Univerza *v Ljubljani*
Fakulteta *za strojništvo*

University of Ljubljana

Faculty of Mechanical Engineering

Matevž Mazovec

# SMART GREENHOUSE

Seminar Report for the Course Mechatronic Actuators

Ljubljana, May 2026

## Abstract

This seminar report describes the upgrade of an existing home greenhouse into a smart greenhouse. The basic greenhouse structure had already been built, therefore the work focused mainly on adding a mechatronic system for measurement, control and monitoring. The main goal was to design a system that automatically monitors air temperature, air humidity, soil temperature, soil moisture and light intensity, and then controls the actuators according to these measurements.

The system is controlled by an ESP DevKit V1 development board with an ESP32 microcontroller. The sensors connected to the controller are the DHT22, DS18B20, BH1750 and YL-69. The output actuators are servomotors for opening the ventilation windows, a water pump for irrigation, fans, lights and an electric heater. The system also includes a Nextion HMI display, which shows the measured values and makes it easier to monitor the operation of the greenhouse. The report describes the components, assembly process, wiring, program logic, user interface and system testing.

The final result is a functional upgrade of the greenhouse that allows basic environmental conditions to be adjusted automatically. This reduces the amount of manual work and gives the user a clearer overview of the conditions inside the greenhouse.

# Contents

# 1 Introduction

The aim of this seminar report for the course Mechatronic Actuators was to apply knowledge of sensors, control systems and actuators in a practical project. For this project I used a home greenhouse that had already been built. Because of this, the report does not focus on constructing the greenhouse from the beginning, but mainly on upgrading it into a smart greenhouse.

The basic idea was that the greenhouse should no longer work only as passive protection for plants against external conditions, but should also detect conditions inside the greenhouse and respond to them. The most important conditions in a greenhouse are temperature, air humidity, soil moisture and light intensity. If these conditions are not suitable, plant growth becomes worse, plants may dry out, overheat, or mould may start to develop. With a smart system, these conditions can be monitored and individual actuators can be switched on automatically when needed.

The report therefore describes the development of a system that measures the main environmental variables and uses the ESP32 to control devices for irrigation, ventilation, heating and lighting. The main emphasis is on the link between sensors and actuators, which is the core of a mechatronic system: sensors detect the current state, the controller processes the data and the actuators produce a physical change in the environment.

**Figure 1:** Existing home greenhouse that was upgraded into a smart greenhouse

# 2 Purpose and System Requirements

The purpose of the project was to create an affordable and understandable upgrade of an existing greenhouse. The system had to measure the most important parameters automatically and control selected actuators. It was important that the solution remained simple enough to build, while still being functional enough to demonstrate the basic principle of a smart greenhouse.

The main system requirements were:

- measurement of air temperature in the greenhouse,
- measurement of relative air humidity,
- measurement of soil temperature,
- measurement of soil moisture,
- measurement of light intensity,
- automatic irrigation when the soil moisture is too low,
- automatic ventilation when the temperature or humidity is too high,
- switching on additional lighting when the light level is too low,

- switching on the heater when the temperature is too low,
- displaying the measured values on the HMI display.

I designed the system so that it could be upgraded later. Possible upgrades include remote monitoring over Wi-Fi, data logging or additional safety functions.

## 3 Theoretical Background of a Smart Greenhouse

A greenhouse allows plants to be grown in a more controlled environment than an open garden. Its main function is to protect plants from cold, wind, heavy rain and other external influences. However, a normal greenhouse does not solve all problems, because the conditions inside still have to be monitored manually. If the temperature rises too much, the windows have to be opened or ventilation has to be switched on. If the soil is too dry, the plants have to be watered. If there is not enough light, plants perform photosynthesis less effectively.

A smart greenhouse partly automates these tasks. The system uses sensors to measure the condition of the environment and actuators to change it. This creates a closed or partly closed control loop. The measurement is the input data. The controller compares this data with a preset threshold and then decides whether to switch on the pump, fan, light, heater or servomotor.

The project is suitable for the course Mechatronic Actuators because it includes several different types of actuation. Servomotors open the ventilation openings, the pump moves water, the fans move air, the heater affects the temperature and the light affects the illumination level. In this way, one practical example shows the connection between sensors, software and the physical response of the system.

## 4 Upgrade Concept for the Existing Greenhouse

Since the greenhouse had already been built at home, the practical part began by choosing suitable positions for the sensors, actuators and control electronics. The construction of the greenhouse was therefore not the main task of the report. It was more important to decide where the measurements would be most meaningful and where the actuators would have the best effect on the conditions inside the greenhouse.

The basic system is divided into three main parts. The first part consists of sensors that collect data about the greenhouse conditions. The second part is the ESP32 controller, which processes the data. The third part consists of actuators that change the operation of the system based on the controller's decisions. The HMI display is added as a user interface, through which the measured values can be monitored.

