

ADAPTIVE WIRELESS SENSOR NETWORK DESIGN FOR ULTRA LOW
POWER IOT APPLICATIONS

A THESIS SUBMITTED TO
THE INSTITUTE OF GRADUATE PROGRAMS
OF
İSTANBUL BİLGİ UNIVERSITY



KARDELEN OLCAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

2024

ADAPTIVE WIRELESS SENSOR NETWORK DESIGN FOR ULTRA LOW POWER IOT APPLICATIONS

Kardelen Olcay
121815003

Assoc. Prof. Dr. Baykal SARIOĞLU
İstanbul Bilgi University

Assoc. Prof. Dr. Yiğit Dağhan GÖKDEL
İstanbul Bilgi University

Assist. Prof. Dr. Murat TÜMER
Türk-Alman University

Date of Approval: 6/25/2024



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Kardelen Olcay

Signature:

ABSTRACT

Internet of Things (IoT) is significantly influencing a conversion of multiple industries, with Wireless Sensor Networks (WSNs) emerging as the primary catalysts of new technology. In this thesis investigates the use of a diverse network model to enhance the placement of sensors in agricultural environments. The main goal is to strategically place sensor nodes in order to optimize energy usage, extend the lifespan of the network, and ensure reliable data transfer. The suggested technique includes an offline model for positioning sensor nodes in the specified area, considering the coverage needs and network connections. The aim in this study is to design a two-stage centralized control mechanism that organizes the sensor node into protective boxes, integrating the decision-making process. This grouping enables the efficient exploitation of common resources, such as batteries and bandwidth, while limiting the number of boxes required for cost-effectiveness. This study initially investigates connection and coverage issues utilizing an offline model. Afterwards, the online model is implemented. The idea here has been created for handling real-time issues. The focus is on optimizing energy economy by consolidating sensors within boxes, reducing data transmission hops, and taking into account energy consumption during sensing, sending, and active/sleep modes. This study offers both theoretical insights and practical applications, demonstrating the potential of WSNs to revolutionize sustainable agriculture in the context of smart agricultural techniques.

Keywords: Internet of Things; Wireless Sensor Network; Energy Consumption; Smart Agriculture; Two Stage Mathematical Model

ÖZ

Internet of Things (IoT), birçok endüstrinin dönüşümünü önemli ölçüde etkilerken, kablosuz sensör ağları (WSN) bu yeniliklerin önde gelen katalizörleri olarak ortaya çıkıyor. Bu çalışma, tarım ortamlarında algılayıcıların yerleştirilmesini artırmak için farklı bir ağ modelinin kullanıldığını araştırıyor. Ana hedef, enerji kullanımını optimize etmek, ağın ömrünü uzatmak ve güvenilir veri aktarımı sağlamak için algılayıcı düğümünü stratejik olarak yerleştirmektir. Önerilen teknik, hedef alanında algılayıcı düğümleri konumlandırmak için, kapsam gereksinimlerini ve ağ bağlantılarını dikkate alarak çevrimdışı bir model içerir. Bu çalışmada amaç, algılayıcı düğümü koruyucu kutulara düzenleyerek karar verme sürecini birleştiren iki aşamalı bir merkezli kontrol mekanizması haline getirmektir. Bu gruplandırma, batarya ve bant genişliği gibi ortak kaynakların verimli bir şekilde kullanılmasını sağlarken, maliyet verimliliği için gerekli kutuların sayısını sınırlar. Bu tez, başlangıç aşamasında çevrimdışı bir dağıtım modeli aracılığıyla bağlantı ve kapsama sorunlarını ele alarak ve sonraki aşamada çevrimiçi bir enerji optimizasyon modeli yoluyla gerçek zamanlı uyum endişeleri ile ilgilenerek önemli katkılar sağlar. Bu çalışma hem teorik anlayışlar hem de pratik uygulamalar sunarak, akıllı tarım teknikleri bağlamında sürdürülebilir tarımda devrim yaratmak için WSN potansiyelini gösteriyor.

Anahtar Kelimeler: Nesnelerin İnterneti; Kablosuz Sensör Ağı; Enerji Tüketimi; Akıllı Tarım; İki Aşamalı Matematiksel Modelleme



Dedicated to my grandmother,

ACKNOWLEDGMENTS

I would like to say appreciation to my supervisor, Doc.Dr.Baykal Sarioğlu,for his supportive encouragement in all aspects of my life, including my academic pursuits and extracurricular activities. His guidance played a crucial role in enabling me to successfully complete my thesis.

I express my heartfelt thanks to Doc.Dr.Yiğit Dağhan Gökdel for providing guidance and support during the process of my thesis, academic journey, and this research effort. I would like to express my gratitude to Banu Kabakulak for her great contribution and work in mathematical modeling. I express my sincere gratitude to my colleague, Melike Akmandor, very much for her work on this thesis. I'm grateful for your support in all your difficulties. The support of İpek Hilal Kacar, whom I see as the rest of my family, is undeniable. I feel extremely fortunate to have them in my life.

Finally, thank you for your eternal contributions to my dear family. I'll always have my place, my dear grandmother, too.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ.....	v
ACKNOWLEDGMENTS	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	x
LIST OF FIGURES.....	xi
LIST OF SYMBOLS.....	xii
LIST OF ABBREVIATIONS	xiv
1 INTRODUCTION	1
2 SYSTEM DESCRIPTION	8
3 MATHEMATICAL MODELLING	13
3.1. Offline Deployment Model	14
3.2. Online Energy Optimization Model	16
4 HARDWARE DESIGN	19
4.1. Gateway Design.....	19
4.2. Sensor Node Design	22
4.2.1. First Sensor Node Design	22
4.2.2. Second Sensor Node Design	24
5 HARDWARE IMPLEMENTATION	27
5.1. Gateway Implementation.....	27
5.2. Sensor Nodes Implementation.....	30
6 EXPERIMENTAL RESULTS AND TESTS.....	34

7 CONCLUSION	45
REFERENCES	47
Appendix A. First Sensor Node Arduino Code	52
Appendix B. Gateway Python Code	56



LIST OF TABLES

Table 1.1. The Comparative Study of Short-Range Wireless Technologies.....	3
Table 1.2. The Comparative Study of Long-Range Wireless Technologies	5
Table 6.1. The Average Current Utilization of the Implemented Hardware.....	37
Table 6.2. Bill of Materials.....	37
Table 6.3. Comparison of LoRa and ESP8266.....	43



LIST OF FIGURES

Figure 2.1. Embedded System Network Schematic	9
Figure 2.2. Embedded System Block Diagram	10
Figure 2.3. LoRa Technology Channels Uplink and Downlink Information for Europe, USA, and Australia.....	12
Figure 4.1. Block Diagram of the Gateway Design	21
Figure 4.2. Block Diagram of the Hardware System Set Up	23
Figure 4.3. Flowchart of the First Hardware System Set Up	24
Figure 4.4. Block Diagram of the Second Hardware System Set Up.....	26
Figure 4.5. Flowchart of the Second Hardware System Set Up.....	26
Figure 5.1. View of the Implemented Gateway in a Waterproof Box.....	27
Figure 5.2. The Flowchart of Gateway Arduino Code	28
Figure 5.3. The Flowchart of Gateway Python Code	29
Figure 5.4. View of the Implemented Sensor Node in Area	30
Figure 5.5. The Algorithm of the Implemented Hardware.....	31
Figure 5.6. The Example Output Code.....	32
Figure 6.1. The Sensor Node General Scheme.....	35
Figure 6.2. The Parking Sensor Node General Scheme	35
Figure 6.3. The Gateway General Overview	36
Figure 6.4. Overview of Sensor Node in the Parking IoT.....	36
Figure 6.5. The Measurement of the Power Sources Consumed During Operation	38
Figure 6.6. The Measurement of the Power Sources Consumed During Operation	40
Figure 6.7. External View of the Sensor Node.....	41
Figure 6.8. Humidity Data with Dates and Times.....	42
Figure 6.9. Temperature Data with Dates and Times.....	42
Figure 6.10. Distance Measurement Map.....	44

LIST OF SYMBOLS

N :	set of candidate locations
K_1 :	set of sensor types
K_2 :	set of node types
β_k :	bandwidth of the type k sensor
c_k :	price of the type k sensor
r_k^c :	transmission area radius of a type k sensor
r_k^s :	sensing radius of a type k sensor
$a_{jkj'}$:	1 if point j' is within the sensing radius of a sensor (j, k) , 0 otherwise
$b_{jkj'}$:	1 if point j' is within the communication radius of a sensor (j, k) , 0 otherwise
h_{jk} :	the size of the data packet collected by a sensor (j, k) during an active period
e_k^c :	energy utilization for a type k sensor to forward a single unit of data
e_k^{ar} :	energy utilization for a type k sensor to achieve a single unit of data in duration of activity
e_k^{sr} :	energy utilization for a type k sensor in an inactive period
e_k^a :	energy utilization for a type k sensor in duration of activity
e_k^s :	energy utilization for a type k sensor for detecting and analyzing data during an active phase
E_j :	first battery energy of a node at location j
$\bar{c}_j$:	price of implementing a node at location j
B :	existing financial resource
$\bar{f}_{ik}$:	coverage requirement of a point i by a type k sensor without redundancy

f_{ik} :	coverage requirement of a point i by a type k sensor with redundancy
α :	minumum number of nodes to communicate with
x_{jk} :	1 if a sensor (j, k) is deployed, 0 otherwise
$\bar{x}_j$:	1 if a gateway is deployed at point j , 0 otherwise
$\bar{y}_j$:	1 if there is a node at point j , 0 otherwise
z_{jkt} :	1 if a sensor (j, k) is active in period t , 0 otherwise
y_{iljkt} :	data transmission rate from sensor (i, l) to sensor (j, k) during period t
E_{jkt} :	energy utilization of a node (j, k) during period t
L :	network duration of existence
t :	period time
I_{avg} :	average current
I_{sleep} :	sleep mode current
I_{idle} :	wake up mode current
$I_{transmission}$:	transmission current
$I_{receive}$:	receive current
T_{total} :	total time for each half hour
T_{sleep} :	sleep time for each half hour
T_{idle} :	wake up time for each half hour
$T_{transmission}$:	transmission time for each half hour
$T_{receive}$:	receive time for each half hour
E_j :	energy usage in joules

LIST OF ABBREVIATIONS

RF:	Radio Frequency
BLE:	Bluetooth Low Energy
IoT:	Internet of Things
LAN:	Local Area Network
PAN:	Personal Area Network
RFID:	Radio Frequency Identification
NB-IoT:	Narrowband IoT
LPWAN:	Low Power Wide Area Network
LTE:	Long Term Evolution
GSM:	Global System for Mobile Communications
LoRa	Long Range
WSN:	Wireless Sensor Network
OND:	Offline Network Deployment
OEO:	Online Energy Optimization
UART:	Universal Asynchronous Receiver Transmitter
EEPROM:	Electrically Erasable Programmable Read Only Memory
WSN:	Wireless Sensor Network
ISM:	Industrial, Scientific and Medical
CSS:	Chirp Spread Spectrum
PHY:	Physical Layer

1 INTRODUCTION

The productivity of agricultural applications has significantly decreased due to changing climatic conditions and the rising global temperature [1]. This decline is further exacerbated by factors such as the reduction in the agricultural labor force, the rise in the cost of agricultural equipment, and the inability to control soil quality [2]. To mitigate these issues, agricultural resources must be utilized judiciously. Sensors can be a preferred solution for measuring real-time changes in agricultural land, soil, and plants, keeping the farmer informed of ongoing developments. Sensors are energy-limited, cost-effective devices capable of detecting and processing data, and they communicate using Radio Frequency (RF) [3]. The utilization of internet-connected sensors plays a crucial role in enhancing soil and plant productivity while concurrently reducing costs.

Agricultural products necessitate applications such as monitoring soil moisture status and overseeing environmental conditions, including temperature and humidity. It is evident that analyzing this data through smart agriculture applications is essential for cultivating productive plants and sustaining agriculture in the future. Furthermore, it emphasizes the importance of analyzing large datasets from agricultural areas, a fundamental aspect of smart farming applications. These applications are essential in cultivating robust, healthy crops, enhancing resilience against natural disasters, and sustaining agricultural operations across diverse landscapes. In various applications, sensors and actuators have the potential to significantly reduce water usage in agriculture [4]. Therefore, it can be deduced that maintaining an area with internet-connected sensors and other devices is crucial for efficiently utilizing resources, increasing product throughput, and reducing production costs.

Smart agriculture involves low-power communication protocols that are categorized into short and long distances, depending on the communication range. These protocols facilitate enhanced communication and decision-making processes for farmers. The wireless protocols that are most commonly favored in the domain of smart agriculture are as follows: IEEE 802.11 WIFI, 2G/3G/4G-Mobile Communications Standards,

LoRaWAN, WiMax, Low Rate Wireless Personal Area Networks, Bluetooth, RFID, and ZigBee [5].

Bluetooth is a widely used wireless technology that operates over short distances and is specifically developed for personal area networks (PANs). Operating at a frequency of 2.4 GHz, it provides data speeds of up to 3 Mbps for regular Bluetooth and up to 24 Mbps for Bluetooth Low Energy (BLE). The average distance it can cover is approximately 10 meters, which makes it well-suited for establishing connections between personal devices like headphones, keyboards, mouse, and various Internet of Things (IoT) gadgets. Bluetooth is renowned for its exceptional energy efficiency, particularly in BLE mode, making it indispensable for devices powered by batteries.

Zigbee is a wireless technology that operates at low power and low data rates. It is mostly utilized for home automation and industrial applications. The device functions on the frequencies of 2.4 GHz, 900 MHz, and 868 MHz, and can achieve data rates of up to 250 kbps. Zigbee enables mesh networking, facilitating the transmission of data between devices by utilizing intermediate nodes. This feature enhances the overall coverage and dependability of the network. This feature makes it well-suited for applications such as intelligent lighting, residential security, and industrial monitoring, where power efficiency and network reliability are of utmost importance.

Z-Wave is a low-power wireless technology that specifically targets home automation. The device operates at a frequency of 908.42 MHz in the US and 868.42 MHz in Europe. It has a maximum range of 100 meters and can transmit data at rates of up to 100 kbps. Z-Wave is specifically engineered to facilitate dependable and efficient communication between smart home devices, including but not limited to lighting, locks, and thermostats. The utilization of a sub-1 GHz frequency aids in minimizing interference caused by other wireless technologies, hence establishing a dependable option for smart home networks.

Wi-Fi is a fast wireless technology that is commonly used for local area networks (LANs) and accessing the internet. By operating on both 2.4 GHz and 5 GHz bands, this device can achieve data rates of up to 3.5 Gbps using Wi-Fi 5, and even greater speeds with Wi-Fi 6. The device has a maximum indoor range of 100 meters, which makes it appropriate for use in residential areas, workplaces, and public Wi-Fi zones. Although Wi-Fi provides

broad coverage and fast data rates, it generally consumes more power than other short-range technologies and can experience interference in congested networks.

