Deggendorf Institute of Technology

Faculty of Electrical Engineering and Media Technology

“Examination and Verification of Wireless Communication Protocols for Industry 4.0 Applications and Sensor Networks”

Master’s thesis in fulfillment of the requirements for the degree of:

Master of Science (M.Sc.) Deggendorf Institute of Technology

Submitted by:

Hassan, Rohaib

00775997

On: 15 March 2023

Examiner:

Prof. Dr.-Ing. Werner Bogner

DEGGENDORF INSTITUTE OF TECHNOLOGY

Author's Declaration

Name of student: Rohaib Hassan

Professor/Supervisor at DIT: Prof. Dr. Werner Bogner

If external: Name of supervisor:

Master's thesis topic:

Examination and verification of wireless communication protocols for Industry 4.0 applications and sensor networks

1. I hereby declare that this Master's thesis is my own work, has not been submitted for any other degree at any other university or institution, does not contain or use any sources or resources other than those referenced, and that all direct and paraphrased quotes have been duly cited as such.

Deggendorf, 15.03.2023 Rohaib

(Date) (Signature student)

2. I hereby consent to my completed Master's thesis being made available to a broader public through the library of the Deggendorf Institute of Technology.

☒ Yes

☐ No

If yes:

I certify and warrant that I am the sole owner and holder of all rights to the Master's thesis, including the right to use the tables, figures, illustrations, plans, or similar, submitted therein and that no third-party rights or claims, nor any statutory provisions, will be infringed as a result of its publication.

Deggendorf 15.03.2023 Rohaib

(Date) (Signature student)

Only to be completed by the examiner if the author consents to the Master's thesis being published.

Collection of a copy of the Master's thesis by the library, including the lending of the copy, is hereby

☐ endorsed

☐ not endorsed.

Deggendorf

(Date) (Signature examiner)

Contents

Abstract

Technology is the powerful growling force behind change, and the Internet of Things (IoT) is the foundation of such revolutionary engines. Because of sensors, good power sources, and internet connectivity, the Industrial Internet of Things (IIoT) is a field that is possible in the real world. A very lightweight, reasonably priced, and low bandwidth protocol is necessary for a technology that is developing so quickly, such as the MQTTmqtt Protocol.It is simple for clients to publish and subscribe to the desired topic through the host serving as the network's server, also known as the broker, due to the lack of a well-established protocol. The research to improve this power consumption requires a lot of study of the behavior of different protocols of this procedure. This thesis serves the purpose of aiding the research. This thesis describes issues with the power consumption of various modules, sends data over the internet without interfering with others, and allows for online system monitoring. In this thesis, we also investigated how to increase the device's power consumption as the temperature is received after a certain interval, so that the device will operate in a different mode when no data has been received during this interval. LoRa is a low-power, long-range radio communication technology that was created for the Industrial Internet of Things. These devices are frequently installed in remote locations where one or more gateways may not be able to provide end-to-end connectivity. In this thesis, communication is made easier to access and more dependable by using an MQTT lens extension between the low-power ESP8266 WiFi client and the clients on smartphones and laptops. The WiFi-enabled ESP8266 board interfaces with Temperature sensor to monitor the temperature. In the future, devices will communicate over long distances with each other using the LoRaWAN Protocol. The device's low power consumption and capability to send data over long distances should be our main concerns. LoRaWAN is, therefore, capable and secure.A detailed explanation of how the mioty is connected to the TTN is discussed in this thesis.In the future, if we need more reliability and a longer range, MIOTY might be a better choice. LoRaWAN might be a more cost-effective choice, though, if we need to support more devices but don't need quite as much range.

List of Abbreveations

Acknowledgements

I would like to express my deepest gratitude to all my professors for their guidance, encouragement, and unwavering support throughout my master's program. Their expertise and dedication have been invaluable in shaping my research and academic growth.

I am also grateful to professor Dr.-Ing. Werner Bogner and my supervisor Michael Benisch for their valuable feedback and constructive criticism that has helped me improve my work. Their mentorship and encouragement have been critical to my success.

I want to express my heartfelt thanks to my mother for their unconditional love and unwavering support and for instilling in me the values of hard work and perseverance. Without their sacrifices and encouragement, I would not have made it this far.

Finally, I want to thank my siblings for always being there for me when I needed them and for their encouragement and support. Their unwavering belief in me has been a great source of strength and motivation throughout this journey. I am deeply grateful to all of you for your contributions to my academic and personal growth.

1 Introduction and Problem Definition

1.1 General Introduction 4.0

Industry 4.0 involves several enabling technologies (e.g., machine-to-machine communications, real-time industrial communications, big data analytics etc.) to design, monitor and improve industrial processes for more efficient and more sustainable production. In this context, wireless technologies play an important role. Industry 4.0 wireless applications typically require to cover large areas and transmit a small amount of data per node. Consequently, while bandwidth is not the main concern [20], reliability, bounded latency for real-time flows, and energy efficiency are the key performance indicators [22], LPWA**lpwa!** (**lpwa!**) networks have recently gained popularity because they provide access to low-power devices dispersed across very broad geographic areas. In a number of applications, such as smart cities, personal IIoTIndustrial Internet of Things (IIoT) applications, agriculture, and vehicular communications [16], [17], [13], LPWAN networks represent a novel communication paradigm that will complement traditional cellular and short-range wireless technologies. By eliminating the need for multi-hop communications protocols, LPWA would lower transmission latency and network overhead. Today's LPWA**lpwa!** technologies compete with one another and use a variety of methods to accomplish long-range, low-power operations and great scalability. In this thesis, the LoRa Alliance's long-range wireless communications system is examined. Long-lasting battery-powered devices are the target market for LoRa, which stands for "Long Range," where energy efficiency is of utmost importance. LoRaWAN, a standardized MAC layer over LoRa, is now responsible for defining the medium control access for LoRa. LoRaWANLong Range Wide Area Network (LoRaWAN) cannot enable real-time industrial applications because it was built for intermittent, non-time-constrained communications among a lot of nodes. With the most recent advancements in wireless sensor networkswireless sensor network (WSN) is now less expensive and energy-hungry, making WSNWSN more practical and simpler to use on a broad scale. There are many benefits if we compare wireless sensor networks and Industry 4.0 with wired ones, then there are a number of benefits, including the following benefits are available[34].

- capacity to monitor dangerous areas
- more adaptability and flexibility
- easier system expansion
- higher factory productivity
- better completed product quality with less scrap and wasted material [28]

1.2 1st Industrial Revolution

The development of new technologies like steam power, the expansion of factories, and the expansion of the railway system were all hallmarks of the First Industrial Revolution, which took place between the late 18th and early 19th centuries. It began predominantly in Britain and afterwards extended to other European and North American nations, as well as the rest of the world. A switch from manual labor to machine-based production, as well as significantly higher economic development and productivity, were all results of the revolution, which also brought about substantial improvements in agriculture, industry, transportation, and communication[36].

1.3 Second Industrial Revolution

The Technological Revolution, often known as the Second Industrial Revolution, occurred between the middle of the 19th and the beginning of the 20th century. It was distinguished by the widespread use of electric power, the development of new communication technologies like the telegraph and telephone, and the advent of new transportation systems like the car and airplane. The development of large-scale manufacturing and assembly line methods during the Second Industrial Revolution also allowed for the more affordable mass manufacture of items. The revolution had a significant impact on society, altering how people interacted with one another and with one another environment. It also helped to set the groundwork for the current industrialized World[36].

1.4 3rd Industrial Revolution

The transition from mechanical and analogue electronic technologies to digital electronics, which started in the late 20th century and is still going strong today, is referred to as the third industrial revolution, or the digital revolution. The creation and widespread application of computers, the internet, and other digital technologies define it. Our manner of living, working, and communicating has changed as a result of the Third Industrial Revolution, which has also resulted in the development of a global digital economy and a connected, interdependent society. With the rise of automation, artificial intelligence, and the Industry Internet of ThingsIIoT, it has also resulted in significant changes in the production and distribution of goods and services[36].

1.5 The Fourth Industrial Revolution

Fourth industrial revolution is also same as the third revolution as it has personal computer, mobile phones and internet. However this technology has taken to the next level as everything is moving towards the automation, interconnected and communication between the machines, devices and humans[21]. It is a true industrial revolution. The internet has a much wider impact than other revolutions because it connects the physical and digital worlds

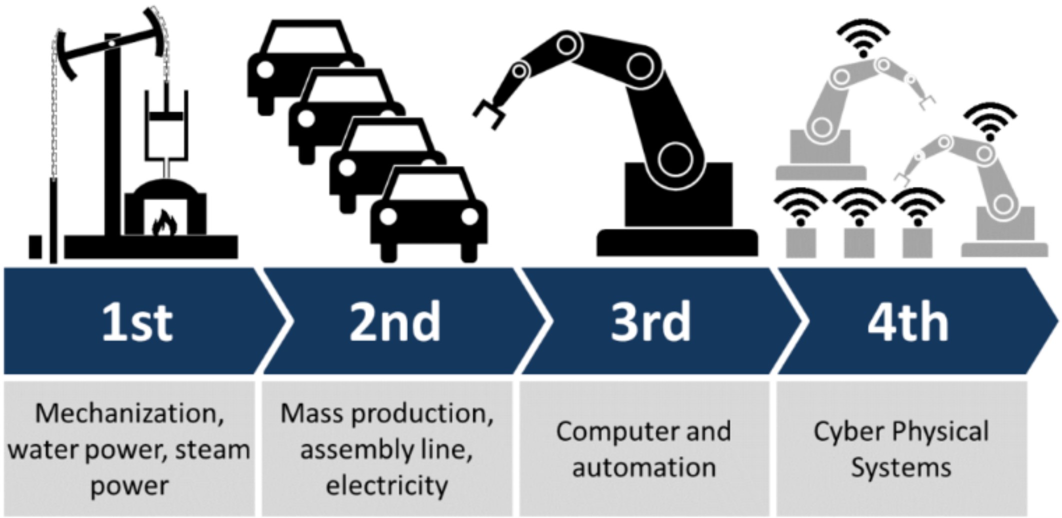

Figure 1.1: Industrial Revolutions [3]

as well as digital and physical systems, blurring boundaries and reducing the visibility of dividing lines[24]. Other revolutions have been linked to a single primary driver(steam power, mass production and personal computers)[29].Here are the some advantage that this industry has bring in the modern life and make the life easier.Healthcare, manufacturing, transportation, and finance are just a few of the sectors that the Fourth Industrial Revolution is revolutionizing. Although it has the potential to spur economic growth, produce new opportunities, and enhance quality of life, it also raises issues related to job displacement, personal privacy, and the effects on society as a whole[21].

- IIoTIIoT devices, such as sensors and smart devices, may gather and transmit data, making it possible to monitor and control processes in real time.
- To protect against cyber attacks and data breaches, cyber security is becoming more and more crucial as gadgets become more linked.

An industrial setting such as a factory, warehouse, or supply chain uses a network of linked machines and devices known as the Industrial Internet of Things(IIoT)IIoT. These gadgets have sensors, software, and other technologies built in that let them gather and share data with other gadgets, systems, and people. This presents countless prospects in a variety of industries, including heavy manufacturing, automotive, vital societal services, connected homes, and relatively straightforward end-user consumer goods. These potential draw both novice start-ups and well-established businesses looking to diversify their product offerings[10]. The sentences that follow should help the reader understand the opportunities that the Internet of Things presents.