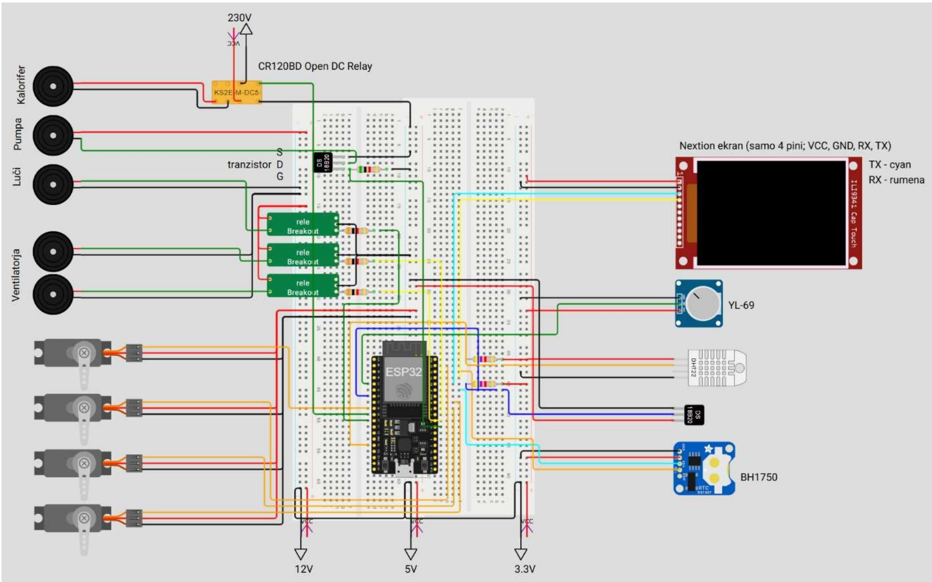


**Figure 2:** Wiring diagram of the electronic components

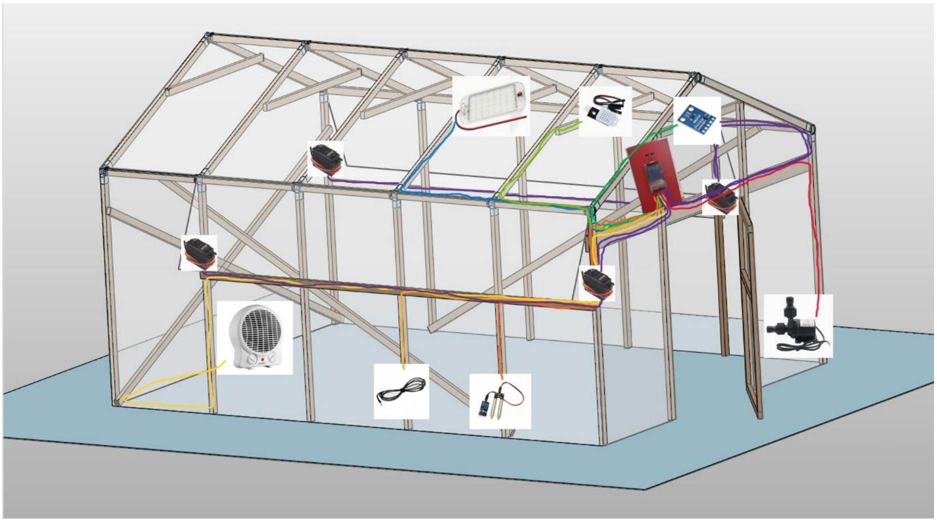

**Figure 3:** 3D view of the greenhouse and approximate position of the planned components

## 5 Used Components

The greenhouse upgrade uses components that enable measurement, control and the execution of mechanical or electrical responses. The components were selected so that the system is affordable, simple enough to assemble and suitable for demonstrating the mechatronic principle of operation.

| Component | Role in the system |
|---|---|
| **ESP DevKit V1 / ESP32** | Main controller of the system. It reads sensor data, executes the control program and controls the actuators. |
| **DHT22** | Sensor for measuring air temperature and relative air humidity. |
| **DS18B20** | Digital temperature sensor used mainly for measuring soil temperature. |
| **YL-69** | Soil moisture sensor used to decide when the water pump should be switched on. |
| **BH1750 / GY-302** | Digital light intensity sensor that measures illumination in lux. |
| **AW500S water pump** | Actuator for watering plants when soil moisture is too low. |
| **MG995 servomotors** | Actuators for opening and closing ventilation windows or flaps. |
| **Fans** | Actuators for forced ventilation and air circulation. |
| **Iskra FH-111 heater** | Actuator for heating the greenhouse when the temperature is too low. |
| **LED light** | Additional artificial lighting when natural light is too low. |
| **Nextion HMI display** | User interface for displaying measurements and monitoring system operation. |
| **Relays, MOSFET and power supply elements** | Intermediate electronic elements used for safe control of devices with higher power consumption. |

**Table 1:** Overview of the main components and their roles in the system

## 5.1 ESP DevKit V1 Controller

The central component of the system is the ESP DevKit V1 development board with the ESP32 microcontroller. Its task is to receive data from the sensors, execute the program and control the actuators through its output pins. The ESP32 is suitable for this project because it has enough input and output pins, supports analogue and digital inputs, I2C communication, UART communication and PWM signals for servomotors. It also supports Wi-Fi and Bluetooth, which makes later upgrades for remote monitoring possible.