Radio Frequency Identification (RFID) operates by assigning a distinct number to each object in order to record information. RFID is a system that includes readers, a host, and tags. The tags are capable of receiving and transmitting radio waves. Tags possess a distinct identification number and environmental data, and these tags are transferred by a chip to a reader in order to identify an object. RFID technology has been widely employed in several agricultural applications, including livestock management systems and supply chain management systems.

Short-range wireless technologies are compared by standard, frequency, transmission range, data rate, and application types. These differences are shown in the Table 1.1 below.

Table 1.1. The Comparative Study of Short-Range Wireless Technologies

Parameters	BLE/Bluetooth	Zigbee	Z-Wave	Wi-Fi	RFID
Standard	IEEE 802.15.1	IEEE 802.15.4	Proprietary	IEEE 802.11	ISO 18000-6C
Frequency	2.4 GHz	2.4 GHz	900 MHz	5-60 GHz	860-960 MHz
Transmission Range(meters)	8-10 m	10-20 m	30 m	20-100 m	1-5 m
Data Rate	1-24 Mb/s	20-250 kb/s	9.6,40,100 kb/s	1 Mb/s - 7 Gb/s	40 to 160 kb/s

Long Range(LoRa) is a wireless technology specifically designed for IoT, providing long-distance communication capabilities of up to 15 kilometers in rural regions. The

device functions within different frequency ranges below 1 GHz, specifically 868 MHz in Europe and 915 MHz in North America, and can transmit data at speeds of up to 50 kbps. The three most often utilized frequency bands are 433 MHz for Asia, 868 MHz for Europe, and 915 MHz for North America [6]. The low power consumption of LoRa makes it well-suited for battery-powered IoT devices utilized in smart cities, agriculture, and environmental monitoring. LoRa is particularly suitable for applications that necessitate sporadic transmission of compact data packets across extensive distances, despite its comparatively low data speeds.

Sigfox is a long-range wireless technology that focuses on IoT. It has coverage of up to 50 kilometers in rural locations. The device functions at frequencies below 1 gigahertz and provides data transfer speeds of up to 100 bits per second. Sigfox is specifically engineered to have extremely low power consumption, which makes it ideal for use in applications such as asset tracking and smart metering. These applications require devices to operate for long periods of time without the need for regular battery replacements. Sigfox is a reliable option for low-bandwidth IoT applications due to its long range and low power consumption, despite its limitations in data rates and message frequency.

Narrowband IoT (NB-IoT) is a technology designed to facilitate the connectivity of various IoT devices and services. It operates on a low-power wide-area network (LPWAN) to ensure efficient power usage and extensive coverage. NB-IoT operates within the established Long Term Evolution (LTE) and Global System for Mobile Communications (GSM) frequency bands and employs narrowband technology, often with a bandwidth of 200 kHz. This enables reliable and long-range communication. It is specifically engineered to effectively support devices that necessitate extended battery life and dependable, extensive coverage, rendering it well-suited for implementations like smart and automated cities.

NB-IoT provides numerous benefits, such as little energy usage, enabling devices to function for a duration of up to 10 years using only one battery. Furthermore, it possesses the potential to accommodate a substantial number of devices within a confined cellular space, hence facilitating a high connection density. The technology offers robust coverage, especially in difficult circumstances such as subterranean locations or under

buildings, thanks to its low-frequency operation and improved link budget. In addition, NB-IoT takes use of its inclusion in the LTE family, allowing it to utilize the already established cellular infrastructure. This results in lower deployment expenses and guarantees smooth integration with mobile networks.

Cellular networks, which include advanced technologies such as 4G (LTE) and 5G, play a vital role in facilitating contemporary mobile communication. These networks function within a frequency range of 600 MHz to 5 GHz for 4G, and can reach up to 40 GHz for 5G. These networks offer broad coverage, usually reaching several kilometers from each cell tower, and enable fast data speeds, reaching up to 1 Gbps for 4G and up to 10 Gbps for 5G. Cellular networks are well-suited for mobile phone users and applications that need continuous connectivity due to their ability to handle high mobility.

The comparison of long-range technologies is presented in Table 1.2, focusing on key factors such as frequency, range, and battery life.

Table 1.2. The Comparative Study of Long-Range Wireless Technologies

Parameters	LoRa	Sigfox	NB-IoT	5G
Standard	LoRa WAN	Sigfox	3GPP	3GPP
Frequency	ISM Bands 433,868,915 MHz	ISM Bands 433,868,915 MHz	Licenced LTE	24-100 GHz
Transmission Range (km)	5 km (urban) 20 km (rural)	10 km (urban) 40 km (rural)	1 km (urban) 10 km (rural)	1000 feet
Data Rate	50 kbhs	100 kbhs	200 kbhs	20 Gb/s
Battery Life	10 years	10 years	15 years	10 years

The LoRa module provides substantial benefits for smart IoT agriculture, rendering it well-suited for extensive implementations. The device has the capability to send data across distances ranging from 10 to 15 kilometers, which guarantees extensive coverage of large agricultural fields and enables efficient communication with sensors located in remote areas. The low power consumption of LoRa devices enables them to function over extended periods using small batteries, hence minimizing the requirement for frequent maintenance and battery changes. This advantage is particularly valuable in isolated locations. In addition, LoRa technology is economically efficient, since it offers reasonably priced modules and gateways. Furthermore, it has the capability to accommodate a large number of devices on a single network, thereby making it extremely scalable. LoRa operates in the sub-GHz frequency spectrum to ensure dependable communication with minimal disruption. Moreover, LoRaWAN, being an open standard, promotes innovation and a wide range of compatible devices, hence improving the versatility and adjustability of smart agriculture solutions. LoRa has a range of advantages that make it a superior option when compared to other IoT technologies such as NB-IoT, Sigfox, Wi-Fi, and Zigbee ([7],[8],[9],[10]). These benefits include longer ranges, lower power consumption, and greater cost-effectiveness and scalability for broad agricultural applications.

Numerous literature reviews and academic research have investigated various sub-problems relevant to the main focus of the research. One of the significant concerns among these studies pertains to the implication of sensor placement on energy consumption during data transmission [11]. As an example of this concern, this issue is defined as a non-concave Boolean optimization problem. The proposed method to solve this is by relaxing both the 's' and Boolean constraints to optimize sensor deployment [11]. In WSNs, effective energy usage represents a crucial concern. Various academic research offers diverse methods for overcoming this problem. An alternative methodology involves the strategic positioning of sensors to cover the desired area, thus aiming to minimize the number of sensors used [12]. An alternative strategy focuses on enhancing inter-sensor communication to minimize the volume of data handled by individual sensors [13].

Different technologies are used for embedded system and hardware design in the literature. The short range and long range wireless technologies mentioned above are located in various areas. XBee protocol is used for communication of sensor nodes designed for agriculture [14]. It has a frequency of 2.4 GHz in the ISM band. Electronically Erasable Programmable Read Only Memory (EEPROM) memory is preferred to prevent data loss by using Atmega8L as the microcontroller. In another study, Zigbee technology, which is in the IEEE 802.15.4 standard, was preferred for low power consumption and healthy data communication. The EMP-ZBR protocol is recommended, which covers issues about energy consumption. This protocol has superior performance compared to Zigbee protocols [15]. Another short range wireless communication technology, Bluetooth, is preferred and the aim is for the sensors to be self-powered [16]. Therefore, energy is collected from the power of ambient light by using BLE as the communication protocol. This energy is stored for sensors that measure data such as humidity and temperature. However, this application can work efficiently in small environments such as greenhouses. Its performance may not be sufficient in areas requiring long distances. In another research, RFID technology is utilized to collect data on soil moisture and temperature, with the objective of enhancing efficiency and minimizing unnecessary usage [17].

In addition, as an example of long range wireless technology, a sensor node is developed using Sigfox technology [18]. The general purpose of this study is to calculate battery life by analyzing energy consumption. In another different study, Sigfox technology is preferred [19]. The ESP32 module is used as a microcontroller. The general purpose is to measure the performance of Sigfox technology in regions where there is no infrastructure. In this measurement, temperature and humidity data are taken into account. NB-IoT technology is suitable for low power consumption and wide distance data transmission. It provides real-time transmission of soil moisture and temperature data. That's why NB-IoT has been preferred in various studies in the field of smart agriculture ([20],[21],[22],[23])

2. SYSTEM DESCRIPTION

Wireless Sensor Networks (WSNs) provide sensors to communicate with each other and send the obtained data to a gateway. These sensors facilitate data transmission and are commonly employed in conjunction with batteries, CPUs, memory, and radio functionality [24]. WSNs can be categorized into two main types: homogeneous and heterogeneous. A homogeneous sensor network consists of sensors that are all of a similar type, unlike a heterogeneous sensor network which includes sensors of different types. The primary objective of heterogeneous sensor networks is to collect data pertaining to various factors within a given geographical area. The role of heterogeneous WSNs is essential for Internet of Things (IoT) technology [25]. Data transfer between sensors in IoT applications is facilitated by WSN technology. WSNs are utilized for a wide range of objectives, including security monitoring of smart residences and devices, detection of forest fires, assistance with healthcare applications, implementation of smart farming techniques, assessment of traffic density in urban areas, management of conflict zones, and assurance of military border security [26].

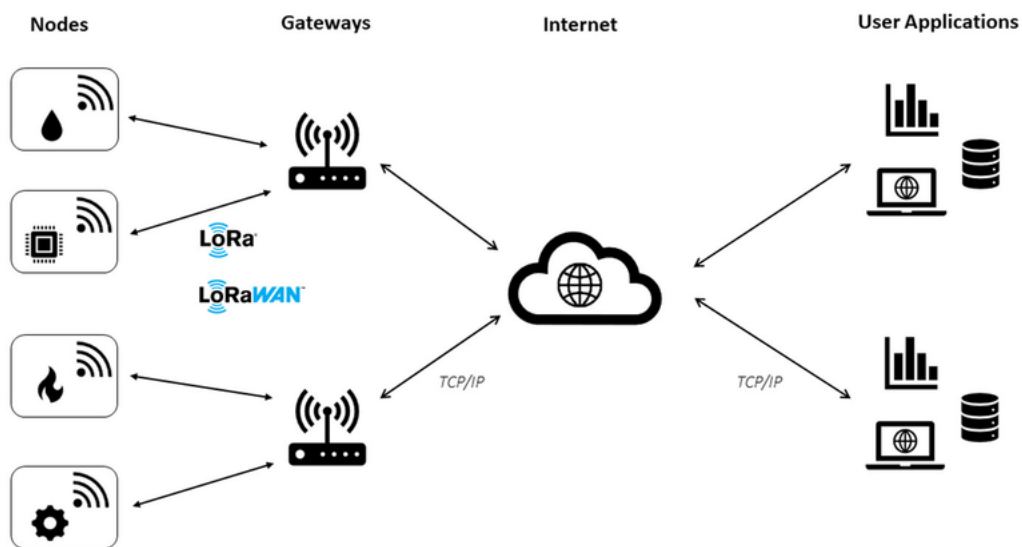
The application of WSNs in agriculture is also crucial. In the world of agriculture, it is employed to enhance productivity and alleviate the labor-intensive workload of farmers. Sensor networks are used for the purpose of monitoring animal movements within the agricultural and cattle industries, as well as enhancing crop productivity through the measurement of ambient humidity and temperature [27]. IoT is currently being utilized in several industries such as manufacturing, security, smart homes, healthcare facilities, urban areas, and agriculture. Monitoring and management play a vital role in the field of agriculture. Data is obtained through the utilization of various sensors inside WSNs. Cloud applications implement remote monitoring to examine this data. The area of remote surveillance encompasses various domains, including soil, air, heat, humidity, water, and illness.

This study [28] centers on the three protocol layers of a WSN, specifically the physical, data link, and network layers. In Section 3, it introduces a central mathematical concept

for designing an optimal triple cross-layer protocol. The procedure described in Section IV is subsequently executed on a low-power hardware design. The overall scheme of the embedded system proposed in this thesis is depicted in Figure 2.1. The embedded system developed within the scope of the project is designed to be configurable for connecting to various sensors and actuators.

In this project, the objective is to employ WSN for monitoring changes in agricultural land, such as air temperature, wind, soil moisture, etc., and to provide instant notifications to farmers. The sensor network envisioned for the project comprises the following elements, illustrated in Figure 2.2. : sensor, gateway,user application.

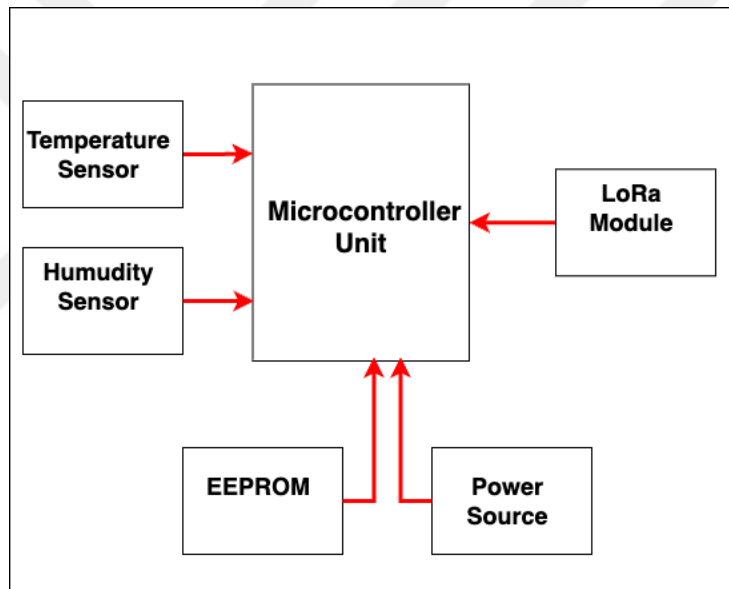
Figure 2.1. Embedded System Network Schematic



(Source: M. Ragnoli, G. Barile, A. Leoni, G. Ferri, and V. Stornelli, "An Autonomous Low-Power LoRa-Based Flood-Monitoring System," *Journal of Low Power Electronics and Applications*, vol. 10, no. 2, p. 15, May 2020, doi: 10.3390/jlpea10020015. [Online]. Available: <https://doi.org/10.3390/jlpea10020015>. [Accessed: 24-Jul-2024].)

In this study, we concentrate on smart agriculture with a focus on five fundamental aspects: (1) strategically placing network components to enable sensors to periodically monitor the field, (2) directing data from the network to the cloud within the constraints of battery energy, memory, storage, and bandwidth availability, (3) maintaining multiple copies of data within the network to prevent data loss, (4) gateways providing feedback to sensors and issuing commands to actuators based on evaluated data, and (5) enabling network components to execute commands received from gateways.

