**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH-QUALITY TRAINING**



**SENIOR 1**

**SMART GARDEN**

**MAJOR: COMPUTER ENGINEERING TECHNOLOGY**

Students: **TRAN QUANG NHAT**

ID: 20119148

**PHAN NHU KHOI**

ID: 20119189

**Advisor: HUYNH THE THIEN**

Ho Chi Minh City, May 2023

**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH-QUALITY TRAINING**

A logo of hands holding a book

Description automatically generated with medium confidence

**SENIOR 1**

**SMART GARDEN**

**MAJOR: COMPUTER ENGINEERING TECHNOLOGY**

Students: **TRAN QUANG NHAT**

ID: 20119148

**PHAN NHU KHOI**

ID: 20119189

**Advisor: HUYNH THE THIEN**

Ho Chi Minh City, May 2023

# **ACKNOWLEDGMENTS**

We express our heartfelt gratitude to Mr. Huynh The Thien for his invaluable support and facilitation throughout the completion of our Smart Greenhouse project. With limited knowledge, we acknowledge that errors may have occurred during the implementation phase. We sincerely appreciate your guidance and feedback to enhance the quality of our project.

Thank you sincerely!

# **ABSTRACT**

Smart gardens have gained significant attention as an innovative approach to gardening, leveraging technology to enhance plant care and optimize gardening practices. This paper provides an overview of smart gardens, highlighting their purpose, benefits, and underlying principles. The concept of the Internet of Things (IoT) is explored in the context of smart gardens, emphasizing its role in creating interconnected ecosystems where sensors, actuators, and control systems work together to monitor and manage plant health. Sensor technology plays a crucial role in smart gardens, enabling precise monitoring of environmental factors such as soil moisture, temperature, light, and humidity. Automation and control systems, powered by microcontrollers, facilitate the automation of tasks like watering and lighting, ensuring optimal conditions for plant growth. Data analysis and communication protocols are also discussed, emphasizing the importance of data-driven insights and seamless communication between garden components. By embracing smart gardening practices, individuals can achieve efficient, sustainable, and thriving gardens. This abstract provides a concise overview of the theoretical foundation of smart gardens, serving as a valuable resource for individuals interested in adopting this technology-driven approach to gardening.

**CONTENTS**

[**ACKNOWLEDGMENTS** 2](#_Toc136641979)

[**ABSTRACT** 3](#_Toc136641980)

[**CONTENT FIGURE AND TABLE** 5](#_Toc136641981)

[**CHAPTER 1: INTRODUCTION** 6](#_Toc136641982)

[**1.1 Problem Statement** 6](#_Toc136641983)

[**1.2 Objective** 7](#_Toc136641984)

[**1.3 A summary of the implementation steps:** 7](#_Toc136641985)

[**1.4 Result** 8](#_Toc136641986)

[**1.5 Sections of Report** 8](#_Toc136641987)

[**CHAPTER 2 THEORETICAL FOUNDATION** 9](#_Toc136641988)

[**2.1 Theoretical Foundation of smart garden** 9](#_Toc136641989)

[**2.2 Wired data transmission standards** 11](#_Toc136641990)

[**2.2.1 I2C protocol** 11](#_Toc136641991)

[**CHAPTER 3: DESIGN SYSTEM** 14](#_Toc136641992)

[**3.1 Customer Requirements** 14](#_Toc136641993)

[**3.2 Technical Requirement** 16](#_Toc136641994)

[**3.2.1 Technical Feature** 16](#_Toc136641995)

[**3.2.2 Specification** 16](#_Toc136641996)

[**3.3 Design Architecture** 17](#_Toc136641997)

[**3.3.1 Block Diagram** 17](#_Toc136641998)

[**3.4 Detailed Design** 18](#_Toc136641999)

[**3.4.1 Soil moisture sensor** 18](#_Toc136642000)

[**3.4.2 Rainwater Sensor** 20](#_Toc136642001)

[**3.4.3 Execution Module** 21](#_Toc136642002)

[**3.4.4 Central Processing** 24](#_Toc136642003)

[**3.4.5 Principal Diagram** 28](#_Toc136642004)

[**3.4.6 Creating PCB** 28](#_Toc136642005)

[**3.5 Design** 32](#_Toc136642006)

[**CHAPTER 4: SYSTEM IMPLEMENTATION AND RESULT EVALUATION** 36](#_Toc136642007)

[**4.1 Overview** 36](#_Toc136642008)

[**4.2 Implementation** 36](#_Toc136642009)

[**4.3 Result** 38](#_Toc136642010)

[**4.4 Appreciating** 40](#_Toc136642011)

[**REFERENCE** 42](#_Toc136642012)

[**APPENDIX** 43](#_Toc136642013)

# **CONTENT FIGURE AND TABLE**

[Figure 1:I2C protocol 11](#_Toc136639076)

[Figure 2: Block diagram of systems 17](#_Toc136639077)

[Figure 3: Schematic Diagram 19](#_Toc136639078)