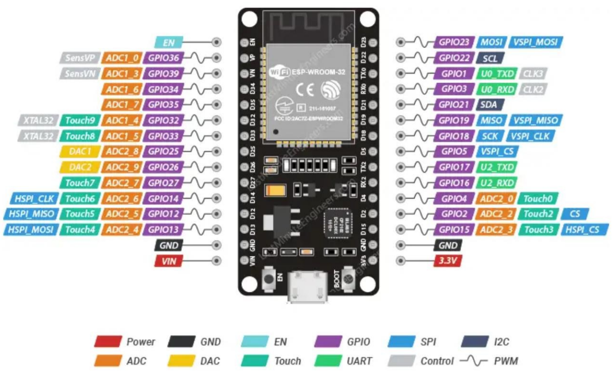


**Figure 4:** ESP DevKit V1 development board

## 5.2 Circuit, Relays and Power Supply Elements

The electronic part was first assembled on a breadboard. This method is suitable for development because it allows quick changes to the connections without soldering. Jumper wires, resistors, relays, MOSFET transistors and power supply modules were used for the connections. It is especially important that the weak output signals of the microcontroller do not directly control devices with higher power consumption. Relays or transistors must be used for this purpose.

Resistors were used with the DHT22 and DS18B20 sensors to ensure correct operation of the data lines. The pump and lights were controlled using a MOSFET or a suitable switching element. For the heater, a more powerful relay was required because the heater operates from mains voltage and has a much higher power consumption. This part of the system requires special attention, good insulation and safe wiring.

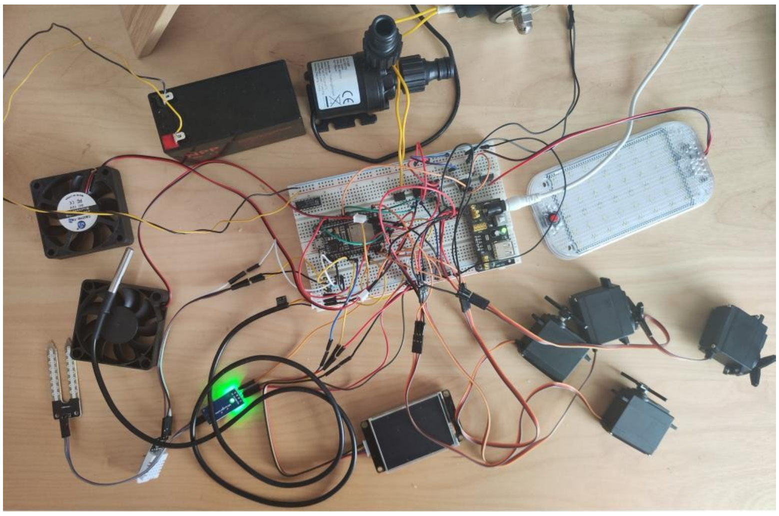

**Figure 5:** Test circuit with controller and auxiliary electronic components

### 5.3 YL-69 Soil Moisture Sensor

The YL-69 sensor was used to measure soil moisture. It works by measuring electrical conductivity between two electrodes. When the soil is wetter, conductivity increases; when the soil is dry, conductivity decreases. The sensor is connected to a module that provides an analogue output. The analogue value is read by the ESP32 and converted in the program into an approximate moisture percentage.

This value is important for automatic irrigation. When the program detects that the soil moisture is below the set limit, the water pump is switched on. When the soil is moist enough, the pump is switched off. During testing, the sensor had to be checked in different soil samples so that the threshold values could be set more realistically.

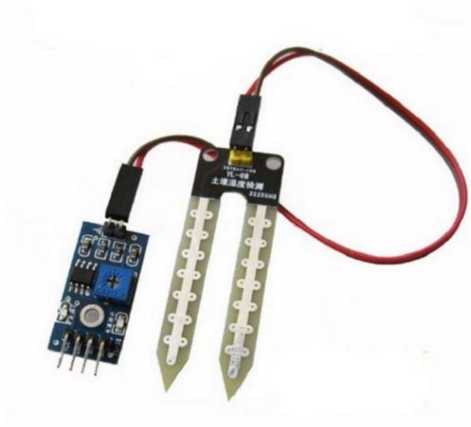

**Figure 6:** YL-69 soil moisture sensor

### 5.4 DS18B20 Temperature Sensor

The DS18B20 is a digital temperature sensor that uses the 1-Wire communication protocol. In this project it was used to measure soil temperature. The advantage of this sensor is that it sends a digital temperature value, so no additional analogue-to-digital conversion is needed on the controller side. For correct communication it requires a data line and a pull-up resistor to the supply voltage.

Measuring soil temperature is useful because the root system of plants reacts to soil temperature. If the soil is too cold, plant growth can slow down, while excessively high soil temperature can also negatively affect the plants. This value therefore provides an additional input for monitoring the state of the greenhouse.

**Figure 7:** DS18B20 temperature sensor

### 5.5 DHT22 Air Temperature and Humidity Sensor

The DHT22 is a combined sensor that measures air temperature and relative air humidity. It is very useful in a smart greenhouse because one sensor provides two important measurements. Air temperature is used for decisions about switching on the heater, fans or opening the ventilation windows. Air humidity is important because of condensation, mould and poor air circulation.

The sensor sends digital data to the microcontroller. In the program, the measured values are checked because such sensors can sometimes return an invalid value. If the program detects an invalid reading, it can treat it as an error and prevent incorrect actuator control.

**Figure 8:** DHT22 air temperature and relative humidity sensor

### 5.6 BH1750 Light Sensor

The BH1750 is a digital light intensity sensor that measures light in lux. In the system it is connected via I2C communication. Its advantage is that it already provides the measurement in a useful unit, so no complicated conversions are needed in the program. With this sensor, the system can determine whether there is enough light in the greenhouse.

