Computers and Information Technology
2025

PIIRRIGATE: A SMART IRRIGATION SYSTEM

Candidate: Virgil-Alexandru CRIȘAN

Scientific coordinator: Conf. dr. ing. Răzvan BOGDAN

Session: June 2025

REZUMAT

Această lucrare descrie proiectarea și realizarea sistemului “PiIrrigate”. Acest sistem are ca scop eficientizarea consumului de apă si optimizarea culturilor agricole sau a grădinilor, prin utilizarea tehnologiilor moderne IoT și a comunicațiilor radio de tip LoRa. Pe măsură ce efectele încălzirii globale se fac din ce în ce mai resimțite, automatizarea si monitorizarea irigațiilor devine o necesitate. Proiectul vizează dezvoltarea unui sistem de monitorizare si control destinat fermierilor care doresc monitorizarea unor suprafețe mari de teren, dar acest sistem poate fi folosit si pentru sere inteligente sau grădinărit normal.

Proiectul utilizează o arhitectură bazată pe microcontrolere Raspberry Pi și T-Beam LILYGO ESP32 LoRa. Aceste componente sunt folosite pentru colectarea si transmiterea datelor in timp real. Sistemul permite monitorizarea parametrilor esențiali (umiditate, temperatură, umiditatea solului și cantitatea de ploaie) prin senzori care sunt conectați la nodurile ESP32. Dupa colectare, datele urmează a fi transmise către un gateway care comunică apoi cu un API web dezvoltat in .NET. Apoi datele urmează a fi stocate într-o baza de date PostgreSQL și trimise folosind SignalR către o aplicație web pentru a fi vizualizate în timp real de către utilizatori.

Utilizatorii au la dispoziție o interfață web care le permite atât vizualizarea datelor în timp real cât și vizualizarea datelor istorice și controlul manual al sistemului. De asemenea, acest sistem implementează un mecanism de înregistrare dinamica a nodurilor în rețea, lucru care permite extinderea facilă a sistemului.

Prin integrarea componentelor hardware și software într-o soluție coerentă, PiIrrigate demonstrează fezabilitatea și eficiența unui sistem IoT dedicat agriculturii inteligente, cu un impact potențial in reducerea consumului de apă și creșterea randamentului agricol.

ABSTRACT

This paper describes the design and implementation of the “PiIrrigate” system. The purpose of this system is to optimize water consumption and improve the management of agricultural crops or gardens by using modern IoT technologies and LoRa radio communications. As the effects of global warming become increasingly evident, the automation and monitoring of irrigation systems is becoming a necessity. The project aims to develop a monitoring and control system intended for farmers who need to oversee large areas of land, but it can also be used for smart greenhouses or regular gardening.

The project uses an architecture based on Raspberry Pi microcontrollers and T-Beam LILYGO ESP32 LoRa modules. These components are used for the real-time collection and transmission of data. The system enables the monitoring of essential parameters (humidity, temperature, soil moisture, and rainfall) through sensors connected to ESP32 nodes. After collection, the data is transmitted to a gateway, which then communicates with a web API developed in .NET. The data is stored in a PostgreSQL database and sent in real time via SignalR to a web application, where it can be viewed by users. Users have access to a web interface that allows them to visualize both real-time and historical data, as well as to manually control the system. Additionally, the system implements a dynamic node registration mechanism, enabling easy expansion of the network. By integrating both hardware and software components into a coherent solution, PiIrrigate demonstrates the feasibility and efficiency of an IoT-based system dedicated to smart agriculture, with the potential to reduce water consumption and increase agricultural productivity.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SNIPPETS

1. INTRODUCTION

1.1 CONTEXT

Agriculture is a vital sector that plays a crucial role in sustaining human life and the economy. Agriculture automation and optimization has become a major concern in recent years. As the global population continues to grow, the demand for food is increasing and the developing need for food along with the effect of climate changes are forcing the agricultural industry to adapt and innovate[1].

In the last 35 years, the wold has seen a doubling of the agricultural production. This has been achieved throught the use of different fertilizers, pesticides and herbicides. This doubling was associated with a 6.87-fold increase in nitrogen fertilization, a 3.48-fold increase in phosphorus fertilization and 1.68-fold increase in the amount of irrigated cropland [2]. In addition, the water consumoption is expected to increase by 50% by 2050 [3].