2 State of the Art of IIoT

The state of the art of IIoTIIoT is constantly evolving, driven by advancements in technology, as well as changes in industrial practices and business models.

2.1 Introduction to Network for IoT

This section's main objective is to study the different protocols and sensors commonly used in the market. In the past, several network protocols were created to assist the data transport for IoT devices. The maximum data capacity, range, number of supported devices, availability of the technology across various geography, scalability, or cost to implement them on the market are varied. The following list summarizes some of the most commonly available technology and their advantages and disadvantages. There are different techniques available in market

- Bluetooth Low EnergyBluetooth Low Energy (BLE) is a Low Effort Bluetooth, commonly referred to as Bluetooth Low Energy, Bluetooth Smart is a wireless technology for communication that uses little power and is inexpensive. It was first introduced in 2010 as a component of the Bluetooth 4.0 protocol, and wearable technology and the IIoT(Industrial Internet of Things) make extensive use of it. The 2.4 GHz ISM (Industrial, Scientific, and Medical) band is where it operates. It is intended to offer low power, short range communication between devices. It is superior to traditional Bluetooth in that it is designed for brief data bursts, which makes it perfect for devices that need to run on battery power for extended periods of time. Smartwatches, heart rate monitors, fitness trackers and many other gadgets are examples of BLEBLE enabled devices[31].
- The French business Sigfox offers an international network for IIoTIIoT gadgets. It runs an low power wide area networklow Power Wide Area Network (LPWAN) which lets devices to communicate over great distances with relatively little power. This network is made to offer inexpensive, dependable and secure connectivity for Industrial Internet of ThingsIIoT gadgets, making it perfect for application in sectors like agriculture, logistics, and energy. Over 60 nations are covered by the this network, which connects millions of devices. The foundation of this technology is UNBUltra Narrow Band (UNB) radio communication, which enables low-power transmission of sparse data over extensive distances. Additionally the business sells cloud based software for monitoring and managing data sent across its network by connected devices[6],
- Weightless is an IIoTIIoT LPWAN**lpwa!** technology. It is an open standard for M2Mmachine to machine (MTM) communication and was created to give IIoTIIoT devices access to

low cost, low power, and long range communication. It transmits data over a narrowband radio frequency while using unlicensed airwaves. It is appropriate for usage in sectors like agriculture, logistics, and energy because it is built to accommodate numerous linked devices. This is an LPWANLPWAN technology that is independent of any one manufacturer, network operator, or service provider, unlike other LPWANLPWAN technologies. The Weightless SIGSpecial Interest Group (SIG) an industry partnership of businesses and organizations working together to advance the creation and upkeep of the Weightless standard, maintains the weightless standard[5].

- Weightless(N) is also used broadly and this will also work on Sub GHz.The transfer data rate fir this technology is 100 byte per second(bps) which is pretty slow, The range for this technology is 5km in Urban which will be higher if this will used in Rural areas, The packet size for this technology is Max 20 byte which is suitable for new technologies,Security for this protocol is fully addressed also[12].

- Weightless (P) Max data rate for this is 100 kbps which is pretty fast as compared to other ones, The range for this is 2km which is less from sigfox but the packet size of that technology is Min 10B.

- Ingenu is also another Sensor which is working on 2,4 GHz. Data rate for this is 19kbps/MHz. The cover area for this sensor is too much and it is mostly be like is 15 km which is pretty too much, but the size of that packet is max 10 KB, Topology used for this sensor is Star and also Tree Topology, the security for this sensor is fully addressed also[15].

- A UK-based technology business called Telensa offers IIoTIIoT solutions for the market of smart cities. The business focuses in wireless sensing and control technologies for urban settings such as intelligent parking and street lighting. The devices from Telensa are intended to give cities real time information and control capabilities, enhancing citizen quality of life while also boosting efficiency and cutting expenses.LPWANLPWAN which enables the deployment of numerous linked devices throughout a city, serves as the foundation of the company's main technology. In the market for smart cities, Telensa is regarded as a pioneer and has worked on numerous high-profile projects all around the world[35].

- DASH7 is a low-power wireless sensor networking technology that is designed to provide long range communication, ultra low power consumption and robust performance in harsh environments. It uses the ISO/IEC 18000-7 standard for active RFID (radio frequency identification) and has a range of up to several kilometers in line-of-sight environments.DASH7 technology is ideal for applications that require battery-operated, low-power, long-range communication[33]. It is often used in industrial and commercial settings for applications such as asset tracking, inventory management, and remote monitoring.

- Direct Sequence Spread SpectrumDirect Sequence Spread Spectrum (DSSS) is a wireless communication technology that uses a spread spectrum modulation technique to transmit data. It spreads the signal over a wide frequency band, which makes it more resistant to interference and jamming[37].When it comes to choosing between DSSSDSSS and LoRaWANLoRaWAN, It really depends on the specific application and requirements. it may be more suitable for applications that require high data rates and low latency, while LoRaWAN may be a better fit for applications that require long range coverage and low power consumption.

- Compared to more established WiFi standards like IEEE 802.11b/g/n/ac, IEEE P802.11ah employs the sub-1 GHz frequency band, more especially the 900 MHz band, to offer a longer range and better penetration through walls and other obstructions. It is made for IIoTIIoT applications that need long-range communication and low power consumption. It is also known as WiFiWireless Fidelity (WiFi). The IEEE P802.11ah standard can operate over distances of up to several kilometers and enables a range of data rates up to 347 Mbps. It is appropriate for applications like smart cities, industrial automation and home automation because it is built to support a huge number of devices. However, one of the disadvantages of IEEE P802.11ah is its lower data rate compared to other WiFiWiFi standards. It is also a relatively new standard and may not be as widely supported as other WiFiWiFi standards.

- A wireless communication standard called LTE-MTCMachine-Type Communication (MTC) was created especially for IIoTIIoT applications. It is a low power, wide area network (LPWAN) technology that makes use of current cellular networks to give IIoTIIoT devices long-range connectivity and low power usage.One of the advantages of LTE-MTCMTC is its compatibility with existing cellular networks, which means that it can leverage the existing infrastructure to provide widespread coverage and connectivity for IIoTIIoT devices. It also provides reliable and secure communication, making it suitable for applications such as remote monitoring, asset tracking, and smart metering.However, one of the disadvantages of LTE-MTCMTC is its higher power consumption compared to some other LPWAN technologies, which can limit battery life for IIoTIIoT devices. It also requires a cellular subscription and may be subject to network coverage limitations in some areas[?].

2.2 LoRaWAN

A LoRaLong Range (LoRa) Sensor comprises several of Lora sensors connected to an Arduino or Raspberry Pi microcontroller and a GPS module; hence they are called the Lora sensor. Therefore, LoRaWAN is made for long-range, low-data connections and low-power operations with sensors that use batteries with few options for charging. IIoTIIoT devices that run on batteries often endure for months before they run out of power. As a result, LPWANLPWAN provides a low-cost, low-consumption and low-energy substitute for 4G and 5G networks. This creates a new business model that allows for the user-friendly distribution of inexpensive IIoTIIoT devices without the requirement for continuous battery recharging. As in Fig[2.1], There is a Lora sensor which is used for measuring the Humidity, Temperature, Wind Speed and Carbon dioxide. After measuring all these values from this sensor, the gateway will receive this data, from there with the help of the gateway, data will be online, and it is easy to see the data from the TTN server or application server or Web Application in these, there are different node available, all the nodes send this data only on one gateway.

Uses of LoRa Sensor

We must establish a successful LoRa network in order to use a LoRa sensor. The following are elements of a typical LoRa network:

LoRa Node

When a LoRa sensor integrates with wireless connectivity and the LoRa protocol, it becomes a LoRa node. The relevant sensor data is gathered by a LoRa node and sent to the LoRa Gateway.

LoRa Gateway

A gateway is typically a device that connects various types of networks. To link the IP and LoRa wireless networks, we use the LoRa Gateway LG01. The following are some of its main traits:

- (1) It is the main point of the gateway. It connects both the Transmit side and the Receiver side.
- (2) Receive the data from the LORA sensor and send this data through IoT through IP address via;
- (3) WIFI
- (4) Ethernet
- (5) 4G/5G
- (6) So now the receiver sends the command to LORA to Receive.

The LoRa sensor network architecture consists of three main components Fig 2.1.

- LoRa sensors: These compact, low-power gadgets are used to gather information from numerous sensors, including temperature, humidity, wind speed, and carbon dioxide sensors. The LoRaWANLoRaWAN protocol is used by LoRa sensors to transfer data to LoRa gateways. As we have different nodes available, these sensors send this data by any public network.
- LoRa gateways transfer data from LoRa sensors to the application server or the cloud. As shown in the figure 2.1, we have several choices. Typically, LoRa gateways connect to the internet through Wi-Fi or Ethernet, and they communicate with the sensors using the LoRaWAN protocol.

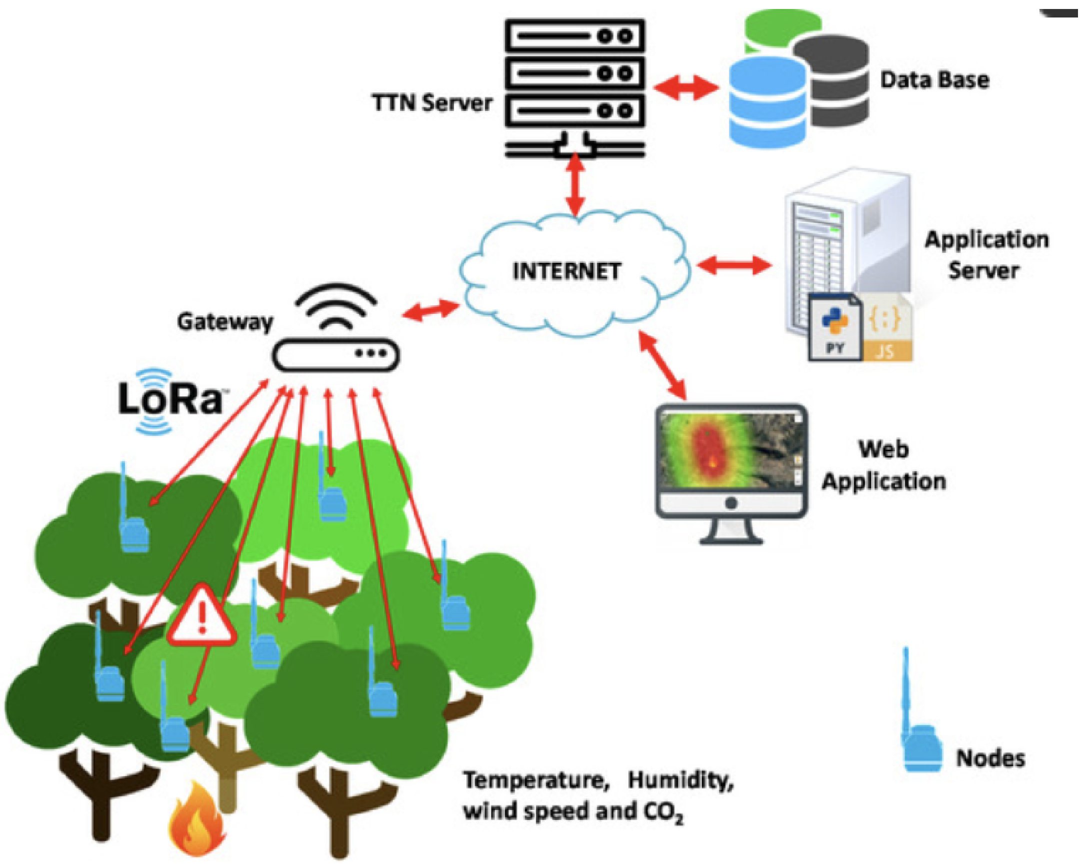

Figure 2.1: Network architecture of Lora Sensor [30]

- The data gathered by the LoRaLoRa sensors is stored in this central database, which can also be referred to as an application server or cloud. The processing and analysis of the data are handled by the cloud or application server, which also provides end users with insights. If you want to see the data live, then it's better that we use TTN or a Web application.