[Figure 4: Soil moisture sensor 19](#_Toc136639079)

[Figure 5: Rainwater sensor 20](#_Toc136639080)

[Figure 6: Servo SG90 22](#_Toc136639081)

[Figure 7: Module Relay 5VDC 23](#_Toc136639082)

[Figure 8: Pin diagram of Arduino NANO 25](#_Toc136639083)

[Figure 9: LCD 16x2 27](#_Toc136639084)

[Figure 10: Simple principal diagram 28](#_Toc136639085)

[Figure 11: Systems Architecture 29](#_Toc136639086)

[Figure 12: PCB layout 30](#_Toc136639087)

[Figure 13: 3D Visualisation of PCB 30](#_Toc136639088)

[Figure 14: Front side of PCB 31](#_Toc136639089)

[Figure 15: Back side of PCB 31](#_Toc136639090)

[Figure 16: Flow chart of Task1 32](#_Toc136639091)

[Figure 17: Flow chart of Task 2 33](#_Toc136639092)

[Figure 18: IDE Arduino 34](#_Toc136639093)

[Figure 19: Complete system model. 37](#_Toc136639094)

[Figure 20: LCD information “not rain” and soil moisture 37](#_Toc136639095)

[Figure 21: Situation 1 38](#_Toc136639096)

[Figure 22: Situation 2 38](#_Toc136639097)

[Figure 23: Situation 3 39](#_Toc136639098)

[Table 1: List of components to be in systems. 36](#_Toc136639134)

# **CHAPTER 1: INTRODUCTION**

## **1.1 Problem Statement**

In today's world, automation technology has achieved tremendous advancements in both industrial production and daily life. The production of automated systems is a multi-billion-dollar industry that continues to grow rapidly. Among these systems, smart gardens and automated aquaculture farms stand out with their unique characteristics. Smart gardens are an integral part of this industry, playing a crucial role in enhancing and improving the quality of our daily intake of food, beverages, and other consumables.

## **1.2 Objective**

Here, the most vivid example is the Pilea plant, which has a delicate and slender stem. Despite its height, the stem is very thin, and its delicate petals make it vulnerable to heavy rainfall, causing the plant to bend and its flowers to easily fall off. Moreover, nowadays, rainwater is increasingly polluted, leading to various diseases and even plant death. This type of plant thrives in warm sunlight but detests rain. Therefore, we propose a solution to address this issue, which involves designing an automatic canopy that opens when the weather is sunny and immediately closes when it rains to protect the plant. This prevents direct exposure of the plant to heavy rainfall, which can cause breakage, petal loss, or even poisoning due to contaminated rainwater.

For conventional systems programmed sequentially, let's assume that when the soil moisture is low, the system proceeds to irrigate to increase the moisture level. The subsequent function is executed only after the irrigation function has completed its task. This causes delays for other functions, such as the canopy closure function mentioned earlier. As a result, the system operates inefficiently. To overcome this issue, we need to find a way to allow the functions to operate in parallel. Real-time multitasking can effectively accomplish this.

## **1.3 A summary of the implementation steps:**

- Research and familiarize yourself with the necessary hardware components.

- Design a suitable block diagram.

- Learn about real-time multitasking.

- Assemble and evaluate the system.

## **1.4 Result**

In general, the model exhibits a high degree of aesthetic appeal, effectively meeting the design standards for outdoor operational devices, thus ensuring both safety and user-friendly functionality. A noteworthy feature of the system is its ability to enable users to observe their cultivation objects without the need to approach them closely. Moreover, the system boasts an extended operational lifespan, guaranteeing the successful execution of tasks over an extended period.

## **1.5 Sections of Report**

Chapter 1: Introduction

This chapter presents the problem statement, introduces the reasons for the research topic, outlines the objectives, provides an overview of the research content, specifies the scope and limitations of the project, and presents the report structure.

Chapter 2: Theoretical Foundation

This chapter discusses the fundamental theories of the system, including an introduction to communication standards, data transmission in IoT applications, and relevant connectivity protocols.

Chapter 3: System Design

This chapter describes the process of designing the system, and technical requirements, and provides a brief overview of the components used.

Chapter 4: System Implementation and Result Evaluation

This chapter provides a detailed introduction to the designed system blocks, including detailed hardware diagrams, software algorithm flowcharts, hardware and software integration, system control operations, and a detailed overview of the system functions.

Chapter 5: Conclusion

This chapter presents the conclusion of the study, summarizes the main findings, and provides recommendations for future research or improvements.

# **CHAPTER 2 THEORETICAL FOUNDATION**

## **2.1 Theoretical Foundation of smart garden**

Smart gardens have emerged as an innovative solution for individuals looking to enhance their gardening experience and achieve optimal plant care. This chapter serves as a theoretical foundation for understanding the key concepts and principles underlying smart garden systems.

The chapter begins by introducing the concept of smart gardens and their purpose. It highlights the benefits of incorporating smart technology into gardening, such as increased efficiency, automation of tasks, and improved plant health monitoring. By leveraging advancements in technology, smart gardens enable individuals to create a more connected and intelligent gardening environment.