This project aims to address the challenges of water scarcity and the need of efficient irrigation systems by presenting the plan, the implementation, the results and future work of a system that can be used in different scenarios and is meant to help reducing the water consumption and increasing the agricultural productivity. The PiIrrigate project intends achuive this by developing an innovative irrigation system that leverages the power of IoT and LoRa radio communication technologies. The main focus of this project is to create a system that can be easly used in different agricultural settings, starting from small gardens to large farms and even smart greenhouses. Beside this, I wanted to create a system that is easy to use and can be extended with ease.

The ESP32 boards with sensors are responsible for colecting the data. Then data is sent using LoRa to another ESP32 board that acts as a gateway connected to a Raspberry Pi, which is responsible for sending the data to a web API. The web API is developed in .NET and is responsible for storing the data in a PostgreSQL database. The data is then sent to a web application using SignalR, which allows real-time communication between the server and the client. The web application is responsible for displaying the live data and the historical data and also for providing a way to control the system manually and to add new nodes to the system.

This system takes advantage of the LoRa radio communication technology, which allows for long-range communication with low power consumption. Meaning that the system can be used in remote areas and it will work even if the internet connection is not available to all the nodes. The Raspberry Pi is the only component of this system that needs to be connected to the internet. Other components can be scatered on a area of 10km or more, depending on the environment and the node setup (mesh or star topology).

1.2 MOTIVATION

The reason why I choose to create such a system was fulled by my passion for technology and smart agriculture. Besides this, I like to observe the data path, from the moment it is collected by the sensors, to the moment it is displayed in a web application. I have always been intested in pieces of technology that can be used to solve real world problems and now I had the chance to create such a system.

Initially, I wanted to create a system for my own lawn, but as I started working on the project, I realized that the system can be used in many other scenarios, such as smart greenhouses or even large farms or vineyards.

2. STATE OF THE ART

2.1 INTRODUCTION

The state of the art chapter provides an overview of the current state of smart irrigation systems and their applications in agriculture. This chapter will explore the existing technologies, methods, and solutions used in smart irrigation. It will also highlight the gaps and challenges in the current systems, and how the PiIrrigate project aims to address these issues.

2.2 EXISTING SMART IRRIGATION SOLUTIONS

2.2.1 TYPES OF SMART IRRIGATION SYSTEMS

There are several types of smart irrigation systems used in modern agriculture:

- **Weather-Based Controllers**
  These systems use weather data to adjust irrigation schedules based on evapotranspiration rates, ensuring that plants receive the right amount of water.
- **Soil Moisture-Based Controllers**
  These systems rely on data from soil moisture sensors placed in the root zone of the plants. Irrigation cycles are triggered when the soil moisture drops below a predetermined threshold, ensuring plants receive water only when necessary. This method is very precise for specific zones[4].
- **Hybrid Systems**
  Many modern systems utilize a hybrid approach, combining data from both weather feeds and soil moisture sensors for more accurate and resilient irrigation decisions. Some research also explores "hybrid" in terms of integrating different energy sources (e.g., solar and wind) to power the systems or combining various irrigation methods (like drip and sprinkler) under one smart control[5].

  The PiIrrigate project place itself in the category of hybrid systems, using both soil moisture sensors and weather sensors to collect data.

  Some of the most popular hybrid smart irrigation systems include:

  – **Netafim's Precision Irrigation System**
    This system cobines data from soil moisture and flow sensors with sattelite weather data and predictive analytics to optimize the irrigation proccess.
    Key features include:
    * Real-time monitoring of soil moisture levels and weather forecasts.
    * Automated irrigation scheduling based on weather forecasts.

    * AI-based algorithms to optimize the irrigation timing and duration.

- **CropX Smart Farming System**
  CropX is a cloud based platform that integrates soil moisture sensors, weather data and machine learning algorithms to optimize the irrigation process.
  Key features include:
    * Irrigation recomandations based on soil variability, crop type and weather.
    * Farmers can apply recommandations or integrate with automated irrigaitons contrllers.
    * Easy to scale and adapt to different farm sizes from small to large-scale farms.
- **Toro EVOLUTION® Series Controller with Smart ET Sensor**
  Combines basic sprinkler system hardware with smart sensors and connectivity, offering both manual and intelligent irrigation options. The evaporation sensors are used to measure the amount of water lost throught evaporation and adjust the irrigation accordingly.
  Key features include:
    * Can be programmed manually or connected to a local weather station.
    * Smart ET sensor measures evaporation rates and adjusts irrigation schedules.
    * Compatible with smart devices for remote monitoring

2.3 COMPARATIVE ANALYSIS OF SMART IRRIGATION SYSTEMS

The table below provides a comparative analysis of some of the most popular smart irrigation systems available today and the PiIrrigate system.