The LoRa sensor network architecture is designed to be scalable and flexible, allowing organizations to deploy and manage large-scale sensor networks quickly. In addition, the use of low-power, long-range radio transmissions enables LoRa sensors to operate for extended periods of time without requiring frequent battery replacement, making them ideal for use in remote and hard-to-reach locations. Overall, the LoRa sensor network architecture is designed to enable organizations to collect and analyze data from a wide range of sources, providing valuable insights that can be used to optimize operations and improve decision-making. In order to achieve low power consumption for M2MMTM applications, current technology developments necessitate the introduction of new protocols and long-distance communication architectures. These limitations imposed by wireless sensor networks include long distances, high speed, and low power consumption.Pre-built network nodes and wireless modules are available in the industry and can be incorporated into individual architectures. We will discuss the solutions that fall into these two categories separately in the sections that follow:

2.3 Data Transmission Modules

Different wireless technology types pertinent to and related to the IIoTIIoT sector have coverage areas ranging from a few millimetres to tens of kilometres. Depending on the transmission distance, we can utilize Bluetooth, Wi-Fi, ZigBee, and 6LowPAN for short and medium distances and LoRA, Sigfox, or GSM for long distances. The distance coverage of several data transmission options is available which compare a number of low-power, long-range transmission techniques[4]. LoRaWAN stands out among the solutions offered by having the following qualities: high interference immunity. The maximum distance at which two devices can communicate without obstructions blocking the transmission path. The range of LOS communication depends on the height of the devices, the power of the transmitter, the sensitivity of the receiver, the frequency of the signal, mesh topology, lowest power consumption (calculated autonomy of up to 10 years), multiple frequency compatibility, and data rates up to 50 kbps (data rate comparable and adequate for most industrial equipment). Nemerus, Multiteh, and Microchip are the companies that offer the solutions. The Microchip module has the most power during transmission and reception compared to the modules offered by the other firms, but it also offers the widest range of communication. The maximum message size for the LoRaWAN protocol is 64 bytes, which is smaller than the maximum message sizes for ZigBee (102 bytes), 6LoWPAN (102 bytes), Low Power Wi-Fi (2312 bytes), and 6LoWPAN (102 bytes) [9]. The reduction of power consumption has been addressed using a variety of techniques, including hardware optimization and the application of adaptive algorithms for data rate, duty cycle, and transmission power. Several adaptive methods are evaluated in [11] to re-transmit data in order to assess the impact on power usage. The experiments were conducted with various sent data packet sizes and various pauses between two broadcasts. The study of the data showed that the sensor nodes can operate autonomously in monitoring applications for at least five years. [9] Depending on the transmission distance, we can utilize Bluetooth, Wi-Fi, ZigBee, and 6LowPAN for short and medium distances and LoRA, Sigfox, or GSM for long distances. The distance coverage of several sensors led to another analysis of a sensor node's power usage using LoRaWAN. Here, a 2400 mAh battery has a 1-year autonomy with a 5-minute transmission rate between two successive packets. In comparison, other protocols, such as Zigbee and Z-Wave, are designed for shorter-range communication and are typically used for home automation and similar applications. Cellular protocols such as LTE-M and NB-IoT are designed for cellular networks and are better suited for IIoT applications that require higher bandwidth and more secure communication.

Table 2.1 describes the summary of different protocols

	LoRaWAN	Sigfox	Ingenu	Telensa	Dash7
Band	868 MHZ	868 MHZ	2.4 GHZ	SUB-GHZ	SUB-GHZ
Max data-rate	50 Kbps	100bps	19Kbps/Mhz	346 Mbps	-
Range(urban)	5km	10km	15km	1km	3km
Packet size	Max 256 B	12 B	Max 10Kb	Max 65 kb	-
Downlink	Yes different plans	yes(not sym)	yes	yes	yes
Topology	Star of Star	Star	Star/Tree	Star/Tree	Star
Roaming	yes	yes	yes	yes	yes
Security	fully addressed	partially addressed	Fully addressed	in development	
Protocol ownership	partially proprietary	proprietary	proprietary	standard	proprietary

Table 2.1: Different protocol for Communication [23]

There are different protocols which are used in different ways. First of we have a different way, but the problem arises when there are too many senders and only one receiver at the time, so a lot of important information we have lost on the way, so for this, we need the Mioty protocol.

2.4 Mioty

In 2020, the Fraunhofer Institute formally unveiled Mioty, a low-power, wide-area networkLPWAN protocol for integrated circuits, to address the common drawbacks of the LPWANLPWAN products currently available on the market. This protocol offers the most scalable, reliable, and green options for industrial lot applications. This protocol uses a license-free spectrum in Europe, and the frequency is 868 MHz; if used in North America, the frequency is 915 MHz [1].

Mioty Technology

Designed for massive IIoT, MIOTY is a LPWAN solution based on ETSI 103 357. This achieves long-range with sub-1 GHz communication through a robust network that gives the market the most efficient wireless communication connectivity solution. This protocol's primary function is splitting the data into small packets and applying error-correcting codes, which can recover data losses up to 50 per cent. Compared to other LPWAN solutions, this protocol has the lowest packet error, making it particularly important in inclement weather or rural areas with signal issues. This protocol can prevent combining the two sensors if two signals are received simultaneously. It also has the ability to send data at staggered intervals across a wide range of frequency bands. The shorter the air time of the packet, the less data collision occurs, ensuring messages reach their destination. All the data which are transmitted through LPWAN will take more time as compared to other protocols. The data is protected and integrity-protected with AES-128, with keys provisioned on a per-device basis[1]. This will achieve unprecedented scalability with 10,0000 devices in a network and the ability to transmit 1,5 million data packets per day. However, Mioty is covering too much area, which is approximately 20 km line of sight range [1].

Description of the Mioty

It is a low-cost gateway with the mioty profiles EU 0, EU 1, and EU 2 for use in Europe, is bidirectional (it can receive and send data), and has these profiles. AVA is simple to set up. The local network can be reached by the gateway via Ethernet. A connection to a web interface that provides a summary of all the online data is made possible by the mioty protocol[32].

Table 2.2: Mioty Description [1]

Number	Features	Ranges
1	Input Frequency	868-870 MHz
2	Tx power	14 dBm
3	Sensitivity(EU1)	-135 dBm
4	Typical Input Current	approx. 900 mA
5	Dimensions	56* 90 * 25 mm
6	Weights	0.16kg
7	Temperature Range	0-35 (indoor use only)
8	Humidity range	20 - 80 percent

Installation and Usage of Mioty protocol

First of all determine the items that are necessary for this. Installing DHCP support is the first step in setting up the Mioty protocol, followed by attaching the antenna to the gateway. It has to connect the system to Arduino because that is how it has to be used. Once that is done, type "Ava" into the address bar to access the gateway's dashboard. In case of a connection failure, contact the administrator to find out how to access the network's gateway. Sometimes there may be a connection issue, in which case the gateway may only be reachable by IP address, which the network administrator can supply[1].

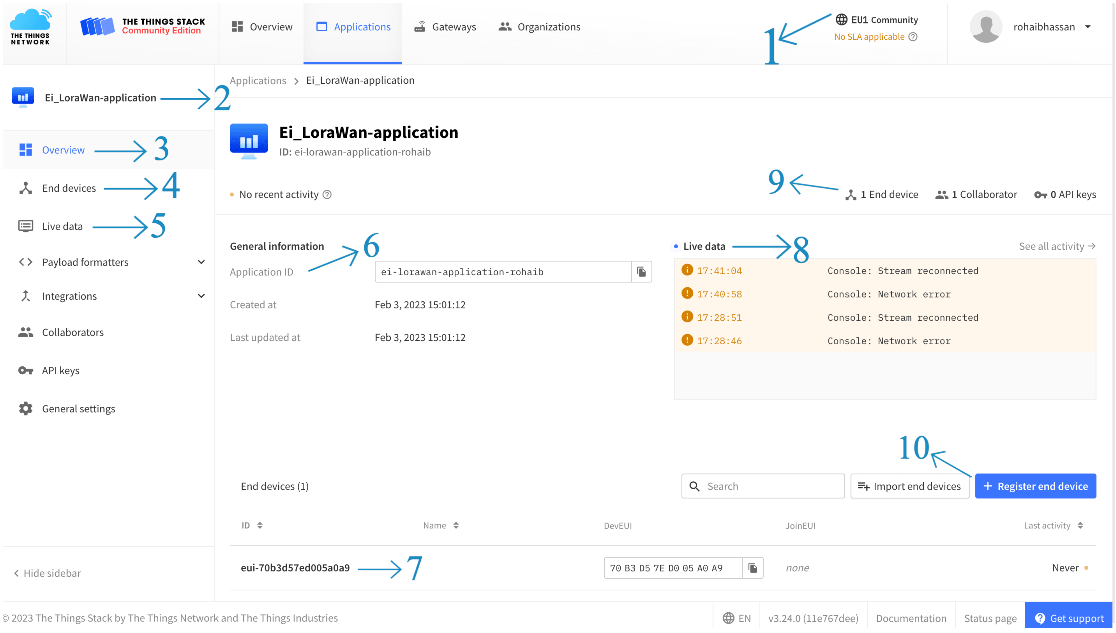

Figure 2.2: Dashboard TTN

- 1) point of the arrow in figure 2.2shows that it has used in Europe, that's why it shows the EU1 community—various communities in the TTN, including those in North America, Australia, and Europe.
- 2) It depends on which console was used because there are various consoles available on the TTN. As it is used for the application Fig 2.2.
- 3) Fig 2.2 A high-level description or summary of a particular feature or element of the TTN network, such as the architecture, the devices, the applications, the gateways, or the data flow, is referred to as an overview.
- 4) A device or sensor that is connected to the network and transmits data to a gateway is referred to as an end node Fig 2.2. End nodes can use a variety of communication protocols, such as LoRaWAN or Bluetooth, and can be powered by batteries or connected to a power source.

- 5) The TTN Dashboard provides a user interface that allows to view live data Fig 2.2 and monitor the performance of your devices and applications.

- 6) An application ID Fig 2.2 is a special identification code given to a TTN network-using application. The application ID is employed to distinguish between various applications that might be connected to the same network and to guarantee that data is sent to the right application.

- 7) EUIeui that's a number for the application that has connected with TTN.

- 8) The sending and receiving of data by the connecting device at this location are clearly visible. Here is a summary of the device Fig 2.2.

- 9) Since there is currently only one connected sensor Fig 2.2, the endpoint will only display one node. There is a number of connected nodes displayed.

- 10) in Fig 2.2 If there is a requirement for adding more applications, then the devices are registered quickly.

Mioty Connection with TTN

If the Above setting is fine, then, first of all, we have to register the sensor to the gateway. Open the AVA gateway after that, and fill out all the fields needed to write the sensor. This online search was conducted using TTN, a website where the Arduino was used Fig 2.3.