When the light level is too low, the program can switch on the LED light or additional artificial lighting. This is useful mainly on cloudy days or during periods with less natural light. With longer cables, I2C communication requires attention to cable quality and, if necessary, suitable pull-up resistors.

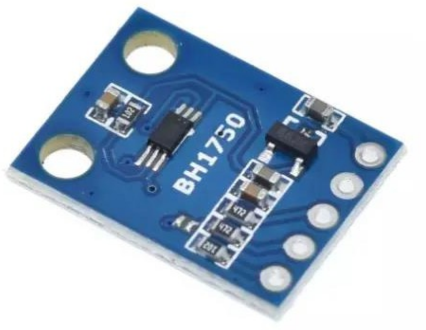


**Figure 9:** BH1750FVI light sensor or GY-302 module

### 5.7 Water Pump

The water pump is an actuator that enables automatic irrigation. A 12 V AW500S submersible pump was used in the project. The pump switches on when the YL-69 sensor detects that the soil moisture is too low. Because the pump requires more current than the microcontroller output can provide, it is controlled through a suitable switching element or MOSFET and a separate power supply.

In this implementation the system was simplified by using only the pump with water tubing, without an additional solenoid valve. This solution is simpler, cheaper and sufficient for the seminar project.

**Figure 10:** Water pump for automatic irrigation

### 5.8 Heater

An Iskra FH-111 electric fan heater was used for heating the greenhouse. Its task is to increase the temperature when the sensors detect that the temperature in the greenhouse is too low. The heater is a high-power actuator and cannot be controlled directly from the microcontroller. A suitable relay or power switching element is needed for switching it on.

Because the heater operates from mains voltage, this is the most safety-critical part of the system. The ESP32 does not power the heater directly. Instead, it sends a control signal to a relay or power element, which switches a separate power circuit for the heater. In practical implementation, proper insulation, protection against moisture and contacts rated for the required current are necessary. It is also sensible to include safety logic in the program to prevent unnecessary or unsafe heater operation at high temperatures.


**Figure 11:** Heater used for greenhouse heating

### 5.9 MG995 Servomotors

MG995 servomotors were used as mechanical actuators for opening and closing ventilation windows or flaps. A servomotor has three connections: supply voltage, ground and a PWM signal line. The position of the servomotor is determined by the pulse width or duty cycle of the signal.

With a typical servomotor, the positions are approximately related to rotations of 0 degrees, 90 degrees and 180 degrees.

In the smart greenhouse, the servomotors enable automatic ventilation. If the air temperature or humidity is too high, the windows open. When the conditions improve, the windows close again. This part of the project is important for the course Mechatronic Actuators because it directly converts an electrical control signal into mechanical motion.

**Figure 12:** MG995 servomotors for opening ventilation windows

### 5.10 Nextion HMI Display

A Nextion HMI display was used to show the measurements and monitor the basic operation of the system. HMI stands for human-machine interface. The display shows temperature, humidity, light intensity and other data in a clearer way than the serial monitor in Arduino IDE.

The user interface was created in Nextion Editor. The advantage of this type of display is that the graphical interface runs on the display itself, while the microcontroller only sends data over serial communication. This makes the program on the ESP32 simpler and gives the user a clearer overview of the system operation.

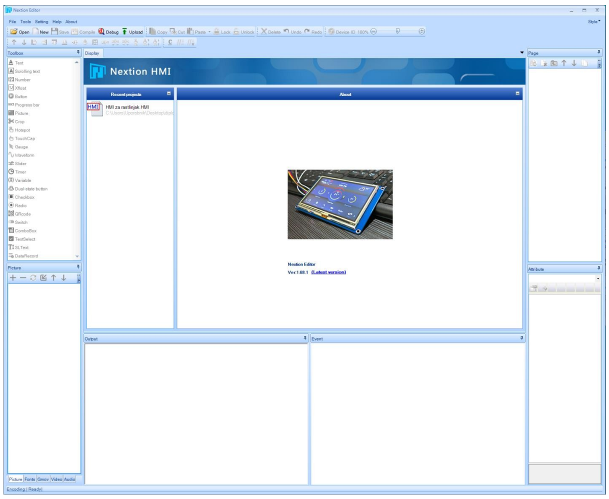


**Figure 13:** Creating the user interface in Nextion Editor

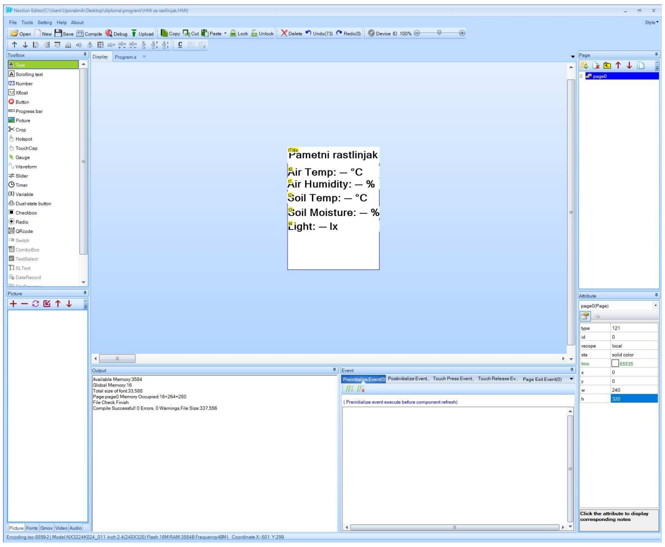


**Figure 14:** Building and editing the user interface for displaying data

**Figure 15:** Installed HMI display on the control system enclosure

## 6 Manufacturing and Installation Process

The smart greenhouse was built gradually. Since the greenhouse had already been constructed, the first step was to inspect the existing state. Then the functions that the system had to perform were defined. This was followed by component selection, breadboard testing, program writing, user interface design, installation of components in the greenhouse and final testing.