One crucial aspect of smart gardens is the integration of the Internet of Things (IoT) technology. This section explains how IoT plays a significant role in the context of gardening. It elaborates on how IoT enables seamless connectivity and communication between various components of the smart garden system, such as sensors, actuators, and control systems. Through IoT, these components work together to create an interconnected ecosystem that optimizes plant care.

Sensor technology is a key component of smart gardens, and the chapter dedicates a section to discuss its significance. It explores different types of sensors commonly used in smart gardens, such as soil moisture sensors, temperature sensors, light sensors, and humidity sensors. The section explains how these sensors collect crucial data about the garden's environment, enabling precise monitoring of plant conditions and the optimization of watering schedules.

Automation and control systems play a vital role in smart gardens, allowing for the automation of various tasks. The chapter explores the use of microcontrollers, such as Arduino or Raspberry Pi, as the brain of the smart garden system. These microcontrollers enable the control and monitoring of garden parameters, ensuring plants receive the necessary care. Additionally, the integration of actuators, such as irrigation systems and lighting control, automates tasks like watering and provides optimal lighting conditions for plant growth.

Data analysis is another significant aspect covered in this chapter. It emphasizes the importance of analyzing the data collected by sensors in smart gardens. By processing and analyzing this data, individuals can make informed decisions regarding plant care. Adjusting watering schedules, optimizing lighting conditions, and identifying potential issues become possible through data-driven insights.

Effective communication between the different components of the smart garden system is crucial for its smooth operation. The chapter discusses communication protocols such as Wi-Fi, Bluetooth, and Zigbee, which facilitate seamless data transfer and control. These protocols enable real-time monitoring and control of the smart garden, ensuring timely adjustments and interventions when necessary.

Overall, this chapter serves as a comprehensive guide to understanding the theoretical foundations of smart gardens. It covers topics such as IoT integration, sensor technology, automation and control systems, data analysis, and communication protocols. By grasping these fundamental concepts, individuals can develop a solid understanding of the principles and technologies involved in creating and managing a successful smart garden system.

## **2.2 Wired data transmission standards**

### **2.2.1 I2C protocol**

**a.Define**

I2C, or Inter-Integrated Circuit, is a serial communication protocol that allows multiple devices to communicate with each other using a shared bus. It was developed by Philips Semiconductors (now NXP Semiconductors) and has become a widely used standard for communication between integrated circuits in various electronic systems.

The I2C protocol uses a master-slave architecture, where one device acts as the master and initiates the communication, while the other devices act as slaves and respond to the master's commands. The devices are connected in a multi-drop configuration, meaning multiple devices can be connected to the same bus.

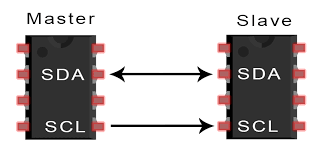


Figure 1:I2C protocol

In an I2C communication, data is transferred in the form of bytes through two bidirectional lines: the Serial Data Line (SDA) and the Serial Clock Line (SCL). The SDA line carries the actual data being transmitted, while the SCL line carries the clock signal used for synchronization.

**b.Operating work**

1. Bus Setup: The I2C bus consists of two lines: the Serial Data Line (SDA) and the Serial Clock Line (SCL). These lines are pulled high by resistors and devices connected to the bus can pull them low to transmit data.

2. Start Condition: The communication begins with a start condition, where the master device pulls the SDA line low while keeping the SCL line high. This signals the start of a communication sequence.

3. Addressing: The master device sends the address of the slave device it wants to communicate with. Each device on the bus has a unique address. The address consists of 7 or 10 bits, depending on the addressing mode used.

4. Read/Write Bit: After sending the address, the master device sends a single bit to indicate whether it wants to read from or write to the slave device. A logic high represents a read operation, and a logic low represents a write operation.

5. Data Transfer: The master device and the slave device take turns transmitting and receiving data on the SDA line. The data is transferred in 8-bit chunks. After each byte, the receiver sends an acknowledgment (ACK) bit to confirm that it has successfully received the data.

6. Clock Synchronization: The SCL line is used to synchronize the data transfer. The master device generates clock pulses on the SCL line, and both the master and slave devices perform data operations on each clock pulse.

7. Stop Condition: When the communication is complete, the master device sends a stop condition. It releases the SDA line, allowing it to be pulled high while keeping the SCL line high. This signals the end of the communication sequence.

Throughout the communication, the master device controls the timing and initiates the data transfer. The slave devices respond to the commands and data sent by the master. Multiple slave devices can be connected to the same bus, and the master can selectively address and communicate with each device using their unique addresses.

I2C supports various data transfer modes, clock speeds, and addressing schemes, allowing for flexible and efficient communication between devices in electronic systems.

One of the key features of I2C is its support for multiple devices on the same bus. Each device has a unique address, allowing the master to selectively address and communicate with specific slave devices. This enables efficient and flexible communication in systems with multiple components.