- These are some settings that have to do, Arrow 1 in Fig 2.3indicates that the device is in Europe, so it is using it in Europe, so the range is 863-870 MHz. All of these frequencies depend on the region in which the devices are being used, so if someone uses that device in the USA, it will be at 915 MHz.

- As in the Fig 2.3that LoRaWANlorawan version. so there are different versions available first of LoRaWANlorawan. Because the old version is less functional than the new version, it needs to know which version you are using. All of these things need to be considered because roaming was not an option in the previous version. LoRaWANlorawan has been updated, and as a result, it now has all of these features, including roaming, and no bugs. There are two activation modes in LoRaWAN.lorawan as shown in Fig 2.3: OTAAotta and ABPabp.

- LoRaWANlorawan uses OTAAotta to establish a secure connection between a LoRalora device and a LoRaWANlorawan network. OTAAotta is when a device requests to join a LoRaWANlorawan network by sending a join request message to the network server. The network server then responds with a join accept" statement that includes a set of session keys used to encrypt and decrypt messages between the device and the network server..

- A secure connection between a LoRalora device and a LoRaWANlorawan network can be made using ABPabp, a substitute for OTAAotaa. ABPabp does not require the device to send a join request to the network server in order to create a secure session, in contrast to OTAAotta.

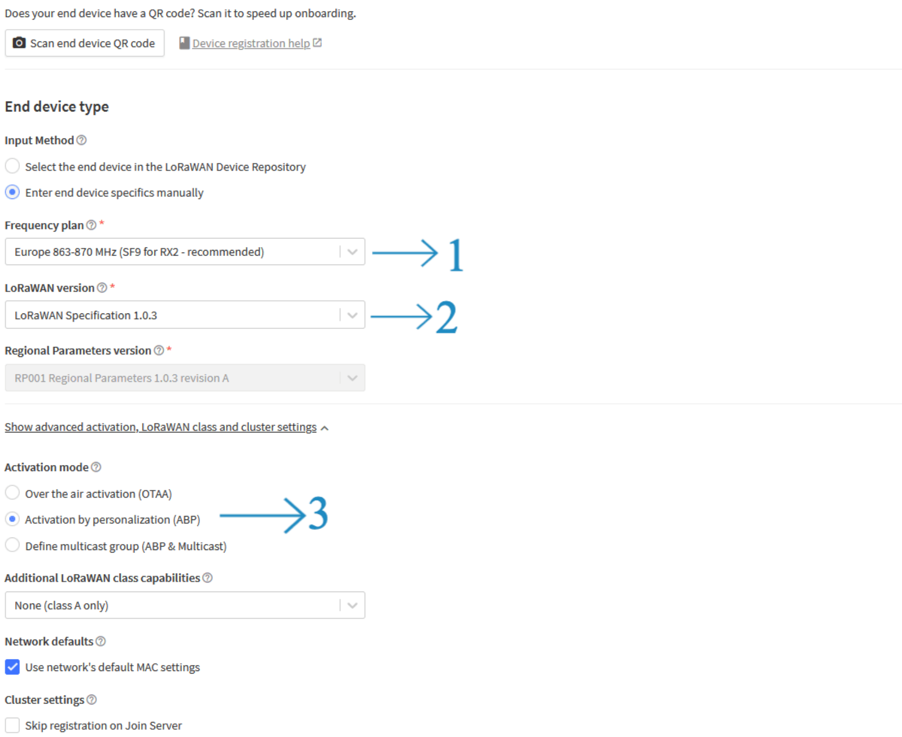

Figure 2.3: Setting of Things of Internet

- These sensors require a brief value in order for the counting to begin from left to right. Our short addresses would be the fifth and sixth bytes of the EUI. For instance, if you look at the previous point, the answer would be "d000."

- There is a Device address, so we can quickly access our device if we know our device address Fig 2.3.

- Fig 2.3 It has the same value as the network security key, so they have to change this number to a hexadecimal number and put it into our code in Arduino. So the only way to connect our TTN device is in that way.

- Application key is also at the bottom of Fig 2.3, so this value is used to access the application data.

- As we have also put the location of the device or gateway where to check the value, so here put the Deggendorf Fig 2.3, so it's up to you which one you want to set.

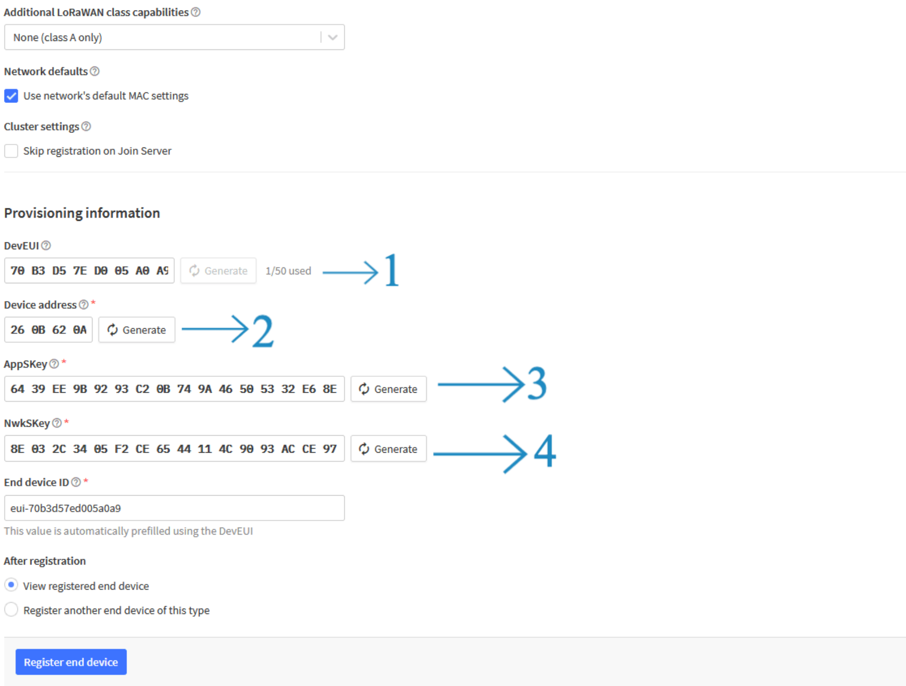

Figure 2.4: connection between Lora and TTN

- In the Fig 2.5 the dashboard of ttn. So there are different points available: if you have to check the live data of the device or if there is any message received, then get the notification. The location has been added to this column.

After entering all the sensor values, the Lora sensor is registered on the AVA Gateway. The sensor should appear at the end of the gateway.

 2 State of the Art of IIoT

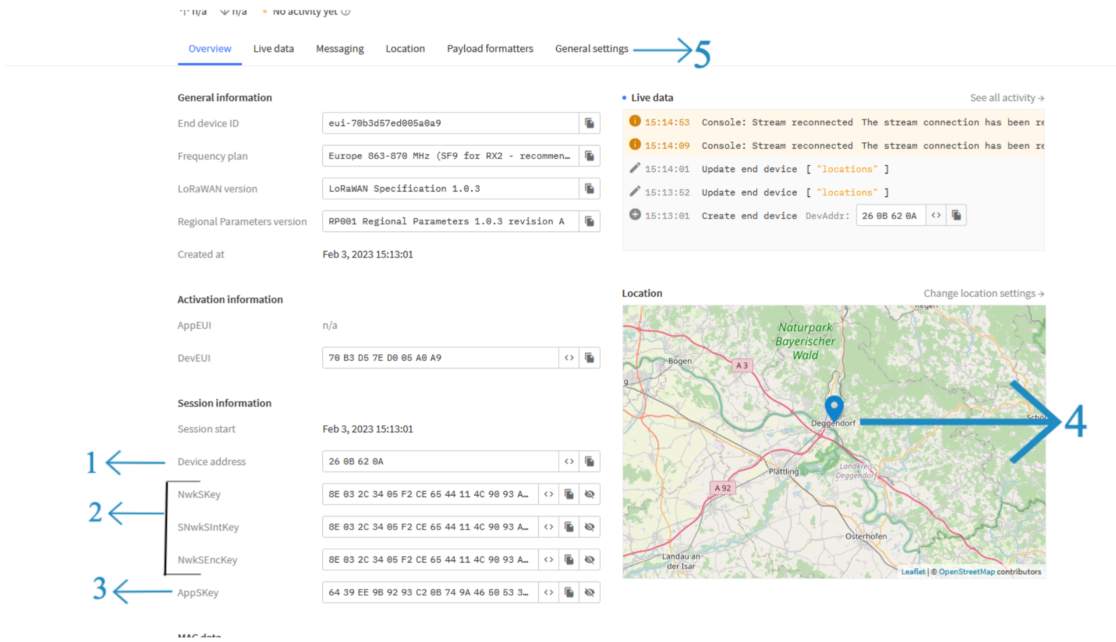

Figure 2.5: Overview of TTN

Transmission and receiving Sensor Signals

- Encoding and Modulation: The data is encoded and modulated at the node side using the LoRa modulation scheme. This involves converting the data to a specific format that can be transmitted wirelessly.

- Transmission: The encoded data is then transmitted wirelessly using radio waves. LoRa uses a spread-spectrum technique to transmit the data, allowing long-range communication with low power consumption.

- Reception: The received signal is demodulated and decoded at the gateway to retrieve the original data. The gateway then forwards the data to the TTN network server using a wired or wireless connection.

- Routing: The TTN network server routes the data to the appropriate application server based on the Dev EUI. The application server can be hosted on the cloud or on-premise[25].

- Response: The application server generates a response (if required) and sends it back to the node using the same LoRaWAN network.

- Node reception: The node receives the response and decodes it to retrieve the original data.

2.5 Hardware and Software

In order to connect hardware to the PC, it requires both a programming language and some software, so the Arduino nano comes into play first.

Microcontroller

The Arduino Nano is a diminutive, slim, and breadboard-friendly microcontroller board based on the ATmega328P. It is designed for experts, academics, and enthusiasts who want to do interactive electrical work [7]. Despite being cheaper and smaller than the Arduino Uno, the Arduino Nano has similar features. Here are a few of the Arduino Nano's salient characteristics:

- The Arduino Nano is built around the ATmega328P microcontroller, which has 32 KB of flash memory, 2 KB of SRAM, and 1 KB of EEPROM[7].

- The Arduino Nano has a USB port for programming and power, as well as a total of 14 digital I/O pins and eight analog inputs, it has also PWM analogue outputs.

- Power: The Arduino Nano has a built-in voltage regulator that allows it to be powered from a voltage range of 7 to 12 volts. Either an external power source or a USB port can be used to power it [26].

- The Arduino Nano can be programmed using the Arduino IDE, a free, open-source program available for Windows, Mac, and Linux.

- The tiny Arduino Nano (just 0.7 by 1.7 inches) is easy to use on a breadboard or in small spaces because of its small size.

From simple LED blinkers to complex home automation systems, the Arduino Nano microcontroller board is adaptable and user-friendly. It is easily accessible, and a sizable maker and enthusiast community supports it.