Feature	**Netafim**	**CropX**	**Toro ET**	**PiIrrigate**
Irrigation Type	Drip	Any	Sprinkler	Custom
Automation	High (AI)	Med–High	Medium	Medium
Sensors	Soil, flow, weather	Soil, temp	ET sensor	Soil, temp, rain
Weather Data	Yes	Yes	Yes	Optional
Manual Control	App/cloud	App/web	Panel/app	Web UI
AI/Analytics	Yes	Yes	No	No
Scalability	Large farms	Small–large	Residential	Small farms/gardens
Cloud Sync	Yes	Yes	Optional	Yes
Use Case	Precision agri	Smart farming	Lawn care	Small to large scale
Cost	High	Med–High	Low–Mid	Low

Table 2.1: Comparison of Smart Irrigation Systems

2.4 IDENTIFIED GAPS AND CHALLENGES

Despite the advancements in smart irrigation systems, several gaps and challenges remain:

- **High Costs**
  Many existing systems are expensive, making them inaccessible for small farmers or home gardeners.
- **Complexity of Use**
  Some systems require specialized knowledge to set up and maintain, which can be a barrier for adoption.
- **Limited Customization**
  Many systems are designed for specific crops or environments, limiting their applicability in diverse agricultural settings.
- **Data Integration Issues**
  Integrating data from multiple sources (e.g., weather, soil sensors) can be challenging, leading to inefficiencies in irrigation management.

Bedside the presented gaps, there are also some other challenges that need to be addressed, such as: data security and privacy concerns, the need for reliable internet connectivity in remote areas, and the need for systems that can operate in harsh environmental conditions.

Another aspect that needs to be considered is the environmental impact of smart irrigation systems. While these systems are designed to optimize water usage, the production and disposal of electronic components can have a negative impact on the environment. So another challenge is to create systems that are environmentally friendly and sustainable. This means that the components used in these systems shlould be madea from recyclable materials and the systems should be designed to have a long lifespan and to be easily repairable.

2.5 SUMMARY

In summary, the state of the art in smart irrigation systems shows significant advancements in technology and methods, but also highlights several gaps and challenges that need to be addressed. The PiIrrigate project aims to fill these gaps by providing a cost-effective, easy-to-use, and customizable solution that leverages the power of IoT and LoRa radio communication technologies.

3. USED TECHNOLOGIES

3.1 OVERVIEW

The Internet of Things (IoT) is a new technology that allows devices to connect remotely to achieve smart farming [6]. The IoT has a wide range of applications in agriculture, and it has began to influence many other industries as well, such as healthcare, transportation, and manufacturing. This was done to improve the efficiency and productivity of these industries, as well as to reduce costs and improve the quality of products and services[7].

The PiIrrigate smart irrigation system is build using a combination of hardware and software technologies. It leverages both low-power edge devices and cloud-based infrastructure to provide real-time monitoring and data collection, as well as remote control capabilities. The core components and their roles in the system are as follows:

- **ESP32 (LilyGo T-Beam)**
  The LilyGo T-Beam is a development board based on the ESP32 microcontroller, it is equipped with LoRa radio communication capabilities, Wifi, Bluetooth, GPS, and a battery management system. It is used to collect data from sensors and send it to the gateway using LoRa radio communication.

- **Raspberry Pi**
  Raspberry Pi is a small, affordable computer that can be used for a wide range of applications. The Raspberry Pi is the core of the PiIrrigate irrigation module, it is responsible for receiving data from the ESP32 nodes and sending it to Azure IoT Hub.

- **Azure IoT Hub**
  Azure IoT Hub is a cloud-based service that enables secure and reliable communication between IoT devices and the cloud. It manages the bidirectional communication between the Raspberry Pi and the web API.

- **Web API**
  The web API is developed in .NET and is responsible for receiving data from the Raspberry Pi, storing it in a PostgreSQL database, and providing a way to access the data. SignalR is used to provide real-time communication between the server and the client. It also provides a way to control the system manually and to add new nodes to the system.

