S.C., Predusca G., Diaconu E., Pressure, temperature and humidity monitoring system using Arty platform, The Scientific Bulletin of Electrical Engineering Faculty (SBEEF), Issue 2, pp.1-5, 2021.
[13] www.sparkfun.com
[14] Brown R., Arduino Real Time Clock Tutorial (DS1302, DS1307, DS3231), 2021.
[15] Shakya A.K., Ramola A., Kandwal A., Vidyarthi A., Soil moisture sensor for agricultural applications inspired from state of art study of surfaces scattering models & semi-empirical soil moisture models, Journal of the Saudi Society of Agricultural Sciences, volume 20 (8), pp.559-572, 2021.
[16] www.instructables.com
[17] Ion F., Predusca G., Jelea A., Data acquisition system for monitoring of functional parameters of pumping stations for drinking water, The Scientific Bulletin of Electrical Engineering Faculty (SBEEF), Issue 1, pp.71-75, 2009.
[18] www.labcenter.com