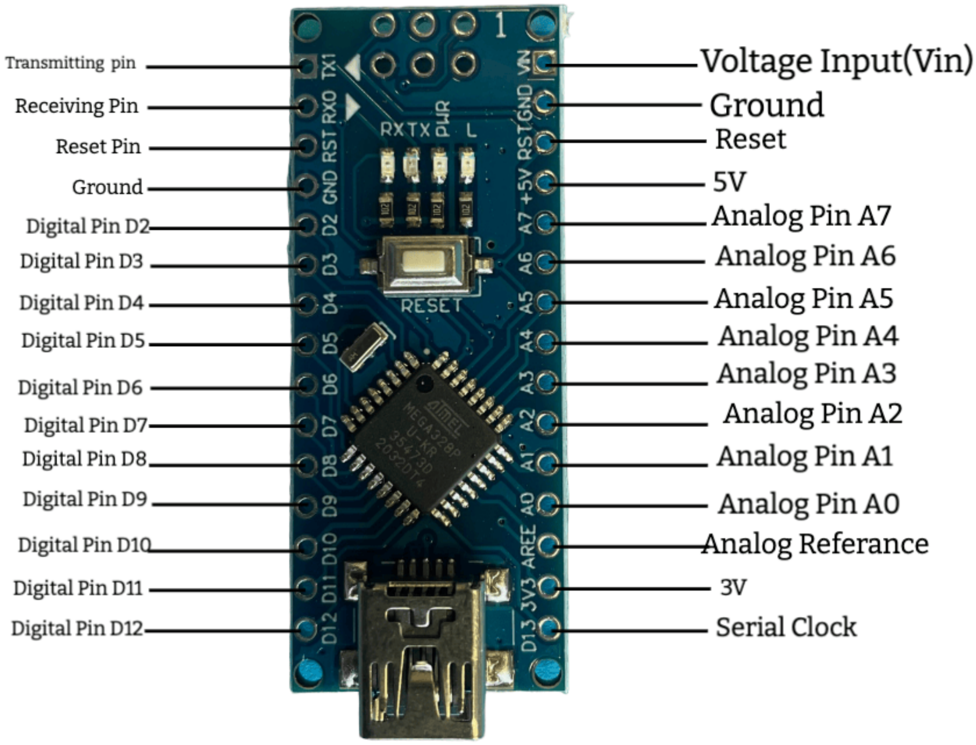

Figure 2.6: A brief description for the the Arduino pins

Using this Arduino Nano instead of an Arduino UNO is preferable for the following reasons:

- Smaller size and easier integration into tiny projects: The Nano has a smaller form factor.
- The Nano has a lower power consumption, making it ideal for battery-powered projects.
- The Nano is typically more cost-effective than the Uno, making it more accessible to everyone.
- More Digital I/O pins: The Nano has greater Digital I/O pins, giving sensors and actuators more connectivity possibilities.
- In order to use more sensors in your projects, the Nano has additional analogue input pins.

- There is no need for an additional converter while using the Nano because it features an internal USB-to-Serial converter. There are no soldered pins on the Nano, so it must have its pins soldered in order to be used on a breadboard. When soldering, extreme caution must be taken to ensure that the circuit is not shorted. If there are two pins at the same point, then it will be short.

2.5.1 Sensor

It has now received the sensor component that it will use. Utilizing the most basic temperature sensor, kTY 81-210, test the circuit first.

KTY 81-210

The KTY 81-210 temperature sensor measures resistance changes in a device to estimate its temperature. The sensor's resistance can be used to determine temperature by observing how it changes with temperature. Because the KTY 81-210 sensor is an NTC (negative temperature coefficient) thermistor, its resistance decreases as the temperature rises. Currently, KTY 81-210's functioning sensor

- The KTY 81-210 sensor is connected to a circuit that measures the resistance of the sensor.
- Since the relationship between temperature and resistance is directly, as the temperature of the sensor rises, the value of the resistor rises.
- The Temperature is then calculated using the resistance value. A voltage divider circuit, in which the voltage across the sensor is proportional to its resistance, can be used to do this. Using an analogue input pin on a microcontroller such as an Arduino, which is A0-A7, the voltage value with the help of a Multimeter. But one thing to keep in mind is that the Red probe is on VHz, and another one is on the ground, which is called the COM when measuring the voltage. The red probe can be set to mA or 20A if you want to measure current, but all of these settings depend on what you're trying to gauge.
- The resistance value is then used to calculate the temperature using the Steinhart-Hart equation. This equation takes into account the resistance value of the sensor at different temperatures, and is used to determine the temperature based on the measured resistance.
- The temperature value can then be displayed or used for further processing, such as controlling a heating or cooling system if there is any display otherwise we can easily measure the voltage from the circuit with the help of Multi meter.

Overall KT 81-210 sensor is widely used for Controlling the temperature ,Monitoring the temperature and thermal protection. This sensor is low cost and easily available in the market.As in Fig 2.7 Both Temperature and Resistance rise concurrently as a result of this increase in temperature.

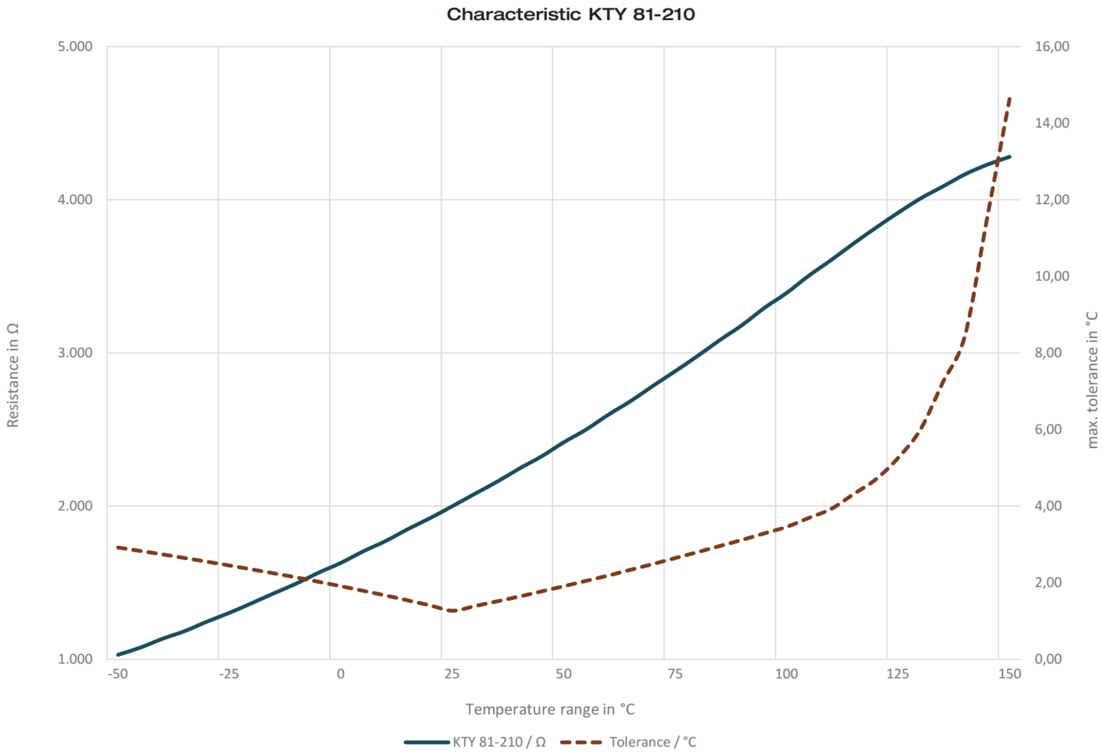

Figure 2.7: Characteristic KTY 81-210 [19]

2.5.2 Wifi Module

There are numerous Wi-Fi modules on the market, each with different features and functionalities. The following Wi-Fi modules are the most popular ones.:

RFM96 Transmitter and Receiver

The RFM96 is a long-distance, low-power transceiver module operating in the 868 MHz bands.

- IIoTiiot is frequently utilized in applications, including machine-to-machine and wireless sensor networks.
- The RFM96 needs to be set up to listen for incoming signals on a specific frequency, decode the broadcast data, and then extract the information included in the data to receive data.
- According to the particular needs of the application, the RFM96 offers a number of modulation schemes, such as ASKask, frequency-shift keying FSKfsk, and Gaussian minimum-shift keying (GMSK)gmsk.

- When using the RFM95 to send data, the signal must be modulated, the data must be encoded, and the signal must be transmitted on the designated frequency. Depending on the desired data transmission rate and the available bandwidth, the RFM96 provides a range of data rates from 0.3 kbps to 300 kbps.

- A microcontroller or microprocessor is commonly used to interact with the RFM96, and it communicates with the module over a serial interface like SPI (Serial Peripheral Interface) or I2C. (Inter-Integrated Circuit). The necessary actions for receiving and transferring data, including configuring the module, encoding and decoding data, and transmitting and receiving signals, can then be performed by the microcontroller through programming.

- The RFM96 module is a highly integrated device. Therefore, choosing and implementing a wireless communication solution must consider these elements carefully. These factors include power consumption, range, data rate, and interference.

As in Fig 2.8 shows that there are different pins available on this module.

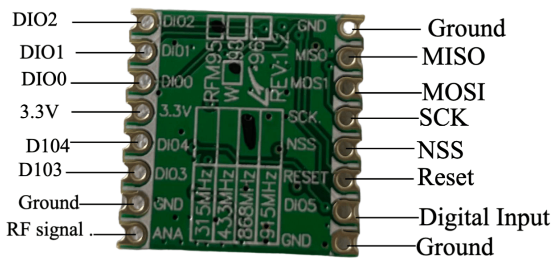

Figure 2.8: RFM96 pins Module

ESP8266

For IIoTiiot and connected device projects, the ESP8266 and Arduino platforms have been used.A wifi module and a software library with the necessary functionality to program the ESP8266 using the Arduino IDE must first be installed on the computer before the ESP8266 has been used.

 2 State of the Art of IIoT

- The serial pins (TX, RX) and a few more pins to supply power and reset the device connect the ESP8266 to the Arduino board.

- The installed library and the Arduino IDE are used to program the ESP8266.The ESP8266 has been used as a wifi client, access point, or both using programs. Additionally, they may be programmed to control other devices, send and receive data over the internet, and carry out various other duties.

- The ESP8266 has been communicated with other devices and servers over the internet once it is configured and linked to a wifi network. For instance, it can communicate with other devices on a local network, send data to a cloud server, or take commands from a mobile app.

- Esp board can be powered by any external source device, or there is also a possibility that it will also run by an Arduino Board, the main advantage of ESP is that it will go into sleep mode when there is no need for wifi module, or there is no data is transferring into it.