I2C also supports different data transfer modes, such as standard mode (up to 100 kbit/s), fast mode (up to 400 kbit/s), and high-speed mode (up to 3.4 Mbit/s). The mode used in a particular communication is determined by the capabilities of the devices involved.

Overall, I2C provides a reliable and efficient means of communication between devices in electronic systems, making it a popular choice for interconnecting integrated circuits in applications such as sensors, displays, EEPROMs, and other peripherals.

# **CHAPTER 3: DESIGN SYSTEM**

## **3.1 Customer Requirements**

Customer Requirements for a Smart Garden:

* Automation: The smart garden system should be able to automatically monitor and control the watering process based on the moisture levels detected by the moisture sensor. It should activate the watering mechanism when the soil moisture is below a certain threshold and deactivate it when the moisture reaches an optimal level.
* Rain Detection: The system should include a rain sensor to detect rainfall. When the rain sensor detects rain, the watering mechanism should be temporarily disabled to avoid overwatering the plants.
* Flexibility: The system should allow customization of watering schedules and moisture thresholds to accommodate different plant types and user preferences. Users should be able to easily adjust these settings without the need for an app or web interface.
* Reliability: The system should be reliable and accurate in detecting moisture levels and responding to rain conditions. It should provide consistent and accurate readings to ensure proper watering of the plants.
* Servo Control: The system should incorporate a servo motor to control additional functionalities, such as adjusting the position of sprinklers or opening/closing of a cover to protect plants from harsh weather conditions.
* User-Friendly Interface: Although the system doesn't require an app or web interface, it should provide a simple and intuitive user interface for manual control and configuration adjustments. This can be achieved through physical buttons, switches, or a small display.
* Power Efficiency: The system should be designed to operate efficiently and conserve power. It should utilize low-power components and incorporate power-saving features to extend the battery life or minimize energy consumption.
* Durability and Weather Resistance: The system should be designed to withstand outdoor conditions, including exposure to moisture, sunlight, and temperature variations. It should be constructed with weather-resistant materials to ensure long-lasting performance.
* Easy Installation: The smart garden system should be easy to install and set up, requiring minimal technical expertise. Clear instructions and documentation should be provided to guide users through the installation process.
* Cost-Effectiveness: The system should be reasonably priced, considering the features and components included. It should provide good value for money while meeting the desired functionalities and performance requirements.

These customer requirements outline the key features and expectations for a smart garden system using moisture and rain sensors, relays, and servos, without the need for an app or web interface.

**3.2 Technical Requirement**

**3.2.1 Technical Feature**

- The system can collect data from various sensors.

- The system is capable of detecting conditions that exceed the required thresholds.

- It can be controlled manually or automatically.

- The system can update data.

**3.2.2 Specification**

- The update time of sensor values: 0.5 s

- Measurement errors of the devices ± 5%

- The devices used in the system are capable of withstanding environmental factors such as high humidity.

- The system is convenient for mobility, disassembly, and connection of devices within the system, and easy to use for device connectivity.

- Ensuring safety when users encounter and control the system.

- The humidity measurement range.

The watering planting system consists of several components that work together to provide an automated way to water plants based on their moisture level. The system includes a soil moisture sensor, an Arduino microcontroller, an LCD, a water pump, a relay module, an I2C module, and an adapter 9V/2A. The soil moisture sensor measures the moisture level of the soil and sends a signal to the Arduino Nano microcontroller, which determines whether the plants need to be watered or not. If watering is required, the Arduino Nano sends a signal to the relay module, which activates the water pump to deliver water to the plants. The LCD provides feedback to the user about the moisture level of the soil and the status of the water pump. The I2C module connects the LCD to the Arduino Nano microcontroller, while the 9V battery provides power to the entire system. Together, these components create an automated and efficient system for watering plants.

## **3.3 Design Architecture**

### **3.3.1 Block Diagram**

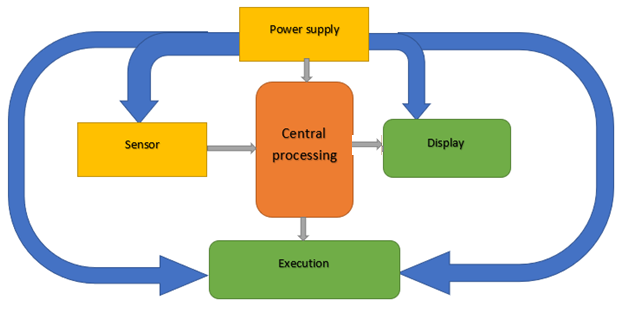


Figure 2: Block diagram of systems

* The sensor module is responsible for measuring various parameters and sending signals to the processing module.
* The execution module receives signals from the processing module to perform the opening and closing of devices.
* The central processing module receives signals from the sensor module, processes those signals, and transmits them to the display module. It also makes decisions regarding the execution module's operations.
* The display module showcases environmental parameters collected by the sensors and presents the device status.
* The power module supplies power to the operating modules.