Figure 2.2. Embedded System Block Diagram



Adverse weather conditions can lead to disruptions in wireless transmission and subsequent loss of data, which is crucial for remote monitoring and essential responses. In order to mitigate the risk of data loss, EEPROM circuits have been integrated into the embedded system. This will guarantee that a duplicate of the data is saved after it has been successfully transmitted.

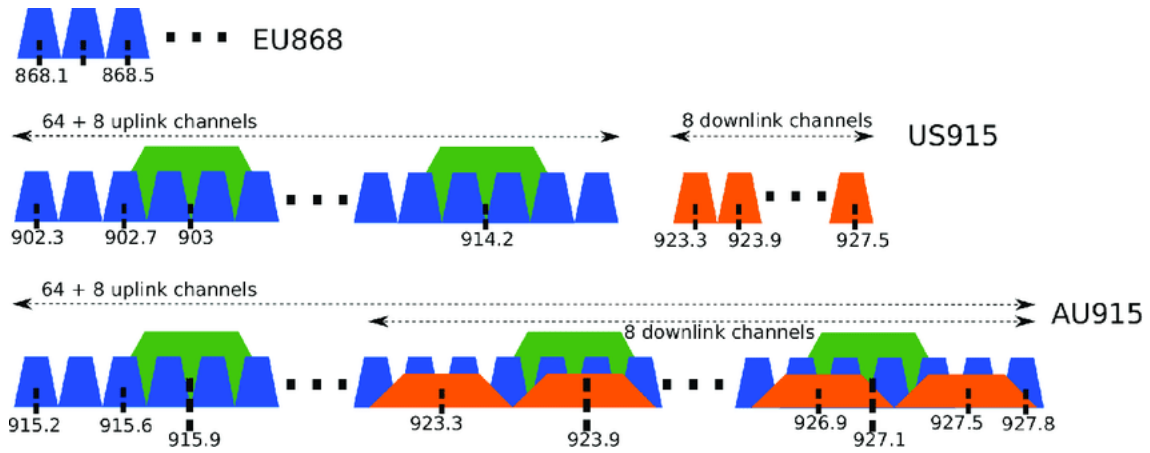
Figure 2.2. demonstrates that the embedded system requires multiple sensor units to be linked. These sensors are capable of detecting variations in communication protocols such

as SPI, I2C, UART, etc., as well as different types of signals such as analog and numerical. Hence, the embedded system will be developed to have the capability of being configured remotely, allowing the activation of the suitable configuration for the sensor integrated into the system.

In this study, LoRa technology is preferred. LoRa is a Wireless Communication Technology specifically designed for low-power smart applications. Its minimal power consumption and long-range data transmission capabilities make it very desirable in the IoT industry. LoRa technology operates inside the Industrial, Scientific and Medical (ISM) frequency ranges. ISM bands are the preferred unlicensed frequency bands for wireless communications. In LoRa technology, different ISM bandwidths are used for each region. The frequency bands that are most frequently utilized include 433 MHz, 868 MHz (in Europe), and 915 MHz (in North America). Each band has different transmission power and channel frequencies. Lower frequencies, such as 433 MHz, offer superior signal penetration across obstructions when transmitting data. Higher frequencies, such as 868/915 MHz, are better suited for transmitting data at faster rates. LoRa includes 3 types of channel bandwidth: 125 KHz, 250 KHz, and 500 KHz. While these 3 types of bandwidth are used for 868 MHz and 915 MHz frequency bands, 7.8-62.5 KHz width is used for 433 MHz [29]. LoRa utilizes designated sub-bands within ISM frequency channels for both uplink (device to gateway) and downlink (gateway to device) communications [30]. Uplink sub-bands facilitate the transmission of data from end devices to gateways. Various geographical areas are assigned distinct frequency ranges for the purpose of uplink communication. The 915 MHz ISM band consists of 64 uplink channels, each with a bandwidth of 125 kHz. These channels, numbered 0 to 63, start at a frequency of 902.3 MHz and increase by 200 kHz up to 914.9 MHz. There are a total of 8 more uplink channels, each with a bandwidth of 500 kHz. These channels, numbered 64 to 71, start at a frequency of 903 MHz and increase by 1.6 MHz until reaching 914.2 MHz. The connection between the gateway and the end-node device is facilitated through 8 downlink channels, each with a bandwidth of 500 kHz. These channels, numbered 0 to 7, operate within the frequency range of 923.3 MHz to 927.5 MHz, with an increment of 600 kHz between each channel. Figure 2.3. illustrates the channel layout of Europe, the

United States, and Australia. The 125 kHz bandwidth is utilized in the United States and Australia areas. Furthermore, it is preferable to have a 64 uplink channel design divided into two areas with increments of 200 kHz. Both sections have a total of eight channels, each with a bandwidth of 500 kHz.

Figure 2.3. LoRa Technology Channels Uplink and Downlink Information for Europe, USA, and Australia



(Source: J. R. Cotrim and J. H. Kleinschmidt, "LoRaWAN Mesh Networks: A Review and Classification of Multihop Communication," *Sensors*, vol. 20, no. 15, p. 4273, Jul. 2020, doi: 10.3390/s20154273. Available: <https://doi.org/10.3390/s20154273>.)

Chirp Spread Spectrum (CSS) method is utilized for the physical layer (PHY) of LoRa. The extensive frequency band property of CSS enables the transmission of data across great distances. This method enhances the security of LoRa by employing a non-uniform distribution of messages [32]. LoRa is the optimal technology for implementation in the IoT domain because of its efficient energy usage and ability to transmit data across vast distances [7]. Short-range wireless technologies such as Wi-Fi and Zigbee are inadequate for long-distance data transmission and low power consumption. That's why LoRa provides maximum efficiency in the IoT area [32].

3 MATHEMATICAL MODELLING

The mathematical modeling designed for the WSN system as defined in Section 2 is given in this section. The WSN optimization problem is divided into two parts: the offline network deployment (OND) model explained in Section 3 and the online energy optimization (OEO) model described in Section 4. In the research [33], offline mathematical modeling is preferred. Thanks to this model, the route to transmit the sensor data to the gateway is calculated. In cases of sensor node failure or battery depletion, the data flow route is automatically calculated. The research [28], using a heterogeneous network model, sensor nodes are grouped. With offline mathematical modeling, sensor nodes are placed optimally in the area. Minimum energy consumption is also targeted with online mathematical modeling. Accordingly, data transmission routes from the sensor node to the gateway are calculated. This thesis is developed based on these two studies.

The OND model is used by network agents such as sensors, routers, and gateways to monitor target areas, monitoring sensor nodes to determine if they are sufficient. The goal here is to minimize the cost. The network duration of existence L of a WSN is determined by the number of periods it works, and this is heavily influenced by the limited battery energy of the devices. To optimize the duration of network operation, the OND system utilizes additional sensor nodes in a WSN.

In addition, the *OEO* model is derived from the agent placements specified in the *OND* model. The objective is to ascertain the sleep/active times of the sensors that consume the least amount of energy. As an online model, *OEO* also identifies the most energy-efficient paths for transmitting data from sensors to gateways at each time period t . The network duration of existence can be established by identifying the initial phase, denoted as L , at which the *OEO* model is unable to produce a viable solution. This indicates that certain sensor nodes have depleted their battery power.

3.1. Offline Deployment Model

The OND model's objective is to strategically deploy a variety of sensors, routers, and gateways capable of detecting the entirety of the designated area. A low-cost network is established using the devices that are utilized.

An actuator is a stationary device utilized for the regulation of electrical components, such as valves, pumps, motors, and switches. It is commonly favored in fields such as irrigation and greenhouse illumination. As the device is stationary, its position in the field is already known, and hence it is not included in the mathematical modeling. The binary decision variable x_{jk} denotes the presence of either a sensor or a router $k \in \mathbf{K}_1 \cup \{r\}$ at location j . Similarly, the binary variable $\bar{x}_j$ denotes the decision on the distribution of paths at point j . When a button is pressed at location j , the binary variable $\bar{y}_j$ toggles between one and zero.

The OND objective function (3.1) mentioned below aims to minimize the overall cost of sensors, routers, and gateway devices. The c_k argument represents a price associated with the element of type k . Equipment must be covered from exterior environmental elements, such as precipitation, extreme temperatures, and snowfall. The sensor nodes are positioned within a protective enclosure to shield them from these elements. $\bar{c}_j$ denotes the expense associated with the fixed box. Employed to reduce the quantity of sensor nodes utilized in the $\bar{y}_j$ network.

$$\begin{aligned} \min \quad & \sum_{j \in N, k \in \mathbf{K}_1 \cup \{r\}} c_k x_{jk} + \sum_{j \in N} c_g \bar{x}_j + \sum_{j \in N} \bar{c}_j \bar{y}_j \\ \text{s.t} \quad & \end{aligned} \tag{3.1}$$

$$\sum_{j \in N, k \in \mathbf{K}_1 \cup \{r\}} c_k x_{jk} + \sum_{j \in N} c_g \bar{x}_j + \sum_{j \in N} \bar{c}_j \bar{y}_j \leq B \tag{3.2}$$

$$\sum_{j \in N} a_{jki} x_{jk} \geq \bar{f}_{ik} \quad i \in N, k \in K_1 \quad (3.3)$$

$$\sum_{j \in N} b_{jgi} \bar{x}_j \geq 1 \quad i \in N \quad (3.4)$$

$$\sum_{j \in N, k \in K_1 \cup \{r\}, (j,k) \neq (i,l)} b_{ilj} x_{jk} \geq \alpha x_{il} \quad i \in N, l \in K_1 \cup \{r\} \quad (3.5)$$

$$x_{jk}, \bar{x}_j, \bar{y}_j \in \{0,1\} \quad j \in N, k \in K_1 \cup \{r\} \quad (3.6)$$

The term B in function (3.2) represents the entirety of the budget. The numerical parameter a_{jki} denotes the sensing capability of sensor (j, k) for point i . The limitations (3.3) ensure that the distributed sensors can satisfy the detection area needs $\bar{f}_{ik}$ for a specific point i and sensor type k . In this case, the parameter $\bar{f}_{ik}$ is selected to be larger than the actual coverage requirement of a point i . This enables the distribution of duplicate sensor nodes in the network while staying within the allocated income B , hence optimizing the network duration of existence L .

The binary variable b_{ilj} denotes the connectivity status between a sensor (i, l) and a node at location j . The gateway devices must establish communication with every network component to issue commands with activating/sleeping states and routing parameters. Function (3.4) dictates the deployment of gateways to establish communication with all target points. To ensure network connectivity, it is assumed that every sensor or router component must interact between a minimum of α distinct nodes, as specified with function (3.5). Finally, the function (3.6) imposes binary restrictions on the choice values.

3.2. Online Energy Optimization Model

In the *OEO* model, the sensor nodes are identified and a path is established at t time intervals for transmitting the data collected by each node using small energy. By using this approach, the network's lifetime L is optimized.

The value of z_{jkt} is one if the sensor node (j, k) is active at time t , and zero if it is inactive. The decision parameter y_{iljkt} represents the quantity of data transmitted from sensor (i, l) to sensor (j, k) during period t . The energy used by a node (j, k) in a period t is determined by the constant variable E_{jkt} , as expressed in Formula (3.7).

$$E_{jkt} = e_k^{sr}(1 - z_{jkt}) + e_k^a z_{jkt} + e_k^{ar} \sum_{(i,l) \in N \times K_2} y_{iljkt} + e_k^c \sum_{(i,l) \in N \times K_2} y_{jkilt} + \sum_{j \in N} e_{k'}^s z_{jkt} x_{jk} \quad (3.7)$$

The third parameter in the above function (3.7) provides the input energy, whereas the fourth parameter denotes the output energy. The final value shows the identification and activation of the sensor (j, k) .

The amount of energy used E_{jrt} of a router node (j, r) in a duration ttt is determined by the sleep and active modes, as well as the total input and total outflow, as indicated in Expression (3.8).

$$E_{jrt} = e_r^{sr}(1 - z_{jrt}) + e_r^a z_{jrt} + e_r^{ar} \sum_{(i,l) \in N \times K_2} y_{iljrt} + e_r^c \sum_{(i,l) \in N \times K_2} y_{jrilt} \quad (3.8)$$

In (3.9) the E_{ajt} of an actuator node (j, a) during a time t represents the cumulative use of energy for sleep, active, and transmitted activities.

$$E_{jat} = e_a^{sr}(1 - z_{jat}) + e_a^a z_{jat} + e_a^c \sum_{(i,l) \in N \times K_2} y_{jailt} \quad (3.9)$$

The energy usage E_{jgt} throughout the (j, g) in a period t of the gateway encompasses both the active energy and the input energy in E_{ajt} the Expression (3.10).

$$E_{jgt} = e_g^a + e_g^{ar} \sum_{(i,l) \in N \times K_2} y_{iljgt} \quad (3.10)$$

The *OEO* model aims (3.12) to minimize total energy consumption. This model takes the remaining energy E_{jkt}^{rem} of a node (j, k) as input to optimize. Expression (3.11) computes the residual energy of a node $(j, k) \in N \times K_2$ at time period t in an iterative manner, given the energy consumption $E_{jk(t-1)}$.

$$E_{jkt}^{rem} = E_{jk(t-1)}^{rem} - E_{jk(t-1)} \quad (3.11)$$

The equation (3.13) imposes a maximum limit E_{jkt}^{rem} on the energy usage of a node (j, k) . Constraints (3.14) enable the activation of a node if it has been deployed using the *OND* model, meaning that $x_{jk} = 1$. The constraints (3.15) trigger the activation of sensor nodes to fulfill the coverage requirement of a certain point $i \in N$.

$$\min \sum_{(j,k) \in N \times K_2} E_{jkt} \quad (3.12)$$

s.t.

$$E_{jkt} \leq E_{jkt}^{rem} \quad (j, k) \in N \times K_2 \quad (3.13)$$

$$z_{jkt} \leq x_{jk} \quad j \in N, k \in K_1 \cup \{r\} \quad (3.14)$$

$$\sum_{j \in N} a_{jki} z_{jkt} \geq f_{ik} \quad i \in N, k \in K_1 \quad (3.15)$$

$$\sum_{(j,k) \in N \times K_2} b_{jki} z_{jkt} \geq \alpha z_{ilt} \quad (i, l) \in N \times K_2 \quad (3.16)$$

$$y_{iljkt} \leq \bar{M} b_{ilj} z_{ilt} \quad (i, l), (j, k) \in N \times K_2 \quad (3.17)$$

$$y_{iljkt} \leq \bar{M} b_{ilj} z_{jkt} \quad (i, l), (j, k) \in N \times K_2 \quad (3.18)$$

$$y_{ililt} = 0 \quad (i, l) \in N \times K_2 \quad (3.19)$$

$$\sum_{(i,l) \in N \times K_2} y_{iljkt} + h_{jk} z_{jkt} = \sum_{(i,l) \in N \times K_2} y_{jkilt} \quad j \in N \quad (3.20)$$

$$\sum_{(i,l) \in N \times K_2} y_{iljkt} = \sum_{(i,l) \in N \times K_2} y_{jkilt} \quad (j, k) \in N \times \{r, a, g\} \quad (3.21)$$

$$\sum_{(i,l) \in N \times K_2} y_{iljkt} + h_{jk} z_{jkt} + \sum_{(i,l) \in N \times K_2} y_{jkilt} \leq \beta_s \quad j \in N \quad (3.22)$$

$$\sum_{(i,l) \in N \times K_2} y_{iljkt} + \sum_{(i,l) \in N \times K_2} y_{jkilt} \leq \beta_k \quad (j, k) \in N \times \{r, a, g\} \quad (3.23)$$

$$y_{iljkt}, E_{jkt} \geq 0 \quad (i, l), (j, k) \in N \times K_2 \quad (3.24)$$

$$z_{jkt} \in \{0,1\} \quad j \in N, k \in K_1 \cup \{r\} \quad (3.25)$$

4 HARDWARE DESIGN

The hardware system, which is superficially mentioned in Chapter 2, is presented in detail in this chapter. The system is designed specifically with the objective of implementing and testing the OND and OEO models discussed in Chapter 3, while providing the lowest power usage. In accordance with the principles of mathematical modeling, two primary elements have been used: a gateway and a sensor node. Two distinct sensor nodes are developed to evaluate disparities in the functionalities of the sensors. The gateway manages these two unique sensor nodes. These sensor nodes and the gateway communicate with each other over a wireless communication standard. The efficiency of the communication protocols has been analyzed with the hardware system developed in this project.