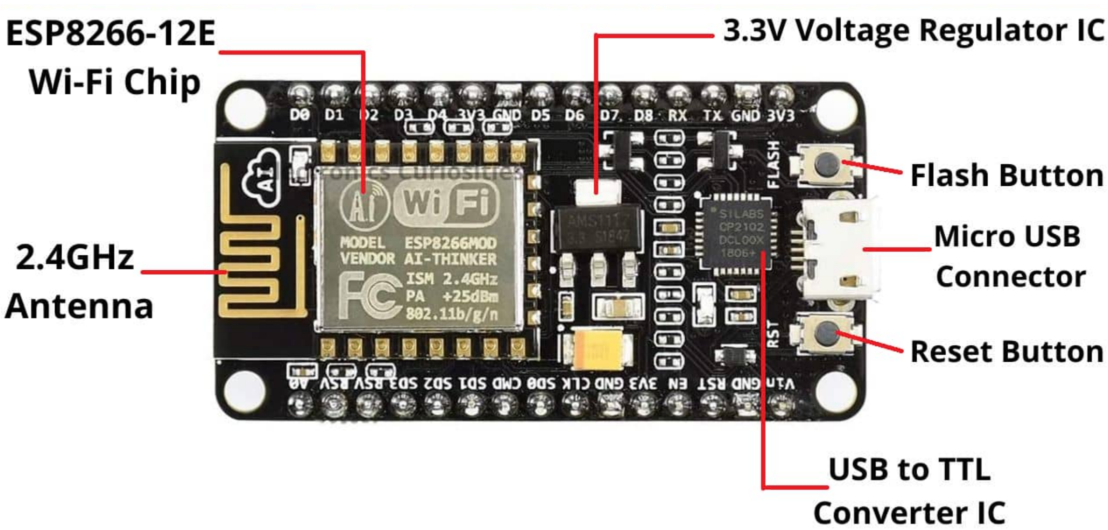

Figure 2.9: ESP8266 Node MCU [2]

ESP32

Espressif Systems created the ESP32 microcontroller, which is strong and adaptable. It is the popular ESP8266's successor and offers many upgrades and new features. The primary attribute of ESP32 is that it has a dual-core processor, which enables it to simultaneously transmit data and receive signals from multiple devices[8]Fig 2.10

- It also has a large amount of RAM and flash memory, which can be useful for storing and executing code, as well as storing data.

- The ESP32 includes a wide range of communication protocols, including WiFi, Bluetooth, and Ethernet. This makes it suitable for a wide range of IIoT applications, such as smart home devices, industrial automation, and more.
- One of the key feature of ESP32 is that it has built in Temperature sensor and other sensor also. so here don't need extra sensor in that module Fig 2.10.

As in Fig 2.10 there is a flash button where code is flashed on the esp32. There is a tiny reset button as well. Additionally, that has an ESP32 wifi chip. The antennae on this module have a great range.

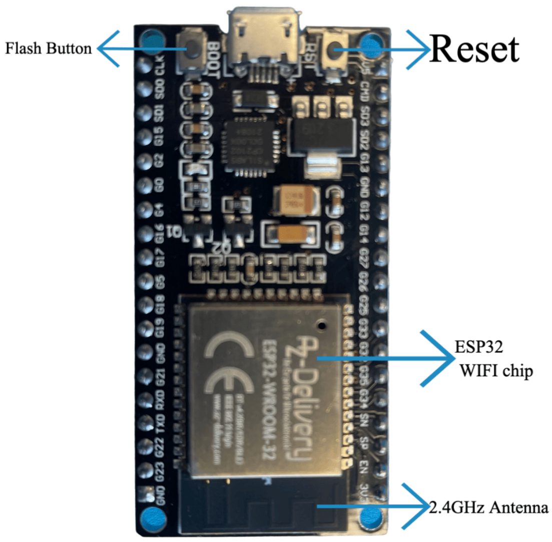

Figure 2.10: ESP32 Node MCU

3 Objective

Every project has some goals that must be met to be deemed finished. The main objective of this study is to interact with another device and gather data from it. The main goals are as a fellow;

- Communication with Different Nodes
- Storage of the Collected Data
- Live Visualization of the Collected Data

3.1 Communication with Different Nodes

The first step is to establish a connection between the two devices. Depending on the devices and the communication protocol being used, this can be done with a wired or wireless connection. Since the MQTT protocol is used to send data, ESP8266 is being used.The system can easily be connected to that device thanks to the simplicity of this protocol's creation and the fact that it is free, making it widely adopted.

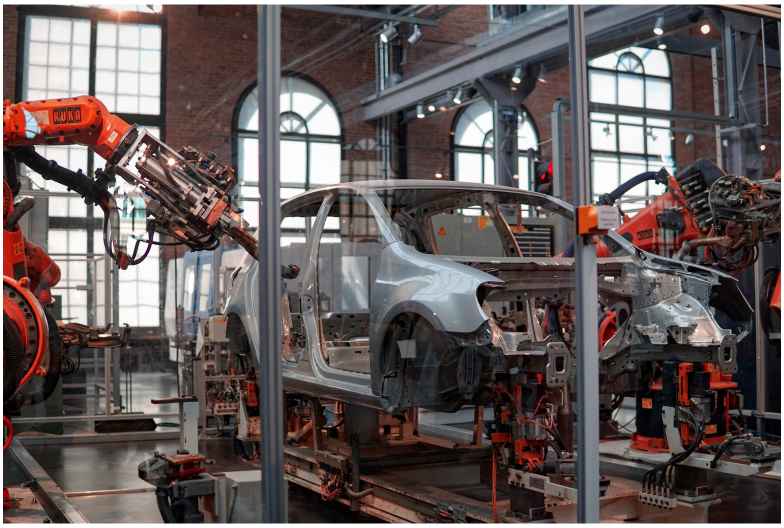

Figure 3.1: Automation Manufacturing of Car [18]

3.2 Storage of the Collected Data

Every sensor receives some data and produces some data, which is typically obtained by a device. The storage of this device and the tracking of data collection both require a separate device that can quickly identify when someone's sensor receives this data. It is necessary to store the data that has been gathered by the sensors. Typically, the data gathered will be useful for the duration of the project. All of the sensor data that is being gathered is therefore essential. The results will be estimated using this data. Either this memory card will be used to store the data, or the data will be stored on the MQTT protocol. You can make minor adjustments to the project, and you can compare the new results to the ones you've already saved. This makes it easier for the researcher to identify the variables influencing the process. Consider the temperature at a certain time, followed by another temperature at a different time, and so on. Then, their different data. Finally, save this data later and create a separate graph from it.

3.3 Live Visualization of the Collected Data

Reading many data might be challenging. In addition, the process of sorting through the data and giving it significance is time-consuming. Data visualization refers to the process of representing data graphically. Data visualization enables the user to quickly read the data. It also aids in the user's observation of the data's trends and patterns. Data visualization highlights relevant information and eliminates irrelevant data. These patterns enable others to conclude the study. There are different types of data visualization available.

- Charts provide us with crucial information about the relationships between data points Fig 2.7.
- Tables: provide the data's exact values table 2.1.
- Maps; that shows the geographical attributes of all places Fig 2.5.

Graphs are a great way to show a relationship between two variables, making them the type of data visualization that would be most suitable for this study. This uses real-time data visualization. Live data visualization is the process of visualizing data as it is being collected. As the temperature will be measured online, data will be stored online. Therefore, code will be written in Arduino so that data can be easily accessed on any protocol, like TTN or MQTT.

4 Implementation Results

The Design will be put into practice in this section. Various components are required to implement the circuit. ESP8266-2F Wifi module, Thermistor (KTY-210), one resistance, a breadboard and jumper wires, two RFM96 Modules, and a USB cable (mini USB to USB port) are the first items needed.

4.1 Connection with ESP8266

First, see the pins on this board for connecting the ESP8266 Module. The voltage must first be supplied to this board using a USB port in order to measure the temperature through the thermometer. Therefore, the esp8266 Module typically requires 4.5V Fig 4.1. In the circuit, there is a thermistor on one side and a 1,6Kohm resistor on the other. Basically, create a voltage divider by connecting a resistor and a thermocouple. Thermistor has two ends: one connected to a resistor and the other to the ground, so circuit must be grounded regardless.

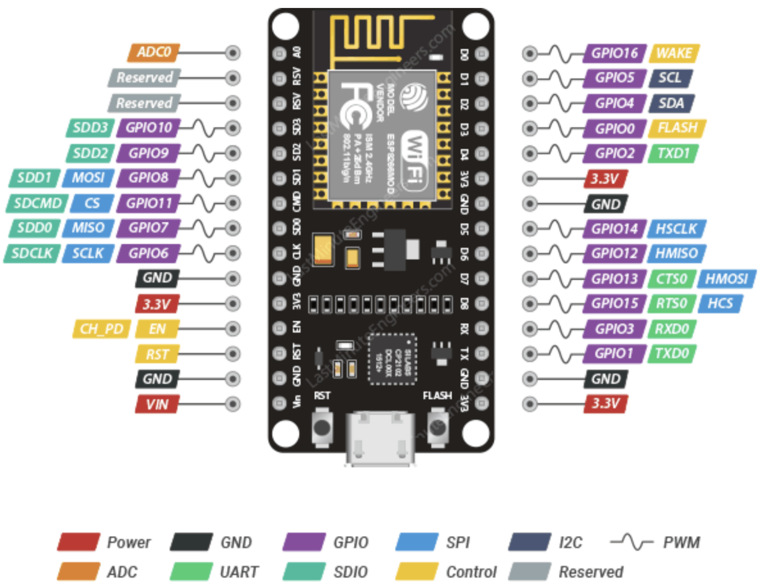

Figure 4.1: ESP8266 Pins [27]

Voltage Divider

Thus, the same current but different voltages are produced by connecting the thermistor and resistor in series. The multimeter will be used to measure the voltage across the thermister.

$$V0 = \frac{R2}{R1 + R2} * Vin \tag{4.1}$$

so as this circuit Fig 4.2 is in series so the voltage will be different as here using the voltage divider rule as shown in the equation 4.1 . Therefore voltage divider means that we are interested in R2 of the voltage so it will be above the equation.Here V0 is the voltage between thermistor and Resistor, Vin is the 5V we are supplying in the circuit while R1 is the resistance of thermistor and R2 is the resistance that we can enter in our bread board and we know the value of that resistor also as we are using in this circuit is R2 is 1.6K ohm. So all the values in above equation except the value of R2 so rearrange this equation in this way that R2 value will be obtained Fig 4.2.

$$R2 = (\frac{Vin}{V0} - 1) * R1 \tag{4.2}$$

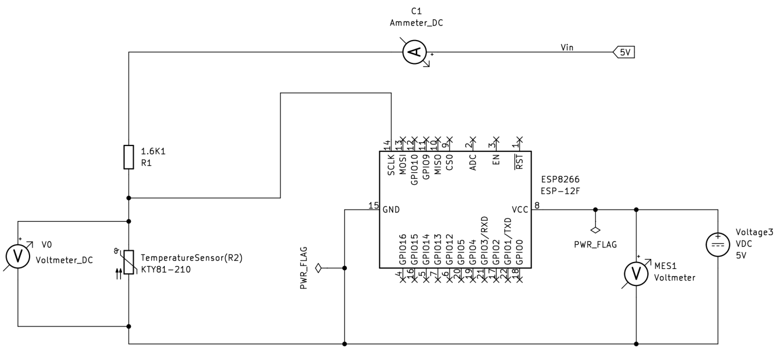

Figure 4.2: Circuit Diagram

After resolving this equation, we discover that the equation for the thermister resistor is identical to that of R2, given by equation 4.2, from which we can infer the resistor value. The ESP8266 module is already connected to this voltage divider circuit, so it is time to complete the connection. Therefore, it is now time to use it from a computer and view the temperature on the MQTT protocol and on the laptop extension.

4.2 Connection with Arduino

Now is the time to establish a connection with Arduino. The Arduino and hardware must be connected to the computer using the proper USB cable; this is essential as we ran into the same problem without it. ESP8266 is receiving the current, but Arduino is not connected. As a result, it must be confirmed that the wire will transmit the bare minimum current needed to turn on the esp, which is 4.5V.

4.2.1 ESP8266WIFI.h

First, if there is a need to connect the ESP8266 to wifi, we have to install the ESP8266WiFi.h library in your Arduino program to successfully connect with your wifi module, which is present near this device.