1. Inspection of the existing greenhouse and selection of positions for sensors and actuators.
2. Definition of the basic functions: irrigation, ventilation, heating, lighting and display of measurements.
3. Selection of the controller, sensors, actuators and power supply elements.
4. Assembly of the test circuit on a breadboard.
5. Basic testing of sensors and checking the measurements.
6. Writing the program in Arduino IDE and checking the logic in simulation and on the physical circuit.
7. Creating the user interface in Nextion Editor.

8. Making a protective enclosure for the electronics and installing the HMI display.
9. Installing the system in the greenhouse and final testing in real conditions.

### 6.1 Preparing the Existing Greenhouse

First, suitable positions for the individual parts of the system had to be selected. The air temperature and humidity sensor must be placed where it can measure realistic greenhouse conditions, but it must not be directly exposed to water. The soil moisture sensor has to be placed in the soil near the plants, while the soil temperature sensor should be placed close to the root area. The light sensor must be positioned so that it detects the actual light level in the greenhouse.

The actuators were placed according to their functions. The pump is connected to the water supply, the servomotors are connected to the ventilation openings, the fans provide air circulation, the light is used for additional lighting and the heater is placed so that it can affect the inside temperature. The electronics were installed in a specially made enclosure for protection.

### 6.2 Component Connections

When connecting the components, different signal types had to be considered. Some sensors use digital communication, some use an analogue input, the servomotors require PWM signals and the BH1750 uses the I2C bus. High-power actuators are not connected directly to the ESP32, but through relays or MOSFET switches.

| Component | Connection type | Purpose of connection |
|---|---|---|
| YL-69 | analogue output | analogue ESP32 input for measuring soil moisture |
| DHT22 | digital data signal | measurement of air temperature and humidity |
| DS18B20 | 1-Wire | measurement of soil temperature |
| BH1750 | I2C (SDA, SCL) | measurement of light intensity |
| MG995 servomotors | PWM | opening and closing ventilation windows |
| Pump | relay/MOSFET | switching irrigation on and off |
| Fans | relay | ventilation |
| Heater | power relay | greenhouse heating |
| LED light | relay/MOSFET | additional lighting |
| Nextion HMI | UART communication | display of measurements and communication with the user |

**Table 2:** Connections between components and the controller

## 7 Program Logic and User Interface

The program was written in Arduino IDE. Its basic structure is divided into initialization in the `setup()` function and the main `loop()` function. In `setup()`, serial communication is started, the sensors, communication buses and output pins are initialized. In `loop()`, the program continuously reads the measurements, checks their validity, displays them on the HMI screen and controls the actuators according to the set thresholds.

A simplified example of the program logic is shown below. It is not the complete program, but it represents the main principle of operation:

read air temperature and air humidity from DHT22
read soil temperature from DS18B20
read soil moisture from YL-69
read light intensity from BH1750
if soil moisture is too low: switch on the pump, otherwise switch it off
if temperature or humidity is too high: open the windows and switch on the fans, otherwise close the windows
if temperature is too low: switch on the heater, otherwise switch it off
if light intensity is too low: switch on the light, otherwise switch it off
send measured values to the HMI display

The analogue value from the soil moisture sensor was mapped to an approximate percentage. The values had to be adjusted according to actual measurements in dry and wet soil. This calibration is important because the sensor readings can change depending on soil type, mineral content and moisture level.

The user interface was designed to give the user an overview of the system operation. The display shows the main measurements, such as temperature, humidity and light intensity. This makes the system not only automatic, but also easier to monitor, because the user can quickly see whether the sensors are working correctly and what the current state of the greenhouse is.

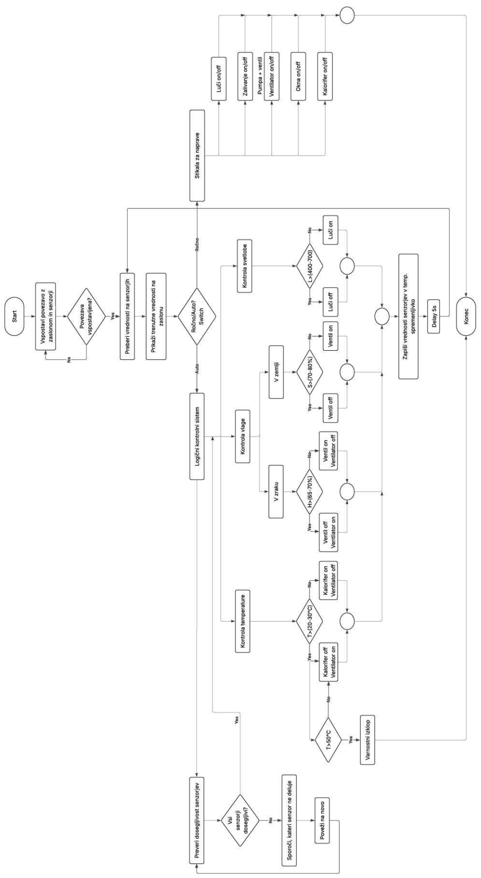

**Figure 16:** Flowchart of the program logic

## 8 System Testing

Testing was carried out in two main stages. First, the sensors and actuators were tested indoors, in a home environment. This made troubleshooting easier because the circuit was accessible and the conditions could be controlled. After that, testing continued after installation in the greenhouse, where the conditions were more realistic.