Section 4.1 discusses the gateway system principle that manages the entire system. Section 4.2 provides an in-depth explanation of the functionality and design of sensor nodes.

4.1. Gateway Design

This section provides a detailed explanation of the gateway system and its operational principles shown in Figure 4.1.

A gateway has been developed to regulate the sensor nodes and modify the operational concept. The hardware gateway system utilized the E32 LoRa module in conjunction with the Atmega 328p microcontroller. The E32 LoRa module has been chosen for wireless communication purposes. The LoRa module is equipped with seven pins: M0, M1, RXD, TXD, AUX, VCC, and GND.

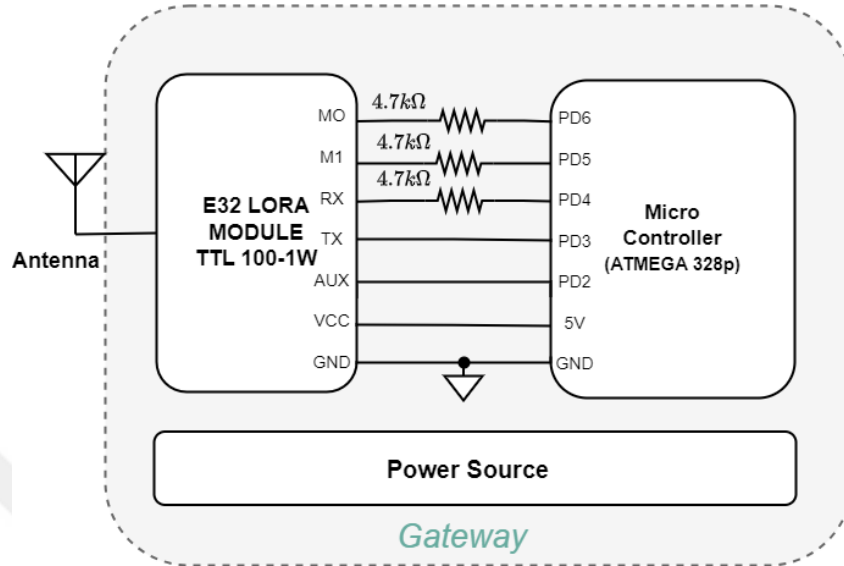
The E32 LoRa TTL 433 MHz 1W module is a flexible wireless communication device that functions at a frequency of 433 MHz utilizing LoRa technology. It supports four

distinct operating modes: normal, wake-up, power-saving, and sleep mode. In Mode 0, the module runs with modest power consumption and minimal latency, making it suitable for real-time data transfer. Wake-up Mode (Mode 1) minimizes power usage by periodically activating to check for data, making it suited for applications that can accommodate some delays. Power-saving Mode (Mode 2) optimizes power consumption by mostly entering a sleep state and only awakening when necessary, making it ideal for battery-operated devices that have sporadic data transfer. Sleep Mode, also known as Mode 3, optimizes power consumption by remaining in a state of deep sleep until activated by an external trigger. This mode is particularly beneficial in situations where maximal power conservation is necessary. The transition between these modes is controlled by the M0 and M1 pins, providing configurability options through the microcontroller. Optimizing the module's performance and guaranteeing reliable communication require proper configuration and optimal antenna selection.

Sensor nodes utilize the Atmega 328p microcontroller to connect various sensors with attributes like humidity, temperature, and pressure. This system has been established to improve a thorough evaluation of its functionality, efficiency, and communication protocols, with a specific emphasis on the wireless connectivity that permits efficient data flow between the gateway and sensor nodes.

The physical connections between LoRa and the microcontroller are shown in Figure 4.1. Since the microcontroller works with 5 V and the LoRa module works with 3.3 V, a 4.7 k Ω resistor is placed between the connections of the two for security purposes. This warning is also included in the datasheet of the LoRa module. If the communication line uses a 5V level, a 1 k Ω and 5 k Ω resistor must be connected in series [34].

Figure 4.1. Block Diagram of the Gateway Design



Since the microcontroller works with 5V, a resistor between 1 k Ω and 5 k Ω should be used and connected in series.

The gateway manages the sensor nodes using code programmed with microcontrollers. The system monitors the data received from the sensor nodes and enables its transmission to consumers. To achieve the minimal power usage described in the mathematical model, sensors avoid performing continuous measurements. The gateway transmits a continuous signal to the sensor nodes. This hardware design involves the communication of two distinct sensor nodes and a gateway across the second and third channels, operating at frequencies of 432 MHz and 433 MHz, respectively.

By improving the code, the gateway supports the activation of the sensor nodes in sleep mode through the transmission of a signal at specified times (initially configured to occur at a frequency of 30 minutes during the initial testing phase). The architecture has been designed to efficiently substitute between various roles, such as gateway, type-1, and type-2 sensor nodes.

4.2. Sensor Node Design

In this section, the sensor nodes designed for the project are presented. Section 4.1 explains the first node designed with an EC Soil and RD-STME-P524 three-in-one Electrical Conductivity sensor. Section 4.2 details the second sensor node designed with an SHT3X air temperature and humidity sensor.

4.2.1. First Sensor Node Design

EC Soil and RD-STME-P524 three-in-one Electrical Conductivity Sensor is utilized in the initial sensor node that is designed. The conductivity, humidity, and temperature of the soil are all things that this sensor can monitor. A universal synchronous receiver (UART) has been incorporated into the system in order to facilitate serial communication between the sensor node and the microcontrollers. This is due to the fact that the microcontroller unit does not support the RS485 protocol. The application of a high-sided switch is carried out with the help of the BC327 PNP BJT transistor in order to reduce the overall power consumption of the apparatus. The power to the hardware is supplied by three AA batteries.

This sensor is used to assess electrical conductivity by detecting the presence of ions that are dissolved in the soil. It is equipped with at least two probes containing electrodes, which enable it to be inserted into the ground. The resistance and electrical conductivity of the soil can be evaluated by passing an electric current between these electrodes. Temperature is measured using a thermistor, which is a type of temperature sensor. Humidity levels are measured by detecting changes in the capacitor, similar to other sensors. Once the measurements are taken, the sensor sends the data to the microcontroller using the RS-485 signal output. RS485 is an industry-specific communication protocol. Unlike alternative communication protocols, it facilitates data transfer via two distinct cables referred to as A and B. The block diagram of the proposed circuit is shown in Figure 4.2.

Figure 4.2. Block Diagram of the Hardware System Set Up

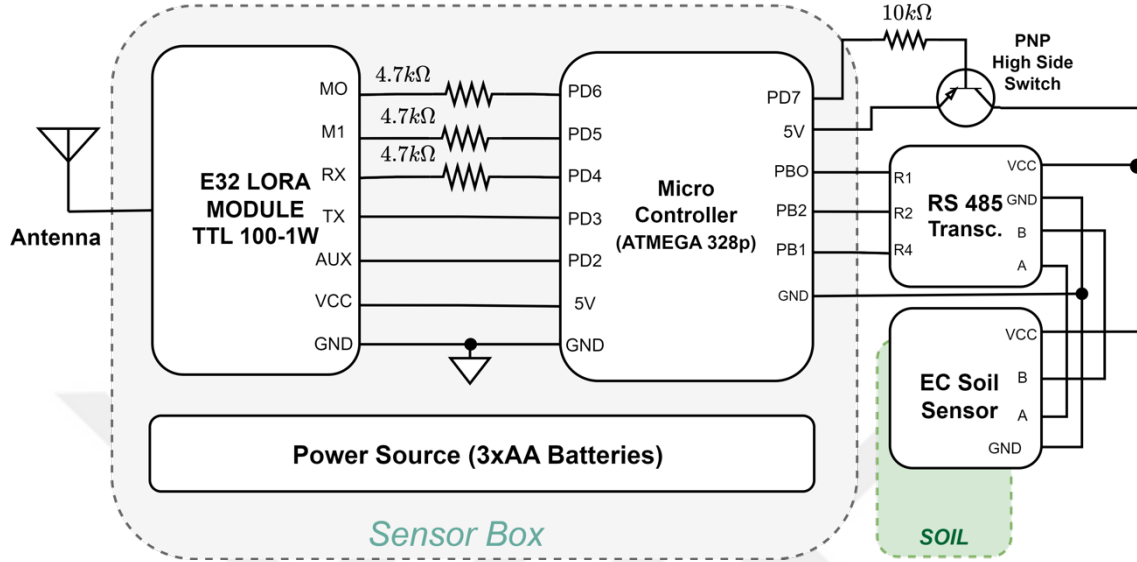
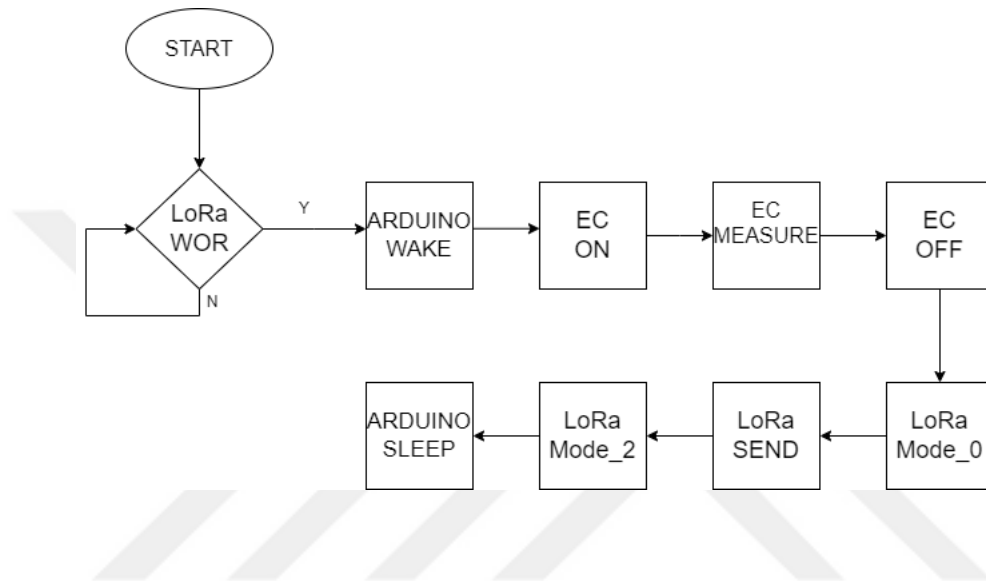


Figure 4.2. shows the detailed circuit diagram of the sensor node. As in the gateway design, a $4.7k\Omega$ resistor is preferred to ensure a secure connection between the E32 LoRa module and the microcontroller. This resistor is used to connect the M0, M1, and RX pins on the E32 LoRa module to the microcontroller. Additionally, in the coding part, a PNP high-side switch transistor is added so that the sensor can work properly in sleep mode. A $10k\Omega$ resistor is used for the connection between the transistor and the microcontroller. The BC327 PNP BJT transistor is employed in this sensor node to operate in conjunction with the designed code, hence minimizing power consumption. The base terminal regulates the activation and deactivation of the switching activity. When the switch is in the open position, it permits the flow of current from the power supply to the sensor. However, when the switch is in the closed position, it functions as an external interrupter and blocks the flow of current from the power supply to the sensor.

Figure 4.3. shows the working principle of the sensor node. The hardware system is engineered with a focus on minimizing power consumption, a crucial aspect for extending battery life. Consequently, when data transmission and measurement processes are

inactive, the microcontroller is configured to enter deep sleep mode. In this state, circuit elements do not consume power during this period.

Figure 4.3. Flowchart of the First Hardware System Set Up



4.2.2. Second Sensor Node Design

In this section, the second sensor node detailed in Figure 4.4. is explained. A SHT3X air temperature and humidity sensor is preferred for the second sensor node designed. This sensor is used to measure air temperature and humidity. Wireless communication was also provided using the Atmega 328p microprocessor and the same LoRa module used in the gateway.

A specialized software has been developed to reduce power usage, similar to the initial sensor node. The system operates based on the following operating principle: The sensor node initially operates in a sleep state but becomes operational upon receiving a signal order from the gateway. Afterwards, the measurement is obtained and transmitted to the gateway. The node then returns to sleep mode until receiving the next order from the gateway. This operational method minimizes energy consumption, hence achieving low power utilization. In comparison to the initial sensor node design, an EEPROM system

has been developed to ensure uninterrupted access to measured data and prevent data loss in the event of power interruptions.

Disruption of the Arduino's power supply will not result in the deletion of data stored in the EEPROM. Once the Arduino initiates its operation, it promptly stores crucial data in this designated region. The location of the EEPROM storage region differs based on the specific microprocessor employed in the Arduino. The Atmega328 microcontroller has a storage capacity of 1024 bytes, while the ATmega168 and Atmega8 microcontrollers have a capacity of 512 bytes. On the other hand, the ATmega1280 and ATmega2560 microcontrollers offer a larger storage space of 4 kilobytes [35]. Since this study uses the ATmega328p microcontroller, it has 1024 bytes of EEPROM memory. The data collected by the sensor nodes is saved in this memory. The enhanced approach allows for the compression of 4 bytes of measurements into a single byte. The measurement of temperature readings requires 1 byte, whereas index values necessitate 2 bytes. A total of 4 bytes of RAM is allocated for the purpose of reading data from the sensor. Over the course of a one day, a total of 192 bytes of memory are consumed in a continuous operation that spans 24 hours, with measurements taken every 30 minutes. Hence, this memory has the capacity to store nearly five days' worth of data. The identical algorithmic approach is used for collecting the humidity data. This method utilizes compression to record data, resulting in the most cost-effective utilization of memory space.