4.2.2 Connection with MQTT Protocol

In order to establish communication between devices on the Internet of ThingsInternet of Things (IoT), Message Queuing Telemetry TransportMessage Queuing Telemetry Transport (MQTT), a lightweight publish-subscribe messaging protocol, is used. An overview of how it functions is provided below:

- One device serves as a broker and other devices serve as clients in the client-server architecture of the MQTT protocol.

- A TCP/IP connection is used by the client to communicate with the broker. SSL/TLS is a secure connection method.

- The clients subscribe to particular topics on the broker. A topic is a string that stands for a particular subject that customers might find interesting. When a client sends a message, the broker receives it and forwards it to the relevant clients who have subscribed to the same topic.

- In order to use the broker, a user must subscribe to specific topics. An acronym for a specific subject that customers might find interesting is called a topic. The broker receives a message from a client and distributes it to the appropriate clients who have subscribed to the same topic.

- The publish/subscribe model is used by the MQTT protocol, in which a client sends a message to a broker on a particular topic, and the broker distributes it to all of the clients who have subscribed to that topic Fig 4.3.

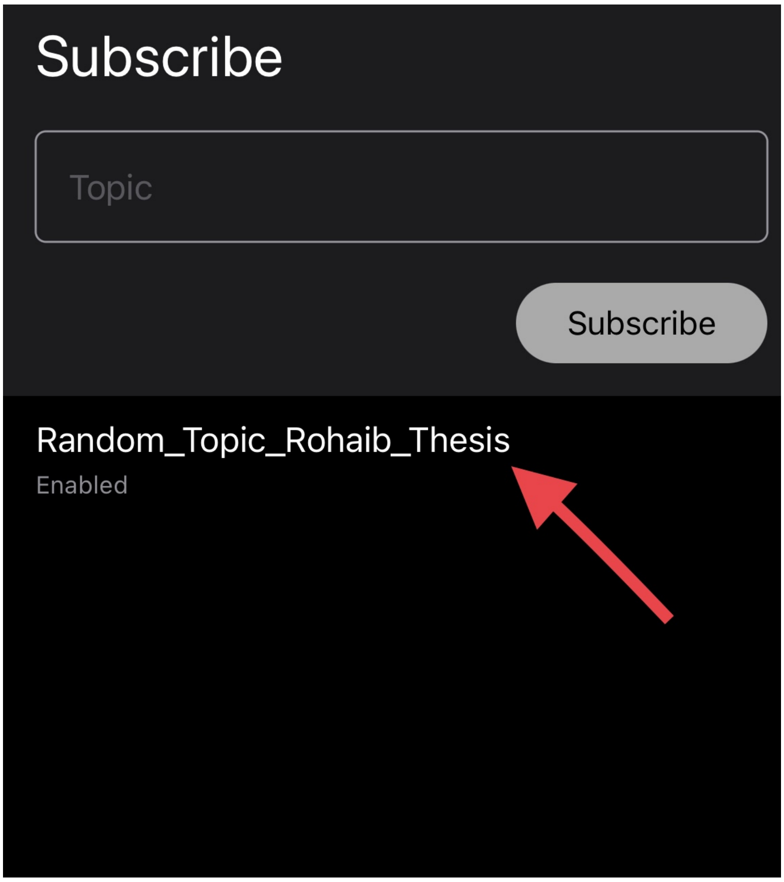

Figure 4.3: Subscription on APP

4.2.3 WIFI connection

It must now be connected to an ESP8266 connection using any wifi module, and there it must enter the device's name and password. The name and password must match what is displayed on the device. The name and password of that device must be checked if there is no connection since a problem should arise if the wifi name is incorrect. Esp8266 is searching for that network, so they don't come across any.

4.2.4 Variable for temperature Sensor

After connecting the WIFI module to the ESP8266, the values of a thermistor must be initialized. The resistance we're using for this circuit is 16000 ohm, which is in series with the thermistor Fig 4.2. As a result, the voltage will vary, but the current will remain constant throughout the circuit. Therefore, the voltage divider rule should be applied across the thermistor. Equations are used to transform the sensor reading into a temperature value because the thermistor's resistance varies with temperature Fig 4.2.

4.2.5 Final Value in Celsius

After using the voltage divider rule in this circuit, we now have all the values in the circuit except for the value of the resistor across the thermistor. All of these circuits will be rearranged in the thermister form. When we have the thermister voltage, we can easily determine the temperature of the area where our ESP8266 and Sensor are currently located. As we are getting this value in celsius, so it's up to us if you want this temperature in Fahrenheit. Also written in the code is that formula, but it is in the comment line. We will therefore obtain the temperature.

4.3 Setup of Practical

As shown in this Fig 4.4, there are two Digital Multi Meters. The one on the left displays the amount of current feeding this circuit, while the one on the right displays the voltage applied to the ESP8266 to initiate the circuit. The least voltage required for our ESP8266-2F Micro Controller is 4.5V. Therefore, you require a minimum of 4.5V and a maximum of 9V to turn on this circuit. Above this voltage, the microcontroller in this circuit will burn up. So, try to provide a range of voltage. Therefore, the ESP8266's current consumption is 75.1 mA.

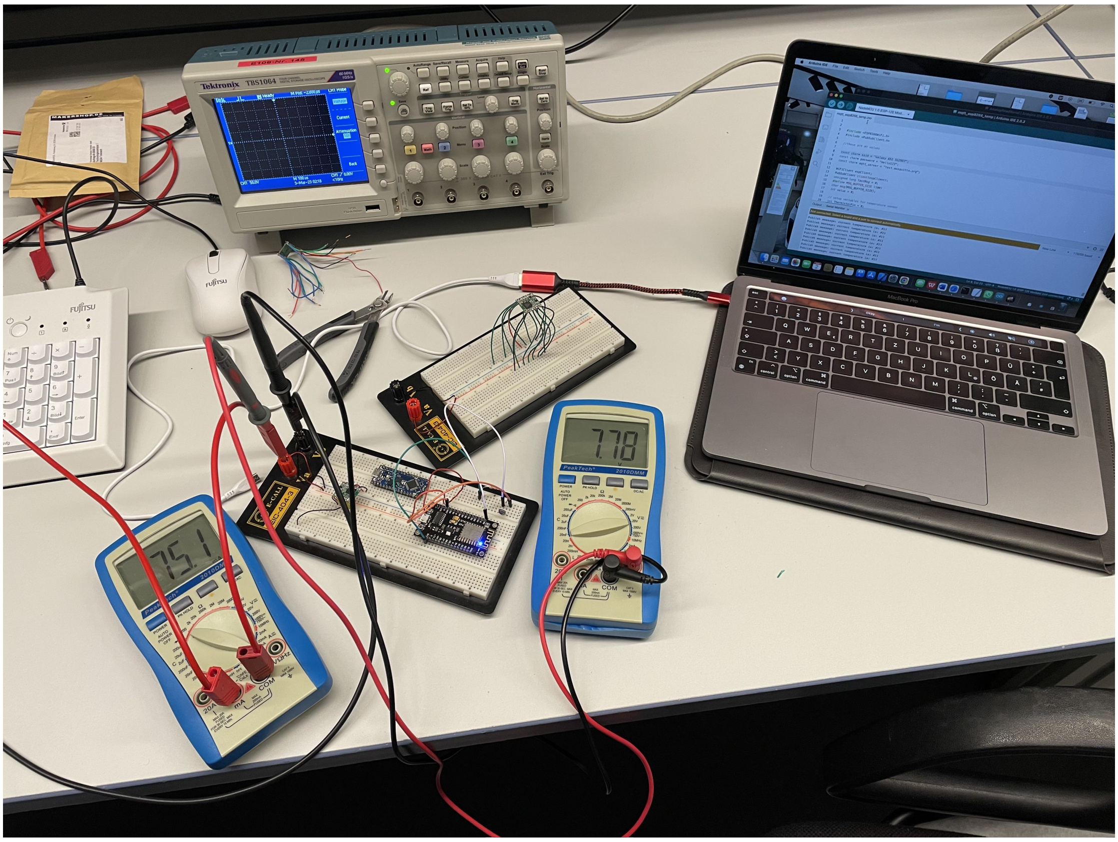

Figure 4.4: Overview of Circuit

MQTT App

It should be obvious that the temperature will show up on the MQTT protocol after five seconds because it has been given access to the MQTT protocol, which will receive the data. Consequently, the current temperature in the area is 23 degrees Celsius Fig4.5.

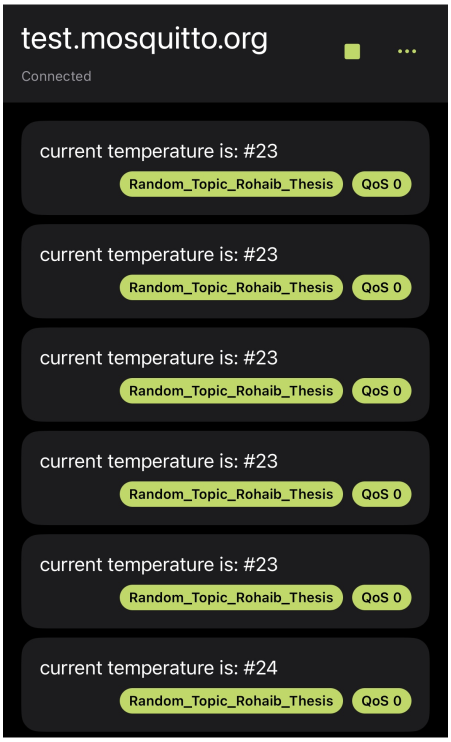

Figure 4.5: Room Temperature

Increase in Temperature

As is evident in Fig4.5, the temperature of the sensor rises after prolonged contact. So there is a little bit of increase in the temperature occur which is 24 degree at the end of that Fig4.5.

4.4 Power Consumption of ESP8266

The primary objective of this section is to calculate the power consumption of the ESP8266 Module. That is an important thing to keep in mind. The ESP8266 typically consumes 70 mA Table4.1 of power. A minimum operating voltage of 4.3V and a maximum operating voltage of 9V is needed by the ESP8266.

$$P = V * I \tag{4.3}$$

$$P = 7.78 * 75.1mA$$

$$P = 0.584W$$

4.4.1 Power consumption for 1 year

The annual power consumption of the NodeMCU can thus be easily calculated. being that the value was taken from the value above, which is only an assumption.

$$E = P * 365 * 24$$

$$E = 0.584 * 365 * 24$$

$$E = 5.118kWh$$

Only for indoor stations, the ESP8266 needed that much energy to last a year. However, if we need an outdoor station for this and it must always run on battery power, then we must be concerned. Therefore, we must first choose which battery we will always need to use in order to run it. There are many different battery types on the market, but we can choose the one we want to use by making that decision. The lithium-ion battery with a capacity of 2000mAh and an operating voltage of 4.5V should be used to determine how long it will take to change this battery.

$$T = \frac{2000mAh}{75.1mA} \quad T = 26.63h$$

It is evident from the 2000mAh value battery that it was changed after 26 hours. Therefore, changing the battery every day is absurd. Therefore, we must take special measures to increase the battery's efficiency.

- For the ESP8266, a 4.5V power supply is necessary. During the discharge, the battery's supplied voltage will drop from 4.5V to 0V. If the supplied voltage falls below 4.5V, the ESP8266 will shut off and stay off. The battery needs to be recharged. As a result, even if our battery has 2000 mAh of capacity, it will never be used up completely because, after less than 4.5V, NODEMCU is turned off, so it will never be turned on until it again receives above this voltage.