## **3.4 Detailed Design**

Taking into consideration the specific needs and technical requirements of the system, we have developed and designed a solution that effectively fulfills the desired functionalities. With a diverse and abundant market of components available, we have carefully selected suitable components that align with both the system's demands and the preferences of the users. This selection process has been conducted thoughtfully to ensure compatibility with the research environment and to maintain originality in our approach.

### **3.4.1 Soil moisture sensor**

The Soil Moisture Sensor, commonly used in automatic watering systems and smart gardens, serves the purpose of determining soil moisture levels. It utilizes a probe to measure the moisture content of the soil and provides analog and digital output through two corresponding pins for communication with a microcontroller, enabling a wide range of applications.

* Technical specifications:

+ Operating voltage: 3.3~5VDC

* Output signals:

+ Analog: Corresponds to the supply voltage.

+ Digital: High or Low, with the desired moisture level adjustable through a potentiometer connected to the integrated LM393 comparator circuit.

* Dimensions: 3 x 1.6cm.

A picture containing text, diagram, line, plan

Description automatically generated

Figure 3: Schematic diagram

When the soil moisture sensor module detects a change in moisture, it results in a corresponding voltage variation at the input of the LM393 integrated circuit. The LM393 recognizes this change and outputs a 0V signal as an indication. The specific changes and their calculations are used to determine the soil moisture readings.

A diagram of a signal

Description automatically generated with low confidence

Figure 4: Soil moisture sensor

Here are some key points about the soil moisture sensor:

* The sensor is highly sensitive to ambient moisture levels and is commonly used to detect soil moisture.
* When the soil moisture exceeds the predefined threshold, the output of the module (D0) goes to a 0V level.
* The D0 output can be directly connected to microcontrollers such as Arduino, PIC, AVR, or STM for high and low detection, enabling soil moisture detection.
* The Analog output (AO) can be connected to an ADC (Analog-to-Digital Converter) module to obtain more precise values of the soil moisture.

### **3.4.2 Rainwater Sensor**

The Rainwater Sensor is a specialized device used to detect water droplets, water levels, rain, or any water present in the environment. This type of sensor operates like a switch. Whenever rain falls on the surface of the sensor plate, the sensor module reads the data from the sensor plate to process it and convert it into analog or digital output.

A close-up of a circuit board

Description automatically generated with low confidence

Figure 5: Rainwater sensor

The rain sensor consists of two components: the sensor board and the sensor module. Whenever rain falls on the surface of the sensor board, the sensor module reads the data from the sensor board, processes it, and converts it into either analog or digital output. Therefore, the output generated by the rain sensor can be in two forms: analog (AO) and digital (DO) signals.

The rain sensor circuit consists of two parts:

* The outdoor-mounted rain sensor component.
* The programming and sensitivity adjustment circuitry, which should be shielded.

Principle of Operation

The rain sensor operates based on comparing the voltage difference between the outdoor sensor circuit and a predefined reference value (which can be adjusted using a blue potentiometer). This voltage difference determines whether the relay is triggered or not through the DO pin.

When there is water on the sensor surface (indicating rain), the conductivity increases, resulting in lower resistance. As a result, the DO pin is pulled down to a low state (0V), and the red LED illuminates. Similarly, when the sensor is dry (indicating no rain), the conductivity decreases, leading to higher resistance. In this case, the DO pin of the rain sensor module is held at a high state (5V-12V). Therefore, the output of the rain sensor primarily depends on the resistance. It is recommended to use low-level trigger relays in conjunction with the rain sensor.

### **3.4.3 Execution Module**

**Servo SG90**

Specifications:

- Weight: 9g

- Dimensions: 22.2x11.8.32 mm

- Torque: 1.8kg/cm

- Operating Speed: 60 degrees in 0.1 second

- Operating Voltage: 4.8V (~5V)

- Operating Temperature: 0ºC - 55ºC

- Wiring Connection: Connect the red wire to 5V, the brown wire to the ground, and the orange wire to the control pulse pin of the microcontroller. Provide a pulse at the control pin ranging from 1 ms to 2 ms to control the desired rotation angle.

A picture containing text, font, diagram, screenshot

Description automatically generated



Figure 6: Servo SG90

**Module relay**

Module 1 Relay consists of a single relay that operates at a voltage of 5VDC and can withstand a voltage of up to 250VAC 10A. The module is designed with a sturdy low-level trigger and has excellent insulation capabilities. It is equipped with a relay drive circuit that uses a transistor and an optocoupler IC, ensuring complete isolation between the control circuit (microcontroller) and the relay. This guarantees stable operation of the microcontroller while maintaining isolation.