Figure 4.4. Block Diagram of the Second Hardware System Set Up

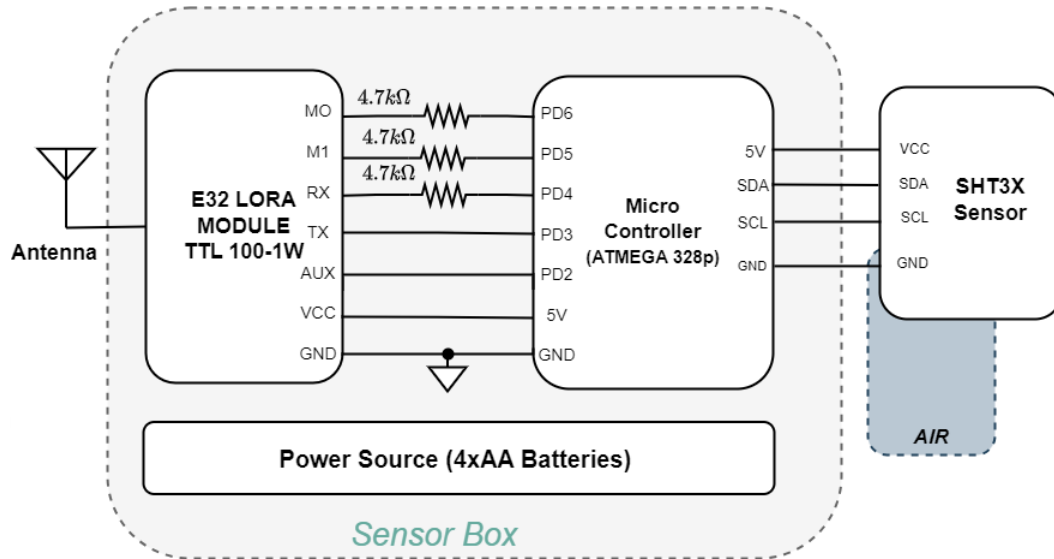
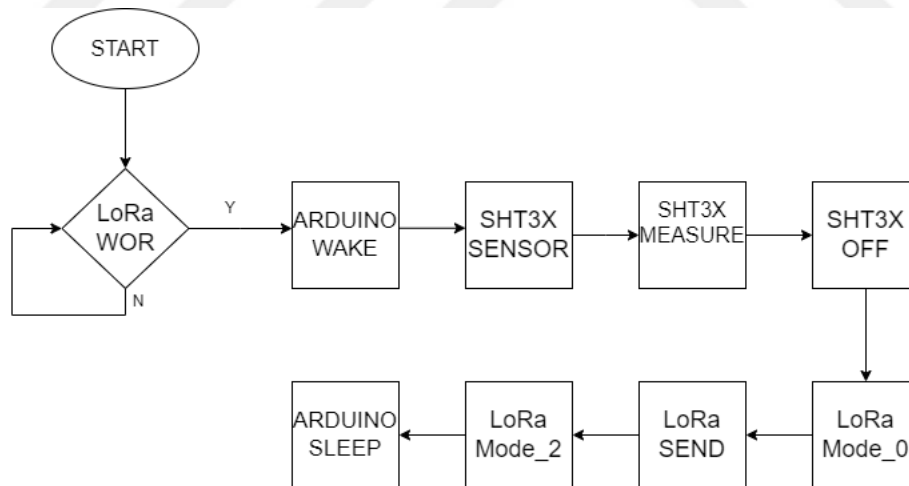


Figure 4.5. Flowchart of the Second Hardware System Set Up



HARDWARE IMPLEMENTATION

5.1. Gateway Implementation

The channels are divided to provide easier measurements. The smart agricultural sensor node transmits information about soil parameters through the 2nd channel. The sensor node, equipped with the SHTX sensor, collects data on air temperature and humidity levels through the 3rd channel. The smart parking sensor node in the parking lot transmits data through the 4th channel.

Figure 5.1. View of the Implemented Gateway in a Waterproof Box

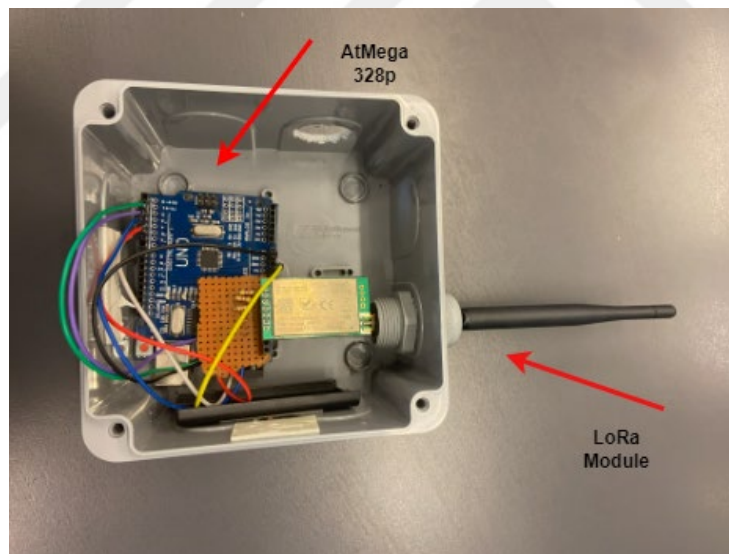
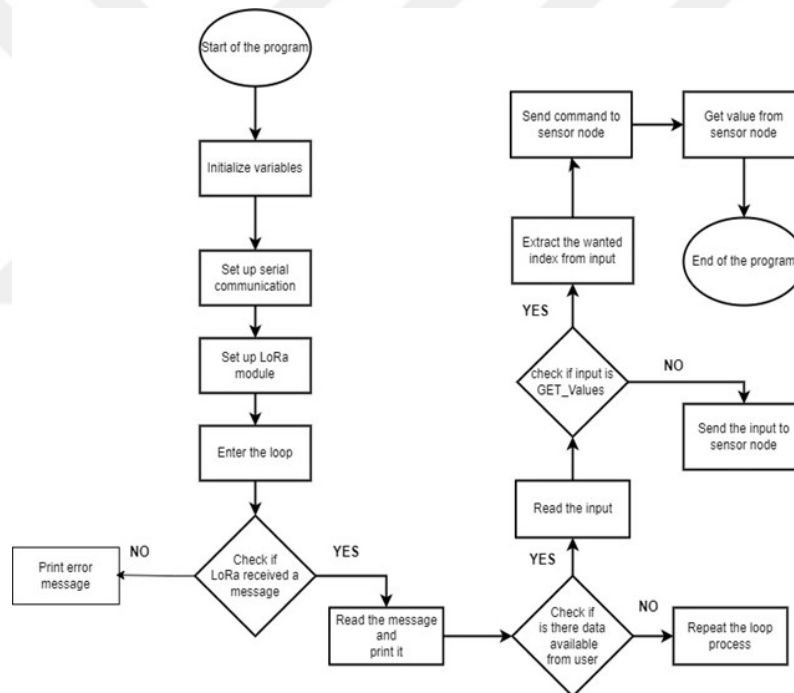


Figure 5.2. shows the coding stages of the gateway developed with the Atmega 328p microcontroller. The system starts with initializing the required variables, followed by establishing serial communication and configuring the LoRa module. Once these initial steps are completed, the program enters a continuous loop. In this cycle, it checks whether a message is received via the LoRa module. If a message is received, it is read and printed. The program then extracts the required index from the input message and checks whether

the input command will retrieve the values GET_Values. If the input is indeed a GET_Values, the program reads the input message and verifies whether there is data from the user. If data is available, it sends a command to the sensor node and receives the relevant value from the sensor node. If the input is not a GET_Values, the input is sent directly to the sensor node and the program continues to the next iteration of the loop. If no message is received via the LoRa module, an error message is printed and the cycle continues. The program repeats this process continuously until it is explicitly terminated.

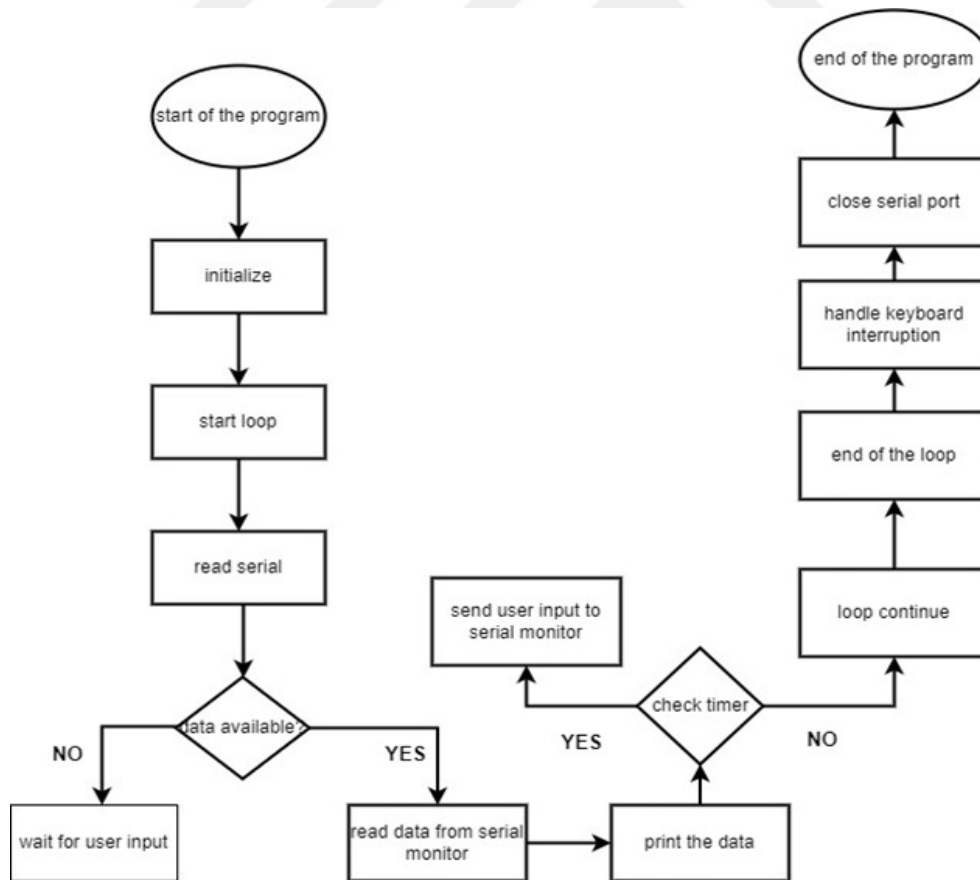
Figure 5.2. The Flowchart of Gateway Arduino Code



A Python code has been developed alongside the Arduino Atmega 328p microcontroller code in the gateway module. This code functions similarly to the serial monitor in Arduino code and automatically retrieves measurements by periodically requesting data from the sensor node that stores the SHTX sensor throughout the daytime, with a frequency of every half hour. The operational rationale of this code can be accessed from Figure 5.3. The system commences by initializing variables and configuring initial

settings. Subsequently, it initiates an uninterrupted cycle in which it retrieves information from the serial input. The application verifies the presence of data in the serial input. In the absence of any available data, the system remains in a state of waiting until the user provides input. When data is present, the program retrieves the data from the serial monitor and transmits the user input to the serial monitor. Afterwards, the program verifies a timer, and if the timer condition is fulfilled, it outputs the data. If the condition of the timer is not fulfilled, the loop will persist. The program iterates through this loop until it reaches the termination condition. Currently, it manages any interruptions from the keyboard, terminates the connection to the serial port, and then concludes the execution of the application.

Figure 5.3. The Flowchart of Gateway Python Code



5.2. Sensor Nodes Implementation

The fundamental operational idea of the hardware is explained in Algorithm 1. The system can be started by means of RF signals. Furthermore, a specialized data package has been developed, comprising a timestamp, the number of bytes in the message, and the source and destination information. The intended action, such as doing a measurement or setting the node as type-1, type-2, or gateway, is transmitted through the message's content.

Figure 5.4. View of the Implemented Sensor Node in Area

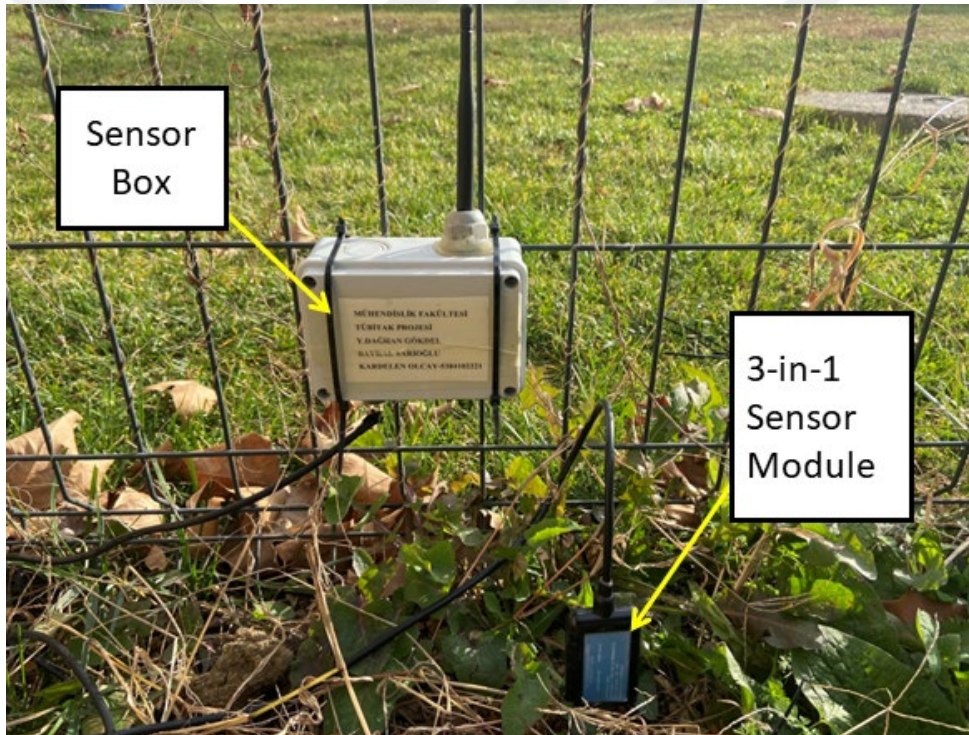


Figure 5.5. The Algorithm of the Implemented Hardware

Algorithm 1 The main loop of the implemented hardware

```
1: Wake-on-RF receive
2: Parse recieved data package for source, destination and
  message
3: Check package source and destination
4: if destination != self then
5:   Configure LORA for the target destination
6:   Fill the trasmit buffer with the message, enable trans-
    mit mode
7: end if
8: if message == measure then
9:   Turn on the sensor module via the high-side-switch
10:  Get the sensor measurement and save it to the EEP-
    ROM data buffer and the transmit buffer, enable transmit
    mode
11: end if
12: if message == configure then
13:   Switch to the target configuration (type-1, type-2,
    gateway)
14: end if
15: Turn off the system via the high side switch and enable
    the sleep mode
```

Figure 5.6. displays an example output for t-h values obtained from the sensor in binary mode and the output of the GET-Values function. The working principle of this code developed for EEPROM is divided into two for humidity and temperature data. The amount of humidity is obtained via the SHT31 sensor and is represented as a floating-point data type. Each number part of this value is then transformed into a binary representation. A comparison is conducted to see if the incoming temperature exceeds 0.5 for the decimal part of the value. If the value exceeds 0.5, it is assigned a value of 1. If the value is below 0.5, it is assigned a value of 0. By utilizing this numerical value alongside the initial binary value, an attempt is made to derive a new value consisting of 8 bits, or equivalently, 1 byte. The objective is to assign the decimal component as the least significant bit and merge it with the binary value. To achieve this, the integer component of the humidity value is initially moved to the left by 1 bit. After that, the binary value is subjected to a bitwise AND operation to obtain the least significant bit. Ultimately, the bitwise OR operation is employed to combine the value obtained from

left shifting with the extracted least significant bit. By following this method, a binary value is achieved.