### 8.1 Indoor Testing

During indoor testing, the individual sensors were first tested separately and then together. The DHT22 was used to check air temperature and humidity, the DS18B20 for soil or ambient temperature, the BH1750 for light intensity and the YL-69 for soil moisture. With the YL-69 sensor, dry, slightly moist and very wet soil samples were used, which helped define the threshold for switching on the pump.

The actuators were also tested. The test checked whether the pump switched on correctly, whether the servomotors moved to the correct positions, whether the relays switched reliably and whether the data was displayed correctly on the screen. Step-by-step testing reduced the possibility of errors during final installation.


**Figure 17:** Indoor testing of sensors and electronics

### 8.2 Testing After Installation in the Greenhouse

After the system was installed in the greenhouse, the operation of the complete system was tested in real conditions. The sensors measured the actual conditions in the greenhouse and the actuators responded according to the set thresholds. The system displayed the measured values correctly on the HMI screen, the pump responded to soil moisture and the servomotors moved according to the ventilation conditions.

In practical use, the servomotors worked, but the mechanical solution was not the most elegant. This means that the mechanical transmission for opening the windows could still be improved. Nevertheless, the system demonstrated its basic functionality and confirmed that the sensors, controller and actuators can work together as one system.

**Figure 18:** HMI display after installation in the greenhouse

## 9 Findings, Problems and Possible Improvements

During the development of the smart greenhouse, it became clear that a useful automated system can be built with relatively affordable components. The most important finding is that the system must be designed carefully already at the stage of connecting the sensors and actuators. If the sensors are not placed correctly or if the threshold values are not set well, the controller can make wrong decisions.

One of the problems is protecting the electronics from moisture. A greenhouse is an environment with water, condensation and temperature changes, so the electronic parts must be properly protected. Another problem is the mechanical implementation of opening the windows with servomotors. The servomotor itself works correctly, but the mechanical transmission must be strong enough and must not overload the motor.

Possible improvements are:

- adding remote monitoring via Wi-Fi,
- storing measurements on an SD card or in an online database,
- adding manual mode through the HMI display,
- improving the mechanical window-opening mechanism,
- more accurate calibration of the soil moisture sensor,
- adding safety alarms for excessive temperature or sensor failure,
- making a more permanent printed circuit board instead of using a breadboard.

## 10 Conclusion

This seminar report presented the upgrade of an existing home greenhouse into a smart greenhouse. The main goal was not to build a new greenhouse structure, but to equip the existing greenhouse with sensors, actuators, a controller and a user interface. The system is based on the ESP32, which reads data from the sensors and, according to the set conditions, controls the pump, servomotors, fans, lights and heater.

During development it became clear that the most important part is the correct connection between measurements and actuator response. The program must use sensor data to correctly determine when each device should be switched on. An incorrect measurement or a poorly selected threshold can cause incorrect system operation.

During testing, the system performed the basic functions: the measurements were shown on the HMI display and the actuators responded to the set conditions. However, the solution could still be improved, especially with a more permanent circuit, better protection of the electronics from moisture, more accurate sensor calibration and improved mechanical window opening.

## 11 Program Operation Description

The program is the central part of the smart greenhouse control system. Its task is to continuously read the sensor values, check the defined thresholds and then switch individual actuators on or off according to the state of the greenhouse. The operation is designed so that all sensors, communication interfaces and outputs are initialized first. After that, measurements and control actions are performed continuously in an endless loop.

### 11.1 Program Structure

The program begins by including the required libraries. These enable communication with different sensors and modules: DHT22, DS18B20, BH1750, servomotors and the Nextion HMI display. Then the pins to which the sensors and actuators are connected are defined.

The program also contains operating thresholds. These are limit values at which a device is switched on or off. Examples include the minimum soil moisture for switching on the pump, the temperature for switching on the heater, the temperature or air humidity for opening the windows and switching on the fans, and the light level for switching on the lighting.