- **PostgreSQL Database**
  PostgreSQL is a powerful, open-source relational database management system. It is used to store the data collected from the sensors and the schedules sent to the system. It also stores the user data and the configuration of the system. The database is hosted in Neon.

- **Web Application**
  The web applicaiton is developed using Angular and is responsible for displaying live, historical data and provide the user interface for controlling the system.

In the following sections, we will explore each of these components in more detail, starting with the hardware components and then moving on to the software components.

3.2 HARDWARE COMPONENTS

3.2.1 SENSORS

The PiIrrigate system uses a variety of sensors to collect data from the environment. The sensors used in the PiIrrigate system are:

- **Soil Moisture Sensor**
  The soil moisture sensor is used to measure the moisture level in the soil. It is used to determine when to irrigate the plants. The sensor is connected to the ESP32 board and sends the data to the gateway using LoRa radio communication.
- **Temperature and Humidity Sensor**
  The temperature and humidity sensor is used to measure the temperature and humidity of the environment. It is used to determine the optimal conditions for plant growth and to adjust the irrigation schedule accordingly.
- **Rain Sensor**
  The rain sensor is used to detect rain and prevent irrigation during rainy weather. It helps to conserve water and prevent over-irrigation.

3.2.2 ESP32 (LILYGO T-BEAM)

The T-Beam ESP32 LoRa Wireless Module is a compact development board thaht combines an ESP32 microcontroller, LoRa transceiver (SX1278), GPS module, and a battery management system into a single unit. This board is ideal for long-range, low-power IoT applications such as mesh networks, asset tracking, smart agriculture and environmental monitoring. Besides this, it has a built-in OLED display. The communication range of the LoRa transceiver can reach up to 10 km in open areas.

Figure 3.1: ESP32 (LilyGo T-Beam) module used in PiIrrigate

3.2.3 RASPBERRY PI 4 MODEL B

The Raspberry Pi 4 Model B is a small, affordable computer that can be used for a wide range of applications. It is equipped with a quad-core ARM Cortex-A72 processor, up to 8GB of RAM, and supports dual-band Wi-Fi and Bluetooth. The Raspberry Pi 4 Model B together with an ESP32LoRa is used in the PiIrrigate project as the gateway that receives data from the ESP32 nodes and sends it to IoT Hub.

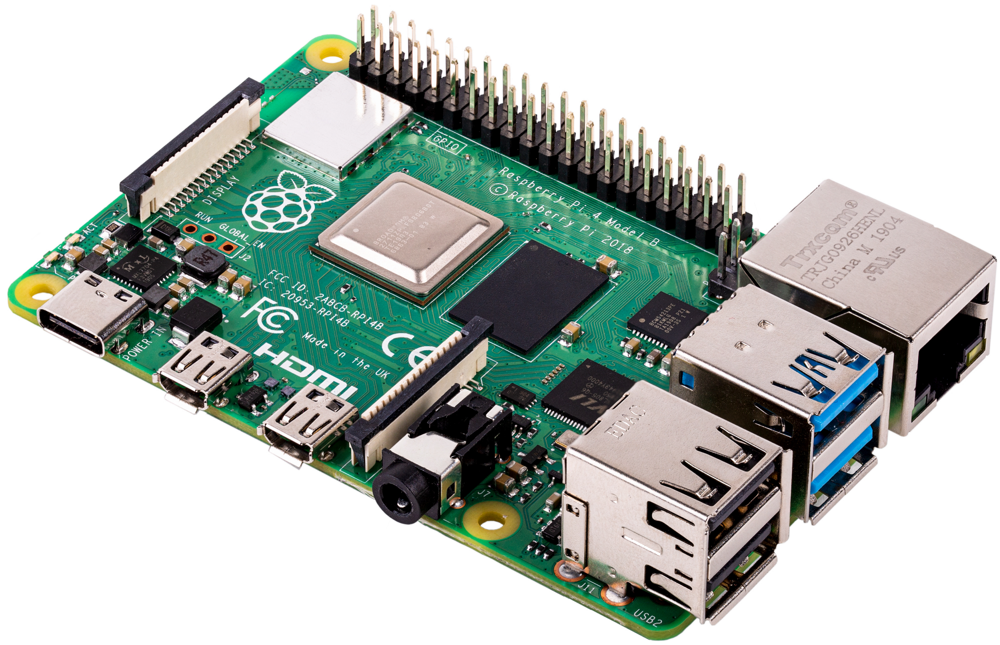

Figure 3.2: Raspberry Pi 4 Model B used in PiIrrigate

3.3 SOFTWARE COMPONENTS

3.3.1 AZURE IOT HUB

Azure IoT Hub is a cloud platform that allows for secure and reliable commmunicaiton between IoT devices and cloud services. It provides a way to connect, monitor, and manage IoT devices at scale.