In the sample code output, the humidity level appears to be 51.0. When this value is inputted into the implemented binary algorithm, the resulting output is observed as 1100110. As the fractional component of the data received from the sensor is less than 0.5, it is represented as 0. The decimal value 51 can be represented in binary as 110011. Thus, by combining these two pieces of information, the resulting output value of 1100110 is assumed to be accurate.

Figure 5.6. The Example Output Code

```
Values1: ['H:102', 'T:51', 'V:4987.36']
Humidity: 1100110
Temperature: 110011
Humidity result: 51.0
Temperature result (process_binary_value): 25.5
Enter 't' or 'h' or 'GET_Values': GET_Values 3
Message sent to node. Waiting for reply. 20 secs.
INDEX 3 H:126 - T:44

Output1: INDEX 3 H:126 - T:44
Values1: ['INDEX', '3', 'H:126', '-', 'T:44']
Humidity: 1111110
Temperature: 101100
Humidity result: 63.0
```

An equivalent technique has been devised for the measurement of the outside temperature via the SHT31 sensor. The main difference from the humidity level method lies in the algorithm's consideration of whether the temperature value is below or above 0. If the temperature value is less than 0, the most significant bit of the binary format to be formed is set to a specific value. If the temperature is positive, it is allocated a value of 1; otherwise, it is assigned a value of 0. The following steps are similar to those of the

humidity algorithm. To ensure that the 8-bit field is utilized in these binary conversions without exceeding its capacity, specific value ranges have been established for both humidity level and air temperature. If the measured value falls outside of these specified ranges, it is inferred that the data is questionable. The procedure determines these value ranges. Once the humidity and temperature data have been translated, they are subsequently written to the EEPROM.

To facilitate monitoring of data, an index value is initially allocated to the data. For instance, under the assumption that the first data is written starting from the index value, the algorithm operates in the following method: The index value 1 is used to write bytes 0 and 1. The subsequent humidity value is represented by the second byte. The temperature value is stored in the third byte. Next, the value at the 2nd index is assigned to the 4th byte. The cycle perpetuates when the byte is overwritten. In this code portion, the variable "index-address" is initialized to facilitate the comparison between the index value and the index address. Four times the index is equal to the index-address. As an example, the value at the 6th index is stored in the 24th byte. Upon reaching a value of 1024, the index-address resets and begins overwriting the current values, starting from the previous data.

A function named get-values-at-index has been implemented to get the previously stored data via the gateway and ensure continued access to the data in case of a potential power failure. This function retrieves the value at a specific index, determined by the user's input value.

Multiple applications have been developed for both the sensor node code and the gateway to enable access to the data received from the SHT31 sensor and saved in EEPROM. This access may be done at any given time and ensures the safety of the data in case of power loss. A code has been implemented on the sensor node to handle the "GET-Values" command received from the gateway. This code extracts the user-entered index value from the designated place in the EEPROM.

6 EXPERIMENTAL RESULTS AND TESTS

The first sensor node designed for WSN was developed to test the car park at Istanbul Bilgi University. The Raspberry Pi 4 Model B (Pi4B) was used as a network gateway to collect data from the sensor and keep the sensor node under control. An Arduino Uno was used as the sensor node. Temperature-humidity sensor and air quality sensor were selected for testing in the parking lot. The accuracy of these two sensors was tested using Arduino Uno. The sensor node was placed in a waterproof case to protect against all kinds of weather conditions, dust, wind, etc. In addition, waterproofness was tested in a laboratory environment.

The purpose of using the sensor node in the parking lot can be explained as follows: Sustainable communities and cities improve quality of life. Sustainable cities are also necessary to prevent damage to nature. Urbanization increases the need for transportation. Therefore, the use of cars in big cities is high. The transportation shortage in cities is an obstacle for sustainability. It is essential that environmentally friendly transportation system to achieve sustainability aims. In this project, the transportation problem in Istanbul, one of the most crowded cities, was researched. The problem of parking is increasing day by day in Istanbul due to the use of cars. Inefficient use of parking spaces leads to traffic congestion. That's why, the main purpose of this project is to reduce traffic congestion by developing a smart parking system. Another main purpose is reducing air pollution in city centers. more parking revenue can be created by reducing human cost and time loss. Also, detecting and recording multiple features of vehicles and drivers is necessary to move the project forward.

The main aim of this project is to establish an intelligent parking system that will help improve the quality of life for Istanbul inhabitants by decreasing traffic congestion, air pollution, noise, and the time spent searching for available parking spaces [36].

Figure 6.1. The Sensor Node General Scheme

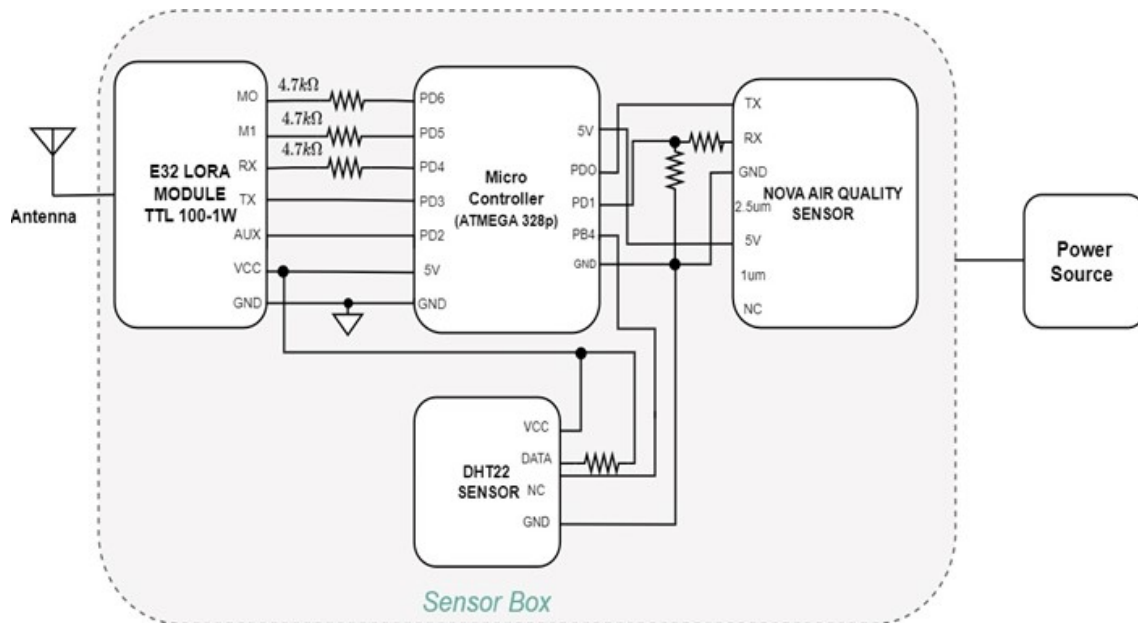


Figure 6.2. The Parking Sensor Node General Scheme

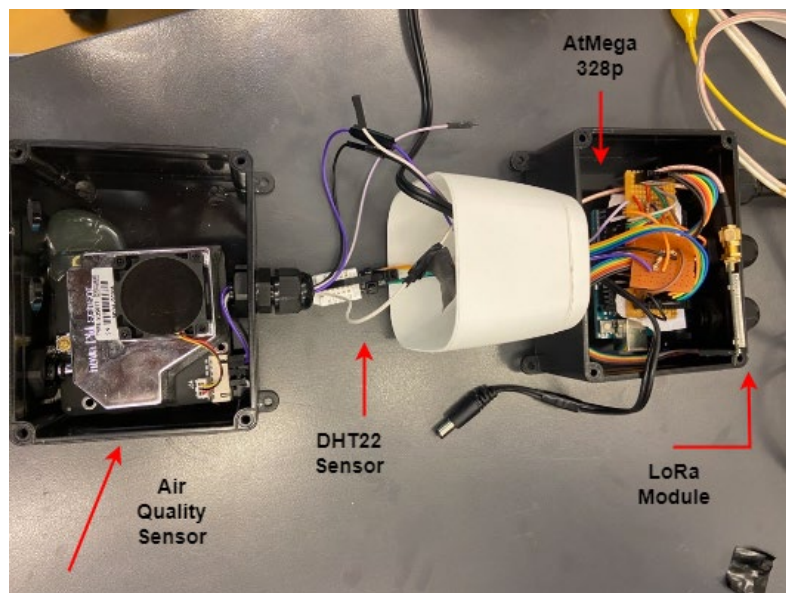


Figure 6.3. The Gateway General Overview

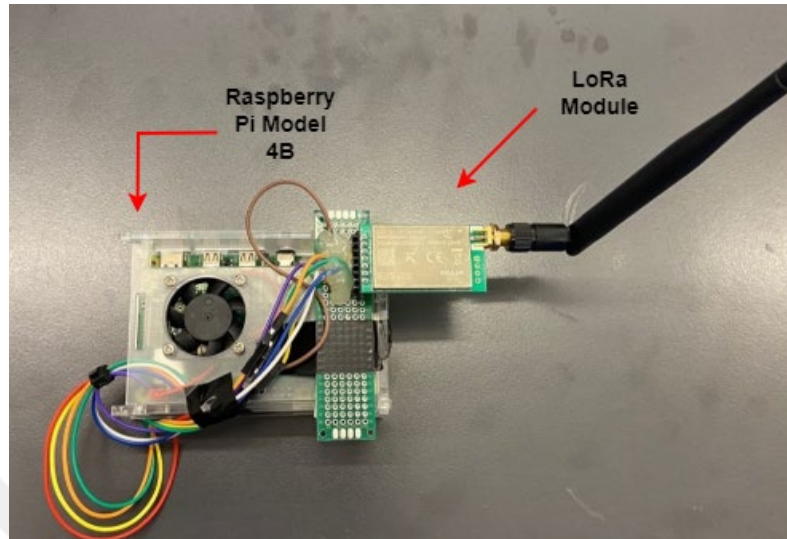


Figure 6.4. Overview of Sensor Node in the Parking IoT

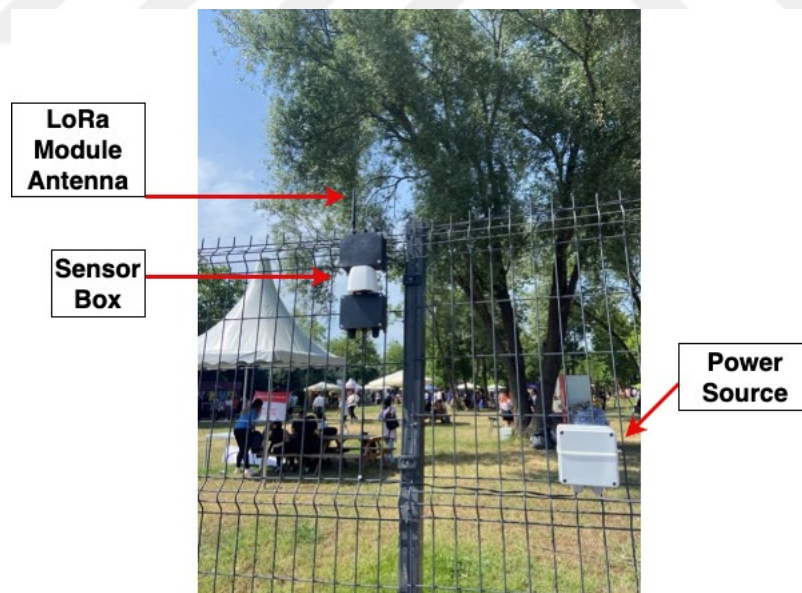


Table 6.1. The Average Current Utilization of the Implemented Hardware

Operation	Sleep Current (mA)	Wake Current (mA)
Sensing and Transmitting	6	31
Only Transmitting	6	89
Only Receiving	6	21

The circuit's current usage during operation is comprehensively displayed in Table 6.1. The sensor node's current consumption is 6 mA in sleep mode, 31 mA when sensing and sending, 89 mA when only transmitting, and 21 mA when only receiving data.