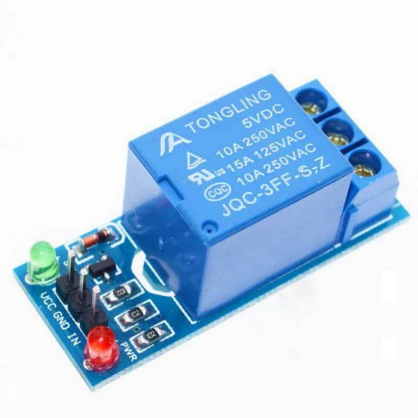


Figure 7: Module Relay 5VDC

Specification:

* Logic Level: 0V (GND)
* Control: DC or AC electrical switching, capable of controlling an AC load of 220V 10A

Available as normally open (NO) and normally closed (NC) contacts:

- NO: Normally open (closes when the control signal is activated)

- COM: Common

- NC: Normally closed (opens when the control signal is activated)

Input:

- Power Supply Voltage: 5VDC

- Control Signal Input: 0V

+ Signal level 0: The relay is closed.

+ Signal level 1: Relay is open.

Output:

+ Relay Contact: 220V 10A (Note: It refers to the contact, not the output voltage)

+ NC: Normally closed.

+ NO: Normally open

+ COM: Common terminal

Power Symbols:

+ VCC, GND: Power supply for the relay

+ IN: Control signal input terminal

### **3.4.4 Central Processing**

The central processing unit (CPU) block is required to handle and process all incoming signals obtained from sensors and actuators. It is responsible for processing the data and displaying the processed measurement values for user monitoring. The entire system's operations pass through this central processing block, which operates in a fully automated manner.

Arduino Nano is a compact and versatile microcontroller board based on the ATmega328P chip. It is a popular member of the Arduino family, known for its small size and wide range of applications. The Nano board is designed with a compact form factor, making it suitable for projects with limited space requirements.

The Arduino Nano offers similar functionality to the Arduino Uno board but in a smaller package. It has 14 digital input/output pins, 8 analog input pins, and 6 pulse width modulation (PWM) pins. This allows for the connection of various sensors, actuators, and other electronic components.

The board is equipped with a micro-USB port for power supply and programming, as well as a barrel jack for an external power source. It operates at a voltage of 5V and has a clock speed of 16 MHz.

The Arduino Nano supports the Arduino programming language, which is based on C/C++. It can be programmed using the Arduino Integrated Development Environment (IDE), making it easy for beginners and experienced users alike to develop projects.

The Nano board is widely used in projects such as robotics, automation, Internet of Things (IoT) applications, and prototyping. Its compact size, versatility, and compatibility with a wide range of shields and modules make it a popular choice among hobbyists, students, and professionals.

Overall, the Arduino Nano is a powerful and flexible microcontroller board that offers a compact solution for a wide range of electronic projects.

A close-up of a blue circuit board

Description automatically generated with low confidence

Figure 8: Pin diagram of Arduino NANO

Technical specifications:

* Designed to match the standard pinout and dimensions of the official Arduino Nano.
* Main IC: ATmega328P-AU.
* Programmer and UART communication IC: CH340.
* Supply voltage: 5VDC via USB port or 6-9VDC through the Raw pin.
* GPIO communication voltage level: TTL 5VDC.
* GPIO current: 40mA.
* Digital pins: 14, including 6 PWM pins.
* Analog pins: 8 (2 more than Arduino Uno).
* Flash Memory: 32KB (2KB bootloader).
* SRAM: 2KB.
* EEPROM: 1KB.
* Clock Speed: 16MHz.
* Integrated power indicator LED, D13 LED, RX, TX LEDs.
* Integrated 5V voltage regulator IC, LM1117.
* Dimensions: 18.542 x 43.18mm.

ATmega328P-AU is a microcontroller integrated circuit (IC) used in various electronic devices and development boards such as Arduino. It is part of the popular AVR family of microcontrollers manufactured by Atmel (now Microchip Technology). The "ATmega" prefix indicates that it belongs to the AVR series, while "328P" specifies the specific model with enhanced features.

The ATmega328P-AU features a high-performance 8-bit RISC architecture with a wide range of peripherals and memory options. It operates at a clock speed of 16MHz and can execute instructions at a rate of one per clock cycle. The "AU" suffix denotes the surface-mount package type, specifically the TQFP package.

This microcontroller provides a wide range of functionality, including digital input/output (I/O) pins, analog-to-digital converters (ADC), serial communication interfaces (UART, SPI, and I2C), timers, and interrupts. It also has built-in flash memory for program storage, SRAM for data storage, and EEPROM for non-volatile data storage.

The ATmega328P-AU is widely used in various embedded systems, robotics, IoT devices, and DIY projects due to its versatility, ease of programming, and compatibility with the Arduino ecosystem.

**Display module**

The 1602 green LCD utilizes the HD44780 driver, capable of displaying 2 lines with 16 characters per line. This display offers high durability, is widely used, and comes with numerous sample codes, making it user-friendly and easily accessible.

Specifications:

* Operating voltage: 5V.
* Dimensions: 80 x 36 x 12.5mm.
* White characters with a blue backlight.
* Convenient 0.1-inch spacing between connecting pins, suitable for Breadboard connections.
* Pin labels are printed on the back of the LCD screen for easy wiring and connection.
* Equipped with a backlight LED, brightness can be adjusted using a potentiometer or PWM to reduce power consumption.
* Can be controlled with 6 signal wires.