I chosed to use Azure IoT Hub for this project because it provides a reliable and secure way to communicate between the Raspberry Pi and the web API. Also I am more familiar with Azure services. Other alternatives could be AWS IoT Core or Google Cloud IoT Core, but I found that Azure IoT Hub provides a more user-friendly interface and better integration with other Azure services. Yet another alternative could be to use a self-hosted MQTT broker, such as Mosquitto, but this would require more setup and maintenance effort.

3.3.2 WEB API

The web API is developed using ASP.NET Core framwork. ASP.NET Core is a cross-platform framework for building modern applications with .NET and C#. One of the main advantages of using ASP.NET Core is that it comes with all the necessary tools to build a web API, such as routing, model binding, and dependency injection.

4. BODY

4.1 FIGURES AND PHOTOGRAPHS

Figures (including images, graphs and screenshots) are numbered in order of their appearance in the paper. Alternatively, figures may be numbered in order in each chapter, including the chapter number. Each figure has a number and a title, which is mentioned under the figure, centered. Where applicable, the source of the figure shall be indicated in brackets after the title of the figure;

All figures and photographs inserted in the paper must be referenced in the text, numbered and titled.

There will be a blank line (Nimbus Sans 12 pt) between the figure and the text. Figures will be centered.

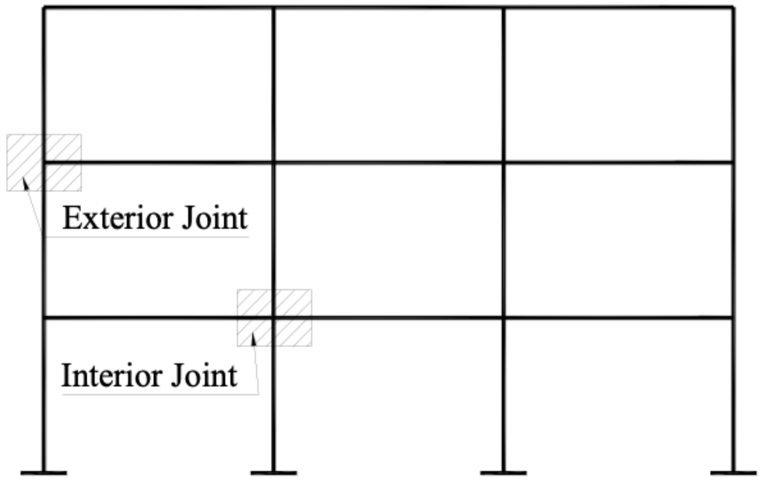

Figure 4.1: Example of a figure (source: The Scientific Bulletin of the UPT – series Building Engineering – Architecture, issue 2/2010)

A reference to a figure can be created: Figure 4.1.

4.2 TABLES

Tables will be numbered in the order in which they appear in the paper. Alternatively, tables can be numbered in order in each chapter, including the chapter number in the numbering. Each table has a number and a title, which is mentioned above the table, in a centered alignment. Where applicable, the data source shall be indicated in brackets after the title of the table.

All tables presented in the paper must be referenced in the text of the paper, must be numbered and accompanied by a title (see example below). If copied figures are used, the source of the photo will be indicated in parenthesis. As far as possible, the usual font (Nimbus Sans 12 pt) will be kept in the table, but there are also accepted ways to highlight important results (Bold, Italics, etc.)

A blank line (Nimbus Sans 12 pt) will be left between the text and the table. Tables will be centered.

Table 4.1: Example of a table

	Yield stress, fy [N/mm2]		Tensile strength, fu [N/mm2]	
Element	Mill certificate	Coupon tests	Mill certificate	Coupon tests
Beam IPE360	285.0	329.8 flange 348.4 web	427.0	463.2 flange 464.0 web
Column HEB300	311.3	313.0 flange 341.8 web	446.0	449.8 flange 464.4 web
End plate	281.0	248.3	424.7	416.0
Cover plate	296.0	273.2	443.0	436.7

A reference to a table can be created: Table 4.1. To create a table, one can use `https://www.tablesgenerator.com/latex_tables`.