**Definicije knjižnic, pinov in objektov**

```
#include <Arduino.h>
#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include "DHT.h"
#include <BH1750.h>
#include <ESP32Servo.h>

#define PIN_SOIL    34      // YL-69 - analogni vhod
#define PIN_DHT     13      // DHT22 - temperatura in vlaga zraka
#define PIN_ONEWIRE 32      // DS18B20 - temperatura zemlje
#define I2C_SDA     21      // BH1750 - SDA
#define I2C_SCL     22      // BH1750 - SCL

#define FAN1_PIN    5
#define FAN2_PIN    17
#define PUMP_PIN    16
#define HEATER_PIN  26
#define LIGHTS_PIN  27

#define SERVO1_PIN  18
#define SERVO2_PIN  19
#define SERVO3_PIN  23
#define SERVO4_PIN  25

DHT dht(PIN_DHT, DHT22);
OneWire oneWire(PIN_ONEWIRE);
DallasTemperature ds18b20(&oneWire);
#define NEXTION_RX  15
#define NEXTION_TX  4

BH1750 lightMeter;
HardwareSerial nextion(2);
Servo servo1, servo2, servo3, servo4;
```

**Figure 19:** Definitions of libraries, pins, sensors and actuators in the program

**Pragovi, stanja in pomožne funkcije**

```
const int SOIL_RAW_DRY = 4095;   // vrednost v suhi zemlji
const int SOIL_RAW_WET = 1500;   // vrednost v mokri zemlji

float T_AIR_HEAT_ON  = 10.0;     // vklop grelnika
float T_AIR_HEAT_OFF = 12.0;     // izklop grelnika
float T_AIR_FAN_ON   = 28.0;     // vklop prezračevanja
float T_AIR_FAN_OFF  = 26.0;     // izklop prezračevanja
float HUM_AIR_VENT_ON  = 85.0;
float HUM_AIR_VENT_OFF = 75.0;
float LUX_LIGHTS_ON  = 3000;
float LUX_LIGHTS_OFF = 4500;
int SOIL_WATER_ON = 35;
int SERVO_CLOSED = 20;
int SERVO_OPEN = 100;

bool pumpOn = false;
bool heatOn = false;
bool fanOn = false;
bool lightsOn = false;
bool windowsOpen = false;

int soilPercentFromRaw(int raw) {
    int pct = map(raw, SOIL_RAW_DRY, SOIL_RAW_WET, 0, 100);
    if (pct < 0) pct = 0;
    if (pct > 100) pct = 100;
    return pct;
}

void setRelay(int pin, bool on) {
    digitalWrite(pin, on ? HIGH : LOW);
}

void nxSend(String cmd) {
    nextion.print(cmd);
    nextion.write(0xFF);
    nextion.write(0xFF);
    nextion.write(0xFF);
}

void setWindows(bool open) {
    int target = open ? SERVO_OPEN : SERVO_CLOSED;
    servo1.write(target);
    servo2.write(target);
    servo3.write(target);
    servo4.write(target);
    windowsOpen = open;
}
```

**Figure 20:** Set thresholds, output states and auxiliary program functions

### 11.2 System Initialization

The `setup()` function is executed only once, when the system starts. In this function, serial communication is started, output pins for relays are defined, the initial output states are set to off, and the sensors and servomotors are initialized.

It is important that all actuators are switched off at the beginning. This prevents the pump, heater, fans or lights from switching on uncontrollably when the system starts. After initialization, communication with the Nextion display is also established, so that measured values can later be shown on the screen.

**Inicializacija sistema v funkciji setup()**

```
void setup() {
    Serial.begin(115200);

    pinMode(FAN1_PIN, OUTPUT);
    pinMode(FAN2_PIN, OUTPUT);
    pinMode(PUMP_PIN, OUTPUT);
    pinMode(HEATER_PIN, OUTPUT);
    pinMode(LIGHTS_PIN, OUTPUT);

    setRelay(FAN1_PIN, false);
    setRelay(FAN2_PIN, false);
    setRelay(PUMP_PIN, false);
    setRelay(HEATER_PIN, false);
    setRelay(LIGHTS_PIN, false);

    servo1.attach(SERVO1_PIN);
    servo2.attach(SERVO2_PIN);
    servo3.attach(SERVO3_PIN);
    servo4.attach(SERVO4_PIN);

    dht.begin();
    ds18b20.begin();
    Wire.begin(I2C_SDA, I2C_SCL);
    lightMeter.begin(BH1750::CONTINUOUS_HIGH_RES_MODE);

    nextion.begin(9600, SERIAL_8N1, NEXTION_RX, NEXTION_TX);
    nxSend("page 0");
    Serial.println("System ready.");
}
```

**Figure 21:** Initialization of sensors, relays, servomotors and HMI display in the setup() function

### 11.3 Main Loop and Actuator Control

The `loop()` function is executed continuously. The program first reads the YL-69 soil moisture sensor, then the DHT22 for air temperature and relative humidity, the DS18B20 for soil temperature and the BH1750 for light intensity. The measured values are then used in conditional statements.

If soil moisture is too low, the program switches on the pump. If air temperature is too low, the heater is switched on. If air temperature or relative humidity is too high, the windows are opened with servomotors and the fans are switched on. If the light intensity is too low, the additional LED light is switched on. At the end of the loop, the current values are sent to the HMI display.