Table 6.2. Bill of Materials

Network Element	Cost (TL)
Gateway	980
Soil Sensor Node	1200
Temperature- Humidity Sensor Node	692

Figure 6.5. The Measurement of the Power Sources Consumed During Operation

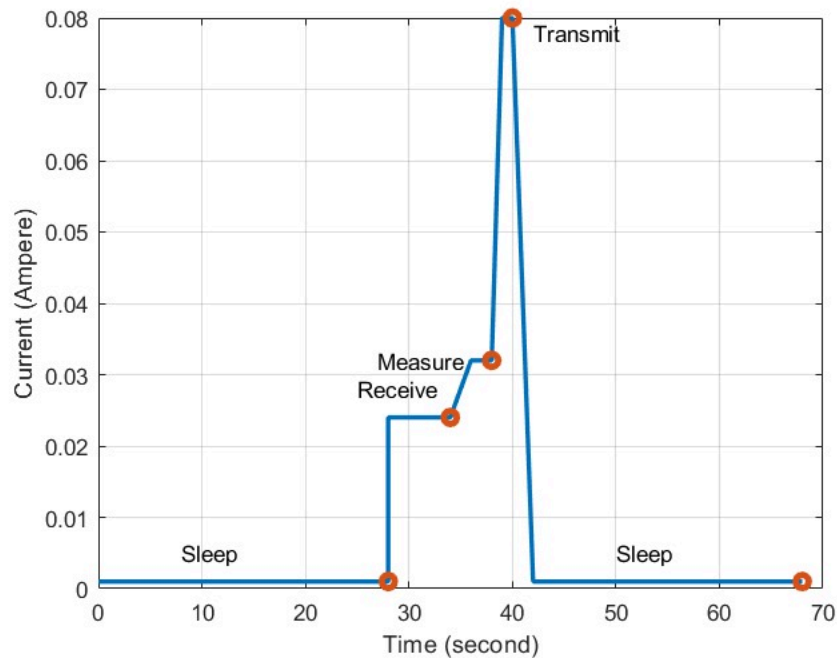


Figure 6.5. presents the simulation results representing the change of the sensor's current consumption in different operating modes over time, while maintaining a constant 5 Volt input. The sensor alternates between sleep, receive, measure, and transmit modes, each distinguished by distinct current profiles. While in sleep mode, the sensors function in a power-saving standby mode, consuming minimal energy. When transitioning to the receiving mode, there is a noticeable increase in the current, which indicates the power demands associated with acquiring the signal. During the measurement mode, which is specifically designed for measuring humidity and temperature values, an average current level is shown. During data transmission, there is a rise in energy demands, which is reflected by a peak in current consumption in the transmission mode. These findings emphasize the significance of modifying power management at every phase for energy efficiency.

In order to determine the average current consumption (I), it is necessary to monitor the duration of time (T) that the device is in each mode for the entire operational period (30 minutes). LoRa module functions in four distinct modes: sleep, idle, receive, and send. According to the Table 6.1, the study shows that the sleep mode 6 mA, standby mode 31 mA, receive mode 21 mA, and transmit mode 89 mA consumes current.

The formula for calculation:

$$I_{\text{avg}} = \frac{(I_{\text{sleep}} \times T_{\text{sleep}}) + (I_{\text{idle}} \times T_{\text{idle}}) + (I_{\text{receive}} \times T_{\text{receive}}) + (I_{\text{transmit}} \times T_{\text{transmit}})}{T_{\text{total}}} \quad (6.1)$$

Since data is received every half hour, T_{total} is chosen as total half hour. Receive is measured in 6 seconds, transmit and idle are measured in 2 seconds. Accordingly, sleep time is 1790 seconds.

$$I_{\text{avg}} = \frac{(0.006 \times 1790) + (0.031 \times 2) + (0.021 \times 6) + (0.089 \times 2)}{1800} \approx 6.17 \text{ A} \quad (6.2)$$

$$\text{Total Charge} = Q_T = I_{\text{avg}} \times T_{\text{total}} = 11.106 \times 1800 = 5.55 \text{ Ah} \quad (6.3)$$

The energy usage in joules (J) can be determined by utilizing the supply voltage of 5 V:

$$E_J = Q_T \times V \times 1.8 = 49.95 \text{ J} \quad (6.4)$$

This study provides a unique framework designed to fulfill the special needs of a flexible sensor network optimized for low-power IoT applications. The suggested architecture prioritizes the integration of adaptable sensor nodes, specifically engineered to achieve an exceptional level of energy efficiency in different application contexts. In order to accomplish this objective, the essential model parameters and formulations required for the efficient functioning of the proposed system were carefully derived. These theoretical features were then effectively implemented at the hardware level to make them feasible.

Furthermore, the research efforts involved conducting thorough field testing of the constructed network. The proposed method was validated by a series of thorough experiments and real-world simulations, collecting empirical evidence to support its effectiveness and feasibility. The hardware deployed in the field is demonstrated in Figures 6.6. and 6.7. The findings from thorough testing clearly confirm that the adaptive sensor network, with its adjustable nodes and improved power productivity, is a potential solution for minimal-power IoT applications. In addition to strengthening the theoretical underpinnings of adaptive sensor networks, this study provides practical insights that help advance and implement energy-productivity solutions in the expanding realm of IoT technologies.

Figure 6.6. The Measurement of the Power Sources Consumed During Operation

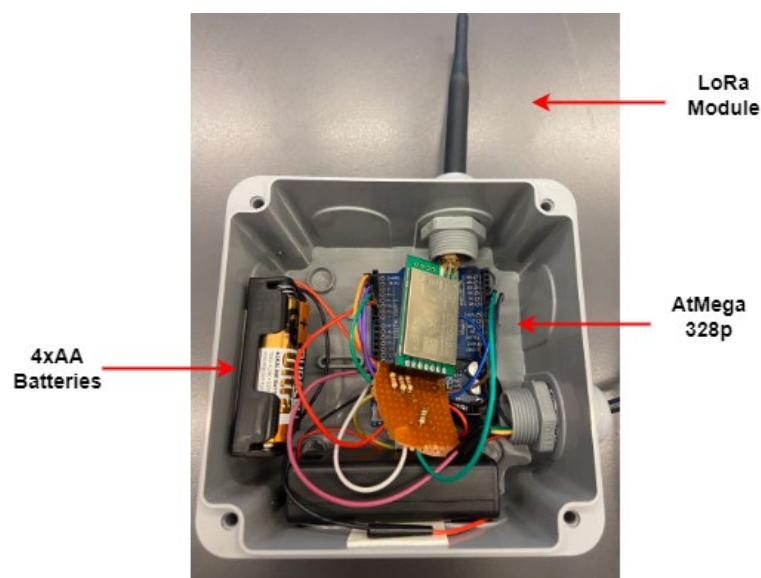
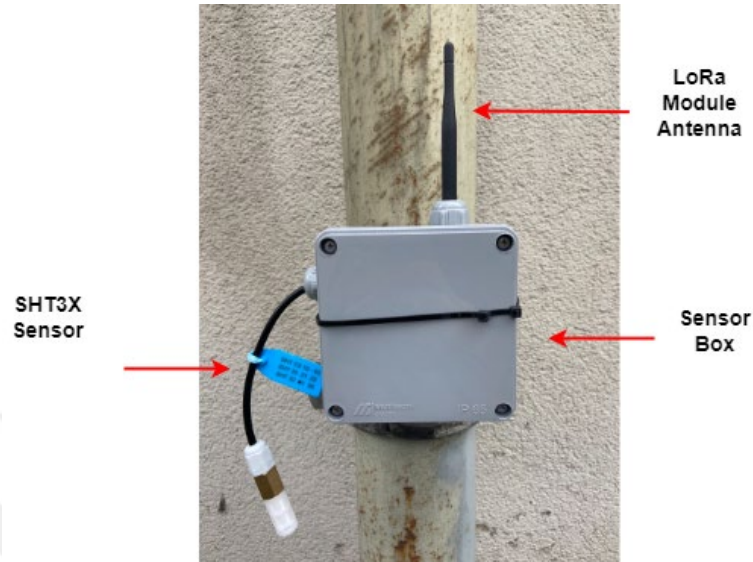


Figure 6.7. External View of the Sensor Node



The statistics show a particular 10-minute timeframe on December 18, 2023, which was chosen for a thorough analysis of temperature and humidity data. The temperature and humidity readings, which were marked with timestamps, were extracted and graphed to illustrate their fluctuations within the required period. Figures 6.8. and Figure 6.9. display the fluctuating temperature and humidity levels at consistent intervals over a span of 10 minutes. The horizontal axis, known as the x axis, displays the timestamp, while the vertical axis, referred to as the y axis, corresponds to the temperature and humidity values, measured in degrees Celsius. This analysis offers a detailed examination of how temperature and humidity change at the same time, providing a nuanced understanding of how they interact within a specific and important period of time. By analyzing both metrics, a comprehensive understanding of the environmental circumstances during this particular time segment is attained.

Figure 6.8. Humidity Data with Dates and Times

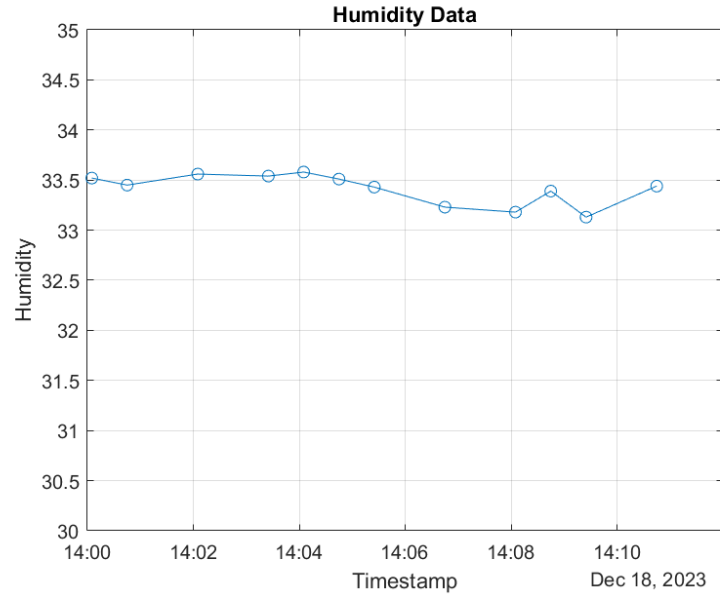
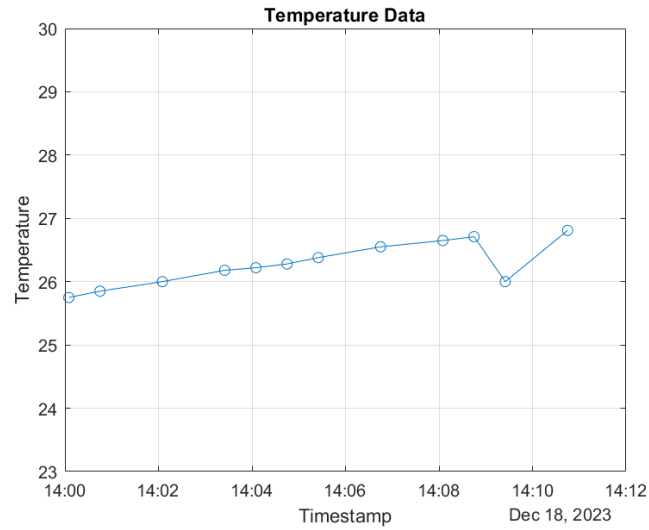


Figure 6.9. Temperature Data with Dates and Times



A mesh network was established at Istanbul Bilgi University to test the accuracy of the mathematical modeling used in this thesis. ESP8266 was used as a micro controller in

this network. Wireless communication was made via Wifi. The sensor nodes were programmed to communicate with each other, as included in the model. As described in the mathematical model, the sensor nodes were found to be more advantageous than the lora module in terms of both coverage and cost. Table 6.3 contains these comparison data.

Table 6.3. Comparison of LoRa and ESP8266

	NodeMCU ESP8266	E32-TTL-1W	NodeMCU E32-TTL-1W ESP8266 and Antenna	ESP8266 and Antenna
Cost	144 TL	2178 TL	2428 TL	196 TL
Range	25 m	3 Km	3 Km	290 m
Cost/Range	5.75	0.726	0.80933	0.67586

The maximum packet between two ESP8266 is limited to 25 meters which is the first distance measurement with the microcontroller. When the connection is established between three ESP8266's, the distance increases up to 108 meters which the reason is that the middle ESP8266 is acting as a forwarder. Data which is coming from the one end to the middle one, it can transmit a data packet to the ESP8266 at the other end. After the distance measurements are completed, temperature data of the room and data obtained from the sensor were compared with the usage of a digital thermometer. The reason is to test the system and control if the reliability of the data flow from the sensors are correct. An external antenna was attached to ESP8266 in order to increase the measurement range of the overall system. According to the obtained results, the distance between two ESP8266 has been increased by 253 meters.

Figure 6.10. Distance Measurement Map



(Source: Google Maps, “İstanbul Bilgi Üniversitesi,” [Online]. Available: <https://www.google.com/maps/place/%C4%B0stanbul+Bilgi+%C3%9Cniversitesi/@41.0677627,28.9441436,17z/data=!3m1!4m6!3m5!1s0x14cab0ce099c5d49:0x9f665fc172ad4624!8m2!3d41.0677587!4d28.9467185!16zL20vMDgxNXJ4?entry=ttu>. [Accessed: 24-Jul-2024].)

7 CONCLUSION

This study presents a two-stage mathematical model and its hardware implementation for the task of deploying sensors throughout agricultural land and collecting data from the field to enhance crop productivity. The implementation of IoT in agriculture will enhance the farmer's ability to monitor and manage their fields, reduce the need for manual work, and enhance overall production.

The primary objective of this study is to explore the practical application of WSNs in the context of agriculture. In order to make the most effective use of the limited energy resources available, researchers have created optimization models. The goal is to reduce network deployment expenses by utilizing the Offline Network Deployment Model and enhance the efficiency of the network operation process through the Online Energy Optimization Model.

Also, this thesis includes the design, production, and evaluation of the smart parking system, a prominent subject of study in the area of smart cities. This system aims to provide solutions to parking issues in Istanbul, particularly at Istanbul Bilgi University parking lot. The sensors in this system enable continuous data measurement and transmission. The transmitted data allows the collection of other metrics, such as air-humidity measurement, in the parking lot. The addition of new sensors will enable the measurement and tracking of gas emissions from automobiles in the parking cells. Additionally, the booking application will address the issue of consumers spending a significant amount of time seeking a parking place. Conducting this research will have a positive impact on both the environment and the users.

After the design of the initial two sensor nodes, a third sensor node has been constructed. The sensor node includes an SHT3X temperature-humidity sensor. The SHT3X sensor is positioned adjacent to the smart agricultural sensor node. While the node is measuring the soil characteristics, the SHT3X sensor is simultaneously measuring the humidity and temperature of the same area. This sensor node utilizes a storing method that has been devised to maintain control over the values. If the power to the system is unexpectedly

shut off, the measurements will not be lost. An EEPROM algorithm was specifically developed for such objectives.

In the future, a solar system will be implemented for the smart agriculture sensor node, resulting in an extended lifespan for the sensor. The system will utilize the solar panel system's power when there is sufficient energy available. In the event that the panel does not have enough energy, the system will rely on the pre-installed battery power in the sensor node.