4.3 FORMULAE

Formulas used in the text will be numbered in order of appearance in the paper. Alternatively, formulas can be numbered in order in each chapter, including the chapter number. The numbering of formulas is done in round brackets. A blank line (arial 12 pt) will be left between the text and the formula. Formulas will be aligned to the right.

$$A = \pi r^2 \tag{4.1}$$

A reference to an equation can be created: (4.1).

4.4 CODE

```
1 public class Client {
2     public static void main(String[] args) {
3         Animal tiger = new Tiger();
4         Animal parrot = new Parrot();
5         tiger.breed(parrot);
6     }
7 }
```

Snippet 4.1: Subtype polymorphism example
Snippet 4.1

Flexibility introduced by polymorphism can be seen in Snippet 4.1, lines *(3)*, *(4)*. Variables, `tiger` and `parrot`, being of type `Animal`, can refer to `Tiger`, or `Parrot` objects. Still, this flexibility becomes a problem when dealing with call such as that in line *(5)*, since „breeding” makes sense only between objects having the same type.

```java
public class Client {
        public static void main(String[] args) {
                Animal tiger = new Tiger();
                Animal parrot = new Parrot();
                tiger.breed(parrot);
        }
}
```

Snippet 4.2: Subtype polymorphism example Snippet 4.1

5. CONCLUSIONS

```
public class Client {
        public static void main(String[] args) {
                Animal tiger = new Tiger();
                Animal parrot = new Parrot();
                tiger.breed(parrot);
        }
}
```

Snippet 5.1: Subtype polymorphism example
Snippet 4.2

The paper will end with a chapter of conclusions. It will contain the main results of the work and their practical implications. In the case of diploma projects, the main synthetic data obtained from the design process will be mentioned.

5.1 BIBLIOGRAPHY

At the end of the paper will be given a list of references for the scientific texts consulted during the work. All sources will be listed, including those on the internet. These will be referenced in the text and listed in alphabetical order, as in the examples below.

IMPORTANT: Citations are not entered manually, but by using the biblatex library. References can be generated with various external tools in .bib format and then inserted into the project structure in bibliography.bib. Further reading:

1. Docs: `https://mirrors.nxthost.com/ctan/macros/latex/contrib/biblatex/doc/biblatex.pdf`
2. Convert DOI to .bib: `https://www.doi2bib.org/`

Bibliography should include all literature titles that have served as a basis for documentation, i.e. authors who have been quoted in the text, in all chapters of the thesis paper.

The Faculty of Automation and Computers requires the use of the IEEE citation style (details `https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf`), used primarily in scientific publications in the field of IT. The three important parts of the reference are:

a. The name of the author indicated as the first initial of the first name, then the full name.
b. The title of the article, the patent, the conference paper, etc., in quotation marks.
c. The title of the magazine or book in italics. How the reference is written depends on the type of publication, please follow the instructions at the link above carefully.

Each citation should be noted in the text using simple sequential numbers. A number in square brackets, placed in the text of the report, indicates the specific reference. Citations are numbered in the order in which they appear. Once a source has been cited, the same number is used in all subsequent references in the text. No distinction is made between electronic and printed sources, except for the details of the cited references.

Each reference number must be enclosed in square brackets on the same line as the text, before any punctuation mark, with a space before the parentheses.

Examples:

a. ". . .the end of my research [13]."

b. "The theory was first introduced in 1987 [1]."

The list of references in the bibliography is composed of all the sources used to document the paper and is made in the numerical order of the citation in the text and not in alphabetical order of the authors.

The identical insertion of a sentence or paragraph shall be made by including the page from the source used, but also by quotation marks and the use of Italics; for sources taken from the Internet, the page addresses shall be included; in the final bibliographic list the works shall be entered in the alphabetical order of the authors' names. For collective works, the rule of alphabetical order applies to the first author.

If websites, magazines or articles are quoted, three asterisks will appear before, and then information on the volume, the issue, the pages consulted, the exact website address of the article, the date of the site visit and the date of downloading, as well as the date of the accessing. Web page addresses will be entered at the end of the list.

The bibliographical sources the author of which cannot be mentioned should be specified as "***" followed by the name of the article and/or book, the publishing house and the place of appearance (for books), the volume, the issue, the first and last page of the quoted work, and the year of appearance.

*** https://ro.wikipedia.org/wiki/Motor accessed February 2022

Example: Einstein **einstein**, mentioning "The intuitive mind is a sacred gift."