**Glavna zanka loop() in krmiljenje aktuatorjev**

```
void loop() {
    int soilRaw = analogRead(PIN_SOIL);
    int soilPct = soilPercentFromRaw(soilRaw);

    float airH = dht.readHumidity();
    float airT = dht.readTemperature();
    ds18b20.requestTemperatures();
    float soilT = ds18b20.getTempCByIndex(0);
    float lux = lightMeter.readLightLevel();

    if (!lightsOn && lux < LUX_LIGHTS_ON) {
        lightsOn = true;
        setRelay(LIGHTS_PIN, true);
    }
    if (lightsOn && lux > LUX_LIGHTS_OFF) {
        lightsOn = false;
        setRelay(LIGHTS_PIN, false);
    }

    if (!heatOn && airT < T_AIR_HEAT_ON) {
        heatOn = true;
        setRelay(HEATER_PIN, true);
    }
    if (heatOn && airT > T_AIR_HEAT_OFF) {
        heatOn = false;
        setRelay(HEATER_PIN, false);
    }

    bool needVent = (airT >= T_AIR_FAN_ON) || (airH >= HUM_AIR_VENT_ON);
    if (!fanOn && needVent) {
        fanOn = true;
        setRelay(FAN1_PIN, true);
        setRelay(FAN2_PIN, true);
        setWindows(true);
    }

    if (!pumpOn && soilPct <= SOIL_WATER_ON) {
        pumpOn = true;
        setRelay(PUMP_PIN, true);
    }

    publishToNextion(airT, airH, soilT, soilPct, lux);
}
```

**Figure 22:** Main program loop in which measurements and actuator control are executed

## 11.4 Summary of Program Operation

- The sensors represent the input part of the system because they measure the state of the greenhouse.
- The ESP32 works as the controller, comparing the measurements with the set thresholds.
- Relays and MOSFET switches allow the ESP32 to safely control devices with higher power consumption.
- The actuators physically affect the greenhouse: they water, ventilate, heat, illuminate or open the windows.
- The HMI display shows the measurements and the system state to the user.

## 12 Bill of Materials and Costs

The table shows the extended bill of materials of the components used in the project. For each item, the type of component is given, whether it is a sensor, actuator, controller or auxiliary element, along with its role in the system, quantity, price and place of purchase. This makes it clear which sensor or actuator was used and how the estimated cost of each component was obtained.

| Type | Component name | Role in the system | Quantity | Unit/total price | Place of purchase |
|---|---|---|---|---|---|
| Controller | ESP Starter Set (ESP32, breadboard, jumper wires) | Main controller and basic development kit | 1 | 20.99 € / 20.99 € | Temu |
| Sensor | DHT22 | Measurement of air temperature and relative humidity | 2 | 2.98 € / 5.96 € | Temu |
| Actuator | LED light | Additional lighting when natural light is too low | 1 | 9.48 € / 9.48 € | Temu |
| Sensor | YL-69 soil moisture sensor | Measurement of soil moisture for automatic irrigation | 1 | 3.72 € / 3.72 € | Temu |
| Auxiliary module | Soil moisture measurement module | Interface for the soil moisture sensor | 1 | 1.67 € / 1.67 € | Temu |
| Actuator | MG995 servomotors | Opening and closing ventilation windows | 1 set | 15.77 € / 15.77 € | Temu |
| Actuator | Fan | Air circulation and forced ventilation | 2 | 2.63 € / 5.26 € | Temu |
| Actuator | AW500S water pump | Irrigation when soil moisture is too low | 1 | 11.49 € / 11.49 € | Temu |
| Interface | Nextion 2.4-inch HMI | Display of measurements and system monitoring | 1 | 38.86 € / 38.86 € | Amazon |
| Sensor | DS18B20 temperature sensor | Measurement of soil temperature | 2 | 5.00 € / 10.00 € | IC Elektronika |
| Auxiliary element | Relays | Switching actuators with higher power consumption | 5 | 1.00 € / 5.00 € | IC Elektronika |
| Auxiliary element | Transistors / MOSFET | Control of pump and lights | 3 | 2.00 € / 6.00 € | IC Elektronika |
| Auxiliary element | Resistors | Pull-up resistors and protective elements | 30 | 0.03 € / 0.90 € | IC Elektronika |
| Auxiliary element | CR120BD Open DC relay | Power switching of the heater | 1 | 9.99 € / 9.99 € | eBay |
| Actuator | Iskra FH-111 heater | Heating the greenhouse when the temperature is too low | 1 | 17.59 € / 17.59 € | Mimovrste |
| Power supply | 12 V NPP battery | Power supply for more demanding components | 1 | 25.89 € / 25.89 € | Velog |
| Structure | Greenhouse construction material | Existing home greenhouse or basic structure | / | 0.00 € / 0.00 € | / |
| Mounting | Mounting material | Pipes, screws and fixing elements | 1 | 47.99 € / 47.99 € | Merkur |
| Enclosure | 3D printer filament | Production of the protective electronics enclosure | 2 | 12.99 € / 25.98 € | 3D Shark |

**Table 3:** Extended bill of materials for sensors, actuators, auxiliary elements and costs

**The total estimated component cost according to the bill of materials is approximately 262.54 €.**

It should be noted that the basic greenhouse structure had already been built at home and is therefore not included as a new cost. The bill of materials clearly marks the actuators and sensors because they are the most important part of the implemented system for the course Mechatronic Actuators.

## 13 References

The preparation of this seminar report was based on previously prepared project material about the smart greenhouse, which described the used sensors, actuators, software, testing and included images. In addition, data from documentation and descriptions of the used components were used: ESP32, DHT22, DS18B20, BH1750, YL-69, MG995 and Nextion HMI.


- Previous project material: *meh akt.docx* - description of the smart greenhouse project.
- Documentation for ESP32 and ESP DevKit V1.
- Documentation for DHT22, DS18B20, BH1750 and YL-69 sensors.
- Documentation for the MG995 servomotor and Nextion HMI display.
- Arduino IDE, Wokwi and Nextion Editor - software tools used in the development of the system.