A picture containing text, display, screenshot

Description automatically generated

Figure 9: LCD 16x2

**Power supply**

We have chosen to use a 7-9V power supply through a voltage regulator IC to provide power to the sensor and control unit.

### **3.4.5 Principal Diagram**

**A computer screen shot of a computer program

Description automatically generated with low confidence**

Figure 10: Simple principle diagram

### **3.4.6 Creating PCB**

We design printed circuit boards for stability and aesthetics.

A diagram of a power supply system

Description automatically generated with medium confidence

Figure 11: Systems architecture

Our system diagram consists of four components: the power section, which aims to convert the 9V power supply from the adapter into a stable 5V power source for the sensors, microcontroller, and actuator. The actuator section includes a servo motor for operating the curtain and a pump. The sensor section comprises a soil moisture sensor and a rain sensor. The central processing unit is an Arduino NANO.

**A computer screen shot of a computer program

Description automatically generated with low confidence**

Figure 12: PCB layout

**A computer screen shot of a computer

Description automatically generated with medium confidence**

Figure 13: 3D Visualisation of PCB

**A picture containing toy, electronics, electronic engineering, person

Description automatically generated**

Figure 14: Front side of PCB

**A picture containing circuit component, electronic component, passive circuit component, circuit

Description automatically generated**

Figure 15: Back side of PCB

The PCB circuit serves as a physical platform for mounting and interconnecting various electronic components, such as resistors, capacitors, integrated circuits, and connectors. It provides a structured and organized layout that ensures proper electrical connections and signal flow.

## **3.5 Design**

To tackle the initial issue, I divided the necessary tasks into three parts and executed them simultaneously using an RTOS.

Task 1: Monitoring the rain sensor and controlling the servo motor for curtain operation

A picture containing diagram, text, line, screenshot

Description automatically generated

Figure 16: Flow chart of Task1

The first step involves reading the value from the rain sensor. If the state is equal to 1, the servo motor is activated to close the curtain as a measure against rain. Subsequently, it returns to the beginning to re-read the state value. If the state is equal to 0, the curtain is retracted, and the process returns to the initial step of reading the sensor value.

Task 2: Reading the soil moisture sensor and controlling the water pump

A diagram of a flowchart

Description automatically generated with low confidence

Figure 17: Flow chart of Task 2

Most plants thrive and grow well within a moisture range of 30-70%. Therefore, we will pump water when the moisture level falls below 30% and stop when it exceeds 70%. If the sensor value falls within the range of 30-70%, no watering is required.

Firstly, the soil moisture sensor value is read. If the moisture level is below 30%, the water pump is activated, and simultaneously, the soil moisture sensor value is read (without returning to the initial step). The value is then compared to the upper limit of 70%. If it is still below 70%, the process returns to the step of activating the water pump. If, during the comparison, the value exceeds the upper limit, the water pump is turned off, and the process returns to the initial step. The same process is applied to the lower limit comparison, returning to the initial step if it is not met.

Task 3: Display value LCD to flow

Three tasks interact through global variables and the information presented on the screen

**Introducing about software Arduino IDE**

Arduino IDE (Integrated Development Environment) is a software application specifically designed for programming Arduino boards. It provides a user-friendly platform for writing, compiling, and uploading code to Arduino microcontrollers. The IDE supports the Arduino programming language, which is based on Wiring, and allows developers of all skill levels to create and upload their programs to Arduino boards.

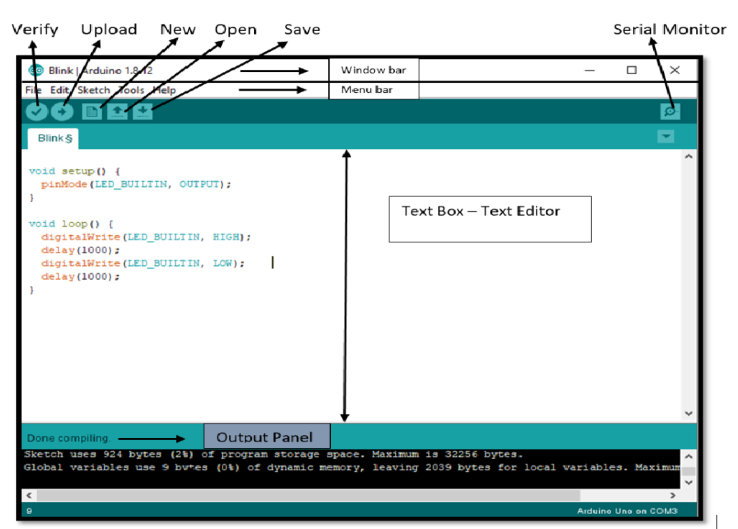


Figure 18: IDE Arduino

With Arduino IDE, users can write code using a simplified and intuitive programming language, like C and C++. The software provides a code editor with syntax highlighting, auto-completion, and error checking, making it easier to write and debug code. It also offers a range of built-in functions and libraries that simplify the development process and enable interaction with various sensors, actuators, and other peripherals.