5.2 AUTHENTICITY DECLARATION

The last page of the thesis paper shall contain the „Statement regarding the authenticity of the thesis paper", in handwriting, filled in according to the UPT's requirements. The Statement shall be downloaded from the web, at:

`http://www.upt.ro/img/files/Regulamente_UPT/2020/Declaratie_de_autenticitate_UPT_2020.pdf`

6. BIBLIOGRAPHY

[1] K. Obaideen et al., “An overview of smart irrigation systems using iot,” *Energy Nexus*, vol. 7, p. 100 124, 2022, ISSN: 2772-4271. DOI: https://doi.org/10.1016/j.nexus.2022.100124. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772427122000791.

[2] D. Tilman, “Global environmental impacts of agricultural expansion: The need for sustainable and efficient practices,” *Proceedings of the National Academy of Sciences*, vol. 96, no. 11, pp. 5995–6000, 1999. DOI: 10.1073/pnas.96.11.5995. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.96.11.5995. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.96.11.5995.

[3] I. Froiz-Míguez et al., “Design, implementation, and empirical validation of an iot smart irrigation system for fog computing applications based on lora and lorawan sensor nodes,” *Sensors*, vol. 20, no. 23, 2020, ISSN: 1424-8220. DOI: 10.3390/s20236865. [Online]. Available: https://www.mdpi.com/1424-8220/20/23/6865.

[4] J. Q. M. Malarie Gotcher Saleh Taghvaeian, “Smart irrigation technology: Controllers and sensors,” [Online]. Available: https://extension.okstate.edu/fact-sheets/smart-irrigation-technology-controllers-and-sensors.html.

[5] U. S. E. P. Agency, “Soil moisture-based irrigation controllers,” [Online]. Available: https://www.epa.gov/watersense/soil-moisture-based-irrigation-controllers.

[6] M. Dhanaraju, P. Chenniappan, K. Ramalingam, S. Pazhanivelan, and R. Kaliaperumal, “Smart farming: Internet of things (iot)-based sustainable agriculture,” *Agriculture*, vol. 12, no. 10, 2022, ISSN: 2077-0472. DOI: 10.3390/agriculture12101745. [Online]. Available: https://www.mdpi.com/2077-0472/12/10/1745.

[7] X. Shi et al., “State-of-the-art internet of things in protected agriculture,” *Sensors*, vol. 19, no. 8, 2019, ISSN: 1424-8220. DOI: 10.3390/s19081833. [Online]. Available: https://www.mdpi.com/1424-8220/19/8/1833.

DECLARAȚIE DE AUTENTICITATE A
LUCRĂRII DE FINALIZARE A STUDIILOR *

Subsemnatul **ION POPESCU**, legitimat cu **CI** seria **TZ** nr. **100772**, CNP **5000102355779**, autorul lucrării **TITLUL LUCRĂRII DE FINALIZARE A STUDIILOR** elaborată în vederea susținerii examenului de finalizare a studiilor de **LICENȚĂ** organizat de către Facultatea **AUTOMATICĂ ȘI CALCULATOARE** din cadrul Universității Politehnica Timişoara, sesiunea **IUNIE** a anului universitar **2022**, coordonator **dr.ing. MIHAI IONESCU**, luând în considerare conținutul art. 34 din *Regulamentul privind organizarea și desfășurarea examenelor de licență/diplomă și disertație*, aprobat prin HS nr. 109/14.05.2020 și cunoscând faptul că în cazul constatării ulterioare a unor declarații false, voi suporta sancțiunea administrativă prevăzută de art. 146 din Legea nr. 1/2011 – legea educației naționale și anume anularea diplomei de studii, declar pe proprie răspundere, că:

- această lucrare este rezultatul propriei activități intelectuale,
- lucrarea nu conține texte, date sau elemente de grafică din alte lucrări sau din alte surse fără ca acestea să nu fie citate, inclusiv situația în care sursa o reprezintă o altă lucrare/alte lucrări ale subsemnatului.
- sursele bibliografice au fost folosite cu respectarea legislației române şi a convențiilor internaționale privind drepturile de autor.
- această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență/diplomă/disertație.

Timișoara,

Data   Semnătura

____________________   ____________________

*Declarația se completează „de mână” şi se inserează în lucrarea de finalizare a studiilor, la sfârşitul acesteia, ca parte integrantă.