REFERENCES

- [1] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3758-3773, 2018. DOI: 10.1109/JIOT.2018.2844296.
- [2] V. K. Quy, N. V. Hau, D. V. Anh, N. M. Quy, N. T. Ban, S. Lanza, G. Randazzo, and A. Muzirafuti, "IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges," *Applied Sciences*, vol. 12, no. 7, 2022, [Online]. Available: <https://www.mdpi.com/2076-3417/12/7/3396>.
- [3] A. Purohit, F. Mokaya, and P. Zhang, "Collaborative indoor sensing with the sensorfly aerial sensor network," in *Proc. 11th Int. Conf. Information Processing in Sensor Networks*, 2012, pp. 145–146.
- [4] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor, "IoT and agriculture data analysis for smart farm," *Computers and Electronics in Agriculture*, vol. 156, pp. 467–474, Jan. 2019.
- [5] N. Islam, M. M. Rashid, F. Pasandideh, B. Ray, S. Moore, and R. Kadel, "A review of applications and communication technologies for internet of things (Iot) and unmanned aerial vehicle (uav) based sustainable smart farming," *Sustainability*, vol. 13, no. 4, p. 1821, 2021.
- [6] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A Study of LoRa: Long Range Low Power Networks for the Internet of Things," *Sensors*, vol. 16, no. 9, 2016, [Online]. Available: <https://www.mdpi.com/1424-8220/16/9/1466>.
- [7] M. Badreddine, E.-B. Bourennane, S. Boucherkha, and S. Chikhi, "A study of LoRaWAN protocol performance for IoT applications in smart agriculture," *Computer Communications*, vol. 164, pp. 148–157, Oct. 2020.

- [8] A. Pagano, D. Croce, I. Tinnirello, and G. Vitale, "A Survey on LoRa for Smart Agriculture: Current Trends and Future Perspectives," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3664–3679, 2023.
- [9] S. Sendra, L. García, J. Lloret, I. Bosch, and R. Vega-Rodríguez, "LoRaWAN Network for Fire Monitoring in Rural Environments," *Electronics*, vol. 9, no. 3, 2020, [Online]. Available: <https://www.mdpi.com/2079-9292/9/3/531>.
- [10] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, and A. F. Skarmeta, "Performance Evaluation of LoRa Considering Scenario Conditions," *Sensors*, vol. 18, no. 3, 2018, [Online]. Available: <https://www.mdpi.com/1424-8220/18/3/772>.
- [11] V. Venkateswaran and I. O. Kennedy, "How to sleep, control and transfer data in an energy constrained wireless sensor network," in *Proc. 2013 51st Annual Allerton Conf. Communication, Control, and Computing (Allerton)*, 2013, pp. 839–846.
- [12] Q. Li, S. P. Gochhayat, M. Conti, and F. Liu, "EnergIoT: A solution to improve network lifetime of IoT devices," *Pervasive and Mobile Computing*, vol. 42, pp. 124–133, 2017.
- [13] D. Bandur, B. Jakšić, M. Bandur, and S. Jović, "An analysis of energy efficiency in Wireless Sensor Networks (WSNs) applied in smart agriculture," *Computers and Electronics in Agriculture*, vol. 156, pp. 500–507, 2019.
- [14] S. Verma, N. Chug, and D. V. Gadre, "Wireless Sensor Network for Crop Field Monitoring," in *Proc. 2010 Int. Conf. Recent Trends in Information, Telecommunication and Computing*, 2010, pp. 207–211.
- [15] G. Sahitya, N. Balaji, C. Naidu, and S. Abinaya, "Designing a Wireless Sensor Network for Precision Agriculture Using Zigbee," in *Proc. 2017 IEEE 7th Int. Advance Computing Conf. (IACC)*, 2017, pp. 287–291.
- [16] R. L. Rosa, C. Dehollain, M. Costanza, A. Speciale, F. Viola, and P. Livreri, "A Battery-Free Wireless Smart Sensor platform with Bluetooth Low Energy

- Connectivity for Smart Agriculture,” in *Proc. 2022 IEEE 21st Mediterranean Electrotechnical Conf. (MELECON)*, 2022, pp. 554–558.
- [17] T. Wasson, T. Choudhury, S. Sharma, and P. Kumar, “Integration of RFID and sensor in agriculture using IOT,” in *Proc. 2017 Int. Conf. Smart Technologies For Smart Nation (SmartTechCon)*, 2017, pp. 217–222.
 - [18] S. Yamazaki and Y. Nakajima, “A Sigfox Energy Consumption Model via Field Trial: Case of Smart Agriculture,” *IEEE Access*, vol. 11, pp. 145320–145330, 2023.
 - [19] M. M. Blacio, V. G. Santos, D. J. Chamba, W. T. Guin, and L. C. Jiménez, “Empowering Low-Power Wide-Area Networks: Unlocking the Potential of Sigfox in Local Transmission,” in T. Guarda, F. Portela, and J. M. Diaz-Nafria, Eds., *Advanced Research in Technologies, Information, Innovation and Sustainability*, Cham, Switzerland: Springer Nature, 2024, pp. 417–429.
 - [20] G. Valecce, P. Petruzzi, S. Strazzella, and L. A. Grieco, “NB-IoT for Smart Agriculture: Experiments from the Field,” in *Proc. 2020 7th Int. Conf. Control, Decision and Information Technologies (CoDIT)*, vol. 1, 2020, pp. 71–75.
 - [21] J. Huan, H. Li, F. Wu, and W. Cao, “Design of water quality monitoring system for aquaculture ponds based on NB-IoT,” *Aquacultural Engineering*, vol. 90, p. 102088, 2020, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0144860919301189>.
 - [22] X. Cao and Y. Li, “Data Collection and Network Architecture Analysis in Internet of Vehicles Based on NB-IoT,” in *Proc. 2018 Int. Conf. Intelligent Transportation, Big Data Smart City (ICITBS)*, 2018, pp. 157–160.
 - [23] R. Ratasuk, N. Mangalvedhe, Y. Zhang, M. Robert, and J.-P. Koskinen, “Overview of narrowband IoT in LTE Rel-13,” in *Proc. 2016 IEEE Conf. Standards for Communications and Networking (CSCN)*, 2016, pp. 1–7.

- [24] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, and S. Roundy, "PicoRadio supports ad hoc ultra-low power wireless networking," *Computer*, vol. 33, no. 7, pp. 42–48, 2000.
- [25] R. Sharma, S. Prakash, and P. Roy, "Methodology, Applications, and Challenges of WSN-IoT," in *Proc. 2020 Int. Conf. Electrical and Electronics Engineering (ICE3)*, 2020, pp. 502–507.
- [26] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [27] M. F. Othman and K. Shazali, "Wireless sensor network applications: A study in environment monitoring system," *Procedia Engineering*, vol. 41, pp. 1204–1210, 2012.
- [28] E. Taparci, K. Olcay, M. O. Akmandor, B. Kabakulak, B. Sarioglu, and Y. D. Gokdel, "A Mathematical Programming Approach for IoT-Enabled, Energy-Efficient Heterogeneous Wireless Sensor Network Design and Implementation," *Sensors*, vol. 24, no. 5, 2024, [Online]. Available: <https://www.mdpi.com/1424-8220/24/5/1457>.
- [29] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *Proc. 2016 23rd Int. Conf. Telecommunications (ICT)*, 2016, pp. 1–6.
- [30] I. da S. Batalha, A. V. R. Lopes, W. G. Lima, Y. H. S. Barbosa, M. C. De Alcântara Neto, F. J. B. Barros, and G. P. S. Cavalcante, "Large-Scale Modeling and Analysis of Uplink and Downlink Channels for LoRa Technology in Suburban Environments," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24477–24491, 2022.
- [31] J. R. Cotrim and J. H. Kleinschmidt, "LoRaWAN Mesh Networks: A Review and Classification of Multihop Communication," *Sensors*, vol. 20, no. 15, p. 4273, Jul. 2020, doi: 10.3390/s20154273. Available: <https://doi.org/10.3390/s20154273>.

- [32] A. Zourmand, A. L. Kun Hing, C. Wai Hung, and M. AbdulRehman, "Internet of Things (IoT) using LoRa technology," in *Proc. 2019 IEEE Int. Conf. Automatic Control and Intelligent Systems (I2CACIS)*, 2019, pp. 324–330.
- [33] K. Olcay, E. Taparci, M. O. Akmandor, B. Kabakulak, B. Sarioglu, and Y. D. Gokdel, "Modeling and Implementation of an Adaptive Wireless Sensor Network for Low Power IoT Applications," in *Proc. 2023 8th Int. Conf. Smart and Sustainable Technologies (SpliTech)*, 2023, pp. 1–4.
- [34] "E32-433T30D User Manual," v1.9, Chengdu Ebyte Electronic Technology Co., Ltd., 2019.
- [35] Arduino, "EEPROM - Arduino Docs," 2024. [Online]. Available: <https://docs.arduino.cc/learn/built-in-libraries/eeprom/>. [Accessed: 24- Jul- 2024].
- [36] Y.D.Gökdel, " *Towards Sustainable Cities: Smart Parking System and Policy Development in Istanbul* ", RDI Project Proposal, İstanbul Bilgi University, Electrical and Electronics Engineering Department, 2021.

APPENDICES

Appendix A. First Sensor Node Arduino Code

```
#include "Arduino.h"
#define E32_TTL_100

#include "LoRa_E32.h"
#include "LowPower.h"

/*-----( Declare Constants and Pin Numbers )-----*/

/*-----( Declare objects )-----*/
SoftwareSerial ec_sensor(8, 9); // RX, TX

SoftwareSerial mySerial(3, 4); // e32 TX e32 RX
LoRa_E32 e32ttl100(&mySerial, 2, 6, 5);
const int wakeUpPin = 2;

void printParameters(struct Configuration configuration) ;
#define sensorPower 7
#define sensorPin A0

const byte ec[] = {0x01, 0x03, 0x00, 0x00, 0x00, 0x04, 0x44, 0x09};

byte values[13];

void wakeUp()
{
```

```

digitalWrite(7, HIGH);

pinMode(10, OUTPUT);

//Serial.println("HI!");

// Startup all pins and UART
e32ttl100.begin();

ResponseStructContainer c;
c = e32ttl100.getConfiguration();
Configuration configuration = *(Configuration*) c.data;
configuration.ADDL = 3;
configuration.ADDH = 0;
configuration.CHAN = 0x04;
configuration.OPTION.fixedTransmission = FT_TRANSPARENT_TRANSMISSION;
configuration.OPTION.wirelessWakeupTime = WAKE_UP_2000;

configuration.OPTION.fec = FEC_1_ON;
configuration.OPTION.ioDriveMode = IO_D_MODE_PUSH_PULLS_PULL_UPS;
configuration.OPTION.transmissionPower = POWER_20;

configuration.SPED.airDataRate = AIR_DATA_RATE_010_24;
configuration.SPED.uartBaudRate = UART_BPS_9600;
configuration.SPED.uartParity = MODE_00_8N1;
}

void ec_setup() {
    ec_sensor.init(8, 9);
    ec_sensor.begin(9600);
    /* ec_sensor.setRX(8);
    ec_sensor.setTX(9); */
    //ec_sensor.listen();
    digitalWrite(7, LOW);

    // Define pin modes for TX and RX
}

void ec_sleep() {
    // ec_sensor.end();
    digitalWrite(10, LOW);
    digitalWrite(7, HIGH);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
}

void setup() {
    Serial.begin(9600);
    // delay(500);
    pinMode(7, OUTPUT);
    // Initially keep the sensor OFF

```

```

e32ttl100.setConfiguration(configuration, WRITE_CFG_PWR_DWN_SAVE);
printParameters(configuration);
c.close();

ResponseStatus rs = e32ttl100.sendMessage("Hello, world?"); // OK The message is received on the other device
//// Check If there is some problem of succesfully send
Serial.println(rs.getResponseDescription());
delay(100);

}

void loop() {

    e32ttl100.setMode(MODE_2_POWER_SAVING);
    delay(100);
    attachInterrupt(0, wakeUp, LOW);
    LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF);

    detachInterrupt(0);

    e32ttl100.setMode(MODE_0_NORMAL);
    delay(2500);

    Serial.println("WAKE!");
    // Disable external pin interrupt on wake up pin.

```

```

    ec_setup();

    delay(1000);
    digitalWrite(10, HIGH);

    delay(10);

    ec_sensor.write(ec, sizeof(ec));
    ec_sensor.flush();

    digitalWrite(10, LOW);

    for (byte i = 0; i < 14; i++)
    {
        values[i] = ec_sensor.read();
        // //Serial.print(values[i], HEX);
        // //Serial.print(" ");
    }
    // //Serial.println();

    delay(100);

    float soil_temp = (int(values[5] << 8 | values[6])) / 100.0f;

    int soil_ec = (int(values[7] << 8 | values[8]));

    float soil_ms = (int(values[3] << 8 | values[4])) / 100.0f;

    int soil_sal = (int(values[9] << 8 | values[10]));

    // delay(1000);
    String my1 = "F:" + String(soil_temp) + "O:" + String(soil_ms)+"S:" + String(soil_sal)+"E:" + String(soil_ec) + "\n";
    e32ttl100.sendMessage( my1 ); // OK The message is received on the other device
    /// Check If there is some problem of succesfully send
    // //Serial.println(rs.getResponseDescription());*/
    Serial.println(my1);
    // delay(2000);

    ec_sleep();
    //
    delay(2500); // Take a reading every second for testing
}

// This function returns the analog soil moisture measurement
int readSensor() {
    digitalWrite(sensorPower, HIGH); // Turn the sensor ON
    delay(2000); // Allow power to settle
    int val = analogRead(sensorPin); // Read the analog value form sensor
    digitalWrite(sensorPower, LOW); // Turn the sensor OFF
    return val; // Return analog moisture value
}

```

Appendix B. Gateway Python Code

```
import serial
import time

port = 'com3' # Replace with the correct serial port
baud_rate = 9600 # Replace with the correct baud rate

ser = serial.Serial(port, baud_rate)

message = "" # Declare the variable outside the loop
timer_start_listen = 0
timer_start_transmit = 0
|

1 usage
def get_values():
    ser.write(b"GET_Values")
    print("Message sent to node. Waiting for reply. 20 secs.")

try:
    while True:
        size = ser.inWaiting()
        if size:
            data = ser.read(size)
```

```

print(res)
output1 = res.strip()

if output1.startswith("H:") and "T:" in output1:
    values1 = output1.split(" ")
    print("Output1:", output1)
    print("Values1:", values1)

elif output1.startswith("INDEX"):
    values1 = output1.split(" ")
    print("Output1:", output1)
    print("Values1:", values1)

else:
    if time.perf_counter() - timer_start_transmit >= 60:
        user_input = "t\r";
        ser.write(user_input.encode())
        timer_start_transmit = time.perf_counter()

```

```

if time.perf_counter() - timer_start_listen >= 20:
    #user_input = input("Enter 't' or 'h' or 'GET_Values': ")

    if user_input == "GET_Values":
        get_values()
    else:
        ser.write(user_input.encode())
        print("Message sent to node. Waiting for reply. 20 secs.")
        timer_start_listen = time.perf_counter()

except KeyboardInterrupt:
    print("Interrupt")
    # ser.close()

```