Arduino IDE supports multiple Arduino board models, allowing users to select the appropriate board and port for their project. It provides a straightforward process for compiling and uploading code to the Arduino board, ensuring seamless execution of the programmed functionalities.

Additionally, Arduino IDE offers a serial monitor feature, which allows users to communicate with their Arduino board and monitor the output of their program in real time. This feature is particularly useful for debugging and troubleshooting purposes.

# **CHAPTER 4: SYSTEM IMPLEMENTATION AND RESULT EVALUATION**

## **4.1 Overview**

Based on the schematic diagram, suitable components were chosen. We proceeded to construct the model according to the system design specifications.

Table 1: List of components to be in systems.

|  |  |  |
| --- | --- | --- |
| Name | Quantity | Function |
| Soil moisture sensor | 1 | Measure soil moisture |
| Rain sensor | 1 | Read state sensor |
| Servo SG90 | 1 | Roll up/down the curtain |
| Water pump | 1 | Pump water for condition |
| LCD 16x2 | 1 | Display value |
| Arduino NANO | 1 | Utilized for the processing unit of the central unit. |
| Relay 5VDC | 1 | Isolate the motor from the central processing unit to ensure stable operation. |

## **4.2 Implementation**

In this model, we design and utilize materials made from form, a cost-effective and commonly available material in the market, suitable for prototyping. After assembling the circuit on the board, we proceed to wire and connect the devices and components.

A picture containing electronics, electrical wiring, electronic engineering, cable

Description automatically generated

Figure 19: Complete system model.

A picture containing text, electronics, battery, indoor

Description automatically generated

Figure 20: LCD information “not rain” and soil moisture

After conducting research and completing the project, we have successfully developed a system model that meets the requirements and achieved specific outcomes throughout the implementation process.

* The system incorporates parallel processing of multiple tasks.
* The measurement system meets the specified requirements and collects data from sensors with high accuracy.
* The components are arranged logically.
* The system meets the user's specified requirements.
* The system operates reliably and maintains stability.

## **4.3 Result**

**Situation 1:**

* Humidity < 30
* Turn on the pump.
* The red light is on.
* No rain, so the curtains are open.

A picture containing text, electronics, screenshot, digital clock

Description automatically generatedA picture containing electronics, electronic engineering, circuit component, electronic component

Description automatically generated

Figure 21: Situation 1

**Situation 2:**

* When it rains,
* Despite the pump still being in operation.
* The curtain has been activated.



Figure 22: Situation 2

**Situation 3:** When the soil moisture reaches a level higher than 70%, the irrigation process should be stopped. Additionally, the red light should be turned off.

A picture containing electronics, electrical wiring, cable, electronic engineering

Description automatically generated

Figure 23: Situation 3

After conducting research and implementing the project, our team completed the Smart Garden project. Throughout the process, we have gained new knowledge and achieved the following results:

- A deeper understanding of Arduino NANO and its programming techniques.

- Utilizing Arduino for communication with expansion modules such as soil moisture sensors, rain sensors, and more.

- Acquiring skills in PCB construction and component selection.

- Developing a sensor measurement system capable of capturing various data points.

- Implementing an automated control system.

- Running tasks concurrently, rather than sequentially.

- Applying RTOS in system development.

Overall, this project has allowed us to expand our knowledge base, gain practical skills, and successfully apply various technologies to build the Smart Garden system.

## **4.4 Appreciating**

Throughout the research, exploration, and implementation process of the system, we encountered some challenges in modifying the system and software to meet the specified requirements. However, there is still a limitation that it has not been applied on a large scale.

Advantage:

* Affordable cost
* Easy to upgrade.
* Easy to maintain.
* Easy to manufacture.

Disadvantage:

* Minimal care modes
* Small scale

Development direction:

* Simultaneous control of multiple planting areas at once
* Additional upgrade of specialized care modes.

**CHAPTER 5: CONCLUSION**

After conducting research and completing the project, we have successfully developed a system model that meets the requirements and achieved specific outcomes throughout the implementation process.

* The system incorporates parallel processing of multiple tasks.
* The measurement system meets the specified requirements and collects data from sensors with high accuracy.
* The components are arranged logically.
* The system meets the user's specified requirements.
* The system operates reliably and maintains stability.

# **REFERENCE**

[1] Grey Perry and Dean Miller – “C Programming Absolute Beginner's Guide 3rd Edition”, QUE Publishing, 2014

[2] Datasheet Arduino Nano. Link: <https://components101.com/microcontrollers/arduino-nano>

[3] Smart Garden System with Arduino Nano IoT. Link:

<https://www.hackster.io/gatoninja236/smart-garden-system-with-arduino-nano-iot-791933>

# **APPENDIX**

**Source code:**

A screenshot of a computer program

Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with medium confidenceA screenshot of a computer program

Description automatically generated with medium confidence