- The board may need to send data to a server as part of a task that we occasionally need to complete. There will therefore be brief intervals of excessive power consumption.

- Every battery experiences calendric ageing, which over time, reduces the usable capacity of the battery based on the cell chemicals and environmental temperatures.

4.4.2 Extend the Charging Time

There are only two possibilities left that can be used to lengthen the battery health;

- A battery with a higher capacity.

- Or find a way that could decrease ESP8266's power consumption.

As we have used 2000mAh, let's say we purchase a battery with a higher capacity. Increasing the battery's capacity is necessary if we plan to use it for the outdoor station. If we take the 6000mAh battery, then we can increase the lifetime of the battery by three times. It will extend to three days as soon as our battery has operated for one day. That is one way to make the ESP's battery last longer.
Sleeping mode is one of the many modes that exist in ESP, as are powerful solutions for these modes. We must adjust the date and time more so that the data will arrive after a certain period of time. ESP will enter a sleeping mode during that period, and battery consumption will be at a minimum. So doing this, we can increase the time interval of the battery.

Table 4.1: Power consumption of ESP8266 [14]

Numbers	Power Mode	Description	Power Consumption
1	Input Frequency	WIFI TX and Rx	70mA
2	Modem Sleep	CPU is working	15mA
4	Light-sleep	-	0.9mA
5	Deep sleep	RTC is working	20uA
6	Shut Down	-	0.5uA

Modem Sleeping

The Wi-Fi module is turned off in the modem sleep mode of the ESP8266 while the CPU and other peripherals remain operational. The ESP8266 uses about 15 mA Table 4.1 when it is in modem sleep mode, which is more than in deep or light sleep mode but less than when it is active.

$$T = \frac{2000mAh}{15mA} \quad T = 134h$$

Light Sleep

The ESP8266 also has a "light sleep" mode, which is a low-power mode that allows the chip to remain connected to Wi-Fi while consuming less power than when it is active. The ESP8266 can use as little as 0.9 mA Table 4.1 when it is in light sleep mode, which is more than deep sleep mode but still considerably less power than when it is active. The ESP8266 keeps its Wi-Fi connection while in light sleep mode and is still able to receive and process incoming data. To conserve power, some peripherals might be turned off, and their processing speed may be slowed down. The chip can wake up and resume normal operation in a matter of milliseconds in response to an interrupt, such as a timer or GPIO event.

$$T = \frac{2000mAh}{0.9mA} \quad T = 2223h$$

Deep Sleep

The module uses only about 20uA Table 4.1 of current in a deep sleep, but it may take longer to wake up from this mode.

$$T = \frac{2000mAh}{20uA} \quad T = 10000h]$$

Shut Down

The battery of the device will operate more efficiently if it is in shutdown mode its only needed 0.5uA Table 4.1.

$$T = \frac{2000mAh}{0.5uA} \quad T = 400000h$$

5 Summary and outlook

As shown in the previous results, we have discussed the different Protocol. When placing a Micro Controller anywhere, the main concern is to monitor the temperature there. The only consideration is that we must supply Arduino with power so that it can receive the temperature continuously via MQTT. We have also discuss about the microcontroller's excessive power consumption, which is shown above. Assuming the microcontroller is always on, we must consider the possibility of reducing the power consumption of the device. If the device sends the data after a period of time, there are several ways to increase power consumption. The Micro Controller must be in one of the following states during that time: shutdown, deep sleep, light sleep, or modem sleeping. If we use it that way, then we increase the life of the battery, as we see clearly above. It will be extremely advantageous if this setup is built in the future with a Lora node so that we measure a wide range of areas, as the Lora module has the capability of sending data over a long distance as well. If we use the Lora Node, then we send the data from a long distance, whereas at the moment, our setup needs the wifi every time so that it can send the data. We can thus broaden our area by doing that. The ESP8266 consumes less power than the ESP32. The ESP8266 consumes only 70mA of power, whereas the ESP32 consumes 240mA. The ESP32 would use much power and draw 790mA when using Bluetooth and wifi at the same time. However, the ESP32 has many functions if we use that device in the future because it has RAM and flash memory. This means that the device has the capacity to store data. They already have a temperature sensor built-in, so we don't need an additional sensor if we just want to check the temperature.

We have also faced too many problems while doing the research. First of all, we have to check the cable which is connected to the micro controller directly to the PC is correct or not because we have faced many issues in connecting the PC. As the light is also on, but there is no port showing in the Arduino, so be careful of choosing the cable. Every cable has different functionality, so some are used just for charging. If anyone has the same situation, then change the cable, and then it will work normally.

In the future, we will use the LoRaWAN Protocol as this is designed for long-range communication between devices with low power consumption. As the main concern, keep in mind that the device will be low power and also they have the ability to send data on longe range.so LoRaWAN has all the ability and is also secure. Also, we used the Mioty protocol for sending the data. It has the ability to send data in dense urban areas Environments. Mioty also supports a large number of devices, up to several million, so it is ideal for smart cities. Due to the low power consumption, the battery will be used for several years, which is ideal for the future. Also, if we need the data continuously, like in a racing car, There we need the data in seconds, then we use the Mioty device in that place.Everybody's individual needs must be considered. If we need more reliability and a longer range, MIOTY might be a better choice. LoRaWAN might be a more cost-effective choice, though, if we need to support more devices but don't need quite as much range.

Bibliography

[1] Designed for massive iiot, mioty is a lpwan solution based on etsi 103357. Wittra website. Accessed: March 15, 2023.

[2] Esp8266 pins, http://www.electronicscuriosities.com/2021/08/esp8266-node-mcu-board-specifications.html.

[3] Industry 4.0 revolution. IndutryRevolutions, http://www.renaix.com/de/industrie-4-0-die-vierte-industrielle-revolution/, note = Accessed: March 15, 2023.

[4] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. Understanding the limits of lorawan. *IEEE Communications magazine*, 55(9):34–40, 2017.

[5] Industry Agenda. Industrial internet of things: unleashing the potential of connected products and services. *White Paper, in Collaboration with Accenture*, 34, 2015.

[6] Tanweer Alam. Cloud-based iot applications and their roles in smart cities. *Smart Cities*, 4(3):1196–1219, 2021.

[7] Nur Aliya Arsyad, Syafruddin Syarif, Mardiana Ahmad, and Suryani As'ad. Breast milk volume using portable double pump microcontroller arduino nano. *Enfermeria clinica*, 30:555–558, 2020.

[8] Marek Babiuch, Petr Foltỳnek, and Pavel Smutnỳ. Using the esp32 microcontroller for data processing. In *2019 20th International Carpathian Control Conference (ICCC)*, pages 1–6. IEEE, 2019.

[9] Thadeu Brito, Matheus Zorawski, João Mendes, Beatriz Flamia Azevedo, Ana I Pereira, José Lima, and Paulo Costa. Optimizing data transmission in a wireless sensor network based on lorawan protocol. In *Optimization, Learning Algorithms and Applications: First International Conference, OL2A 2021, Bragança, Portugal, July 19–21, 2021, Revised Selected Papers 1*, pages 281–293. Springer, 2021.

[10] Sabyasachi Dash, Sushil Kumar Shakyawar, Mohit Sharma, and Sandeep Kaushik. Big data in healthcare: management, analysis and future prospects. *Journal of Big Data*, 6(1):1–25, 2019.

[11] Shu Du, Amit Kumar Saha, and David B Johnson. Rmac: A routing-enhanced duty-cycle mac protocol for wireless sensor networks. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 1478–1486. IEEE, 2007.

[12] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *sensors*, 12(9):11734–11753, 2012.

[13] Javier Gozálvez, Miguel Sepulcre, and Ramon Bauza. Ieee 802.11 p vehicle to infrastructure communications in urban environments. *IEEE Communications Magazine*, 50(5):176–183, 2012.

[14] Espressif Inc. Espdata. `http://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf`, 2023.

[15] João Jaime, Ivo Sousa, Maria Paula Queluz, and António Rodrigues. Planning a smart city sensor network based on lorawan technology. In *2018 21st international symposium on wireless personal multimedia communications (WPMC)*, pages 35–40. IEEE, 2018.

[16] John B Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[17] Karolis Kiela, Vaidotas Barzdenas, Marijan Jurgo, Vytautas Macaitis, Justas Rafanavicius, Aleksandr Vasjanov, Leonid Kladovscikov, and Romualdas Navickas. Review of v2x–iot standards and frameworks for its applications. *Applied sciences*, 10(12):4314, 2020.

[18] Lenny kuhne. Automationcar, https://unsplash.com/photos/jhz70nrk7ns, 2023.

[19] Volker Lange, Felipe Ribeiro, Walter Tews, and Dietrich Kühlke. Multicolor led sensor. In *Optical Sensors 2008*, volume 7003, pages 185–191. SPIE, 2008.

[20] Luca Leonardi, Filippo Battaglia, Gaetano Patti, and Lucia Lo Bello. Industrial lora: A novel medium access strategy for lora in industry 4.0 applications. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 4141–4146. IEEE, 2018.

[21] Jun Lin, Zhiqi Shen, and Chunyan Miao. Using blockchain technology to build trust in sharing lorawan iot. In *Proceedings of the 2nd International Conference on Crowd Science and Engineering*, pages 38–43, 2017.

[22] Gökan May, Ilaria Barletta, Bojan Stahl, and Marco Taisch. Energy management in production: A novel method to develop key performance indicators for improving energy efficiency. *Applied energy*, 149:46–61, 2015.

[23] Kais Mekki, Eddy Bajic, Frédéric Chaxel, and Fernand Meyer. A comparative study of lpwan technologies for large-scale iot deployment. 5:1–7, 03 2019.

[24] Rabeh Morrar, Husam Arman, and Saeed Mousa. The fourth industrial revolution (industry 4.0): A social innovation perspective. *Technology innovation management review*, 7(11):12–20, 2017.

[25] Paul Murley, Zane Ma, Joshua Mason, Michael Bailey, and Amin Kharraz. Websocket adoption and the landscape of the real-time web. In *Proceedings of the Web Conference 2021*, pages 1192–1203, 2021.

[26] Arduino Nano. Arduino nano. *A MOBICON Company*, 2018.

[27] Juliana Ngozi Ndunagu, Kingsley Eghonghon Ukhurebor, Moses Akaaza, and Robert Birundu Onyancha. Development of a wireless sensor network and iot-based smart irrigation system. *Applied and Environmental Soil Science*, 2022, 2022.

[28] George-Cristian Pătru, Dumitru-Cristian Trancă, Ciprian-Marian Costea, Daniel Rosner, and Răzvan-Victor Rughiniş. Lora based, low power remote monitoring and control solution for industry 4.0 factories and facilities. In *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–6. IEEE, 2019.

[29] Nathan Rosenberg. The role of electricity in industrial development. *The Energy Journal*, 19(2), 1998.

[30] Sandra Sendra, Laura García, Jaime Lloret, Ignacio Bosch, and Roberto Vega-Rodríguez. Lorawan network for fire monitoring in rural environments. *Electronics*, 9(3), 2020.

[31] Suranga Seneviratne, Yining Hu, Tham Nguyen, Guohao Lan, Sara Khalifa, Kanchana Thilakarathna, Mahbub Hassan, and Aruna Seneviratne. A survey of wearable devices and challenges. *IEEE Communications Surveys & Tutorials*, 19(4):2573–2620, 2017.

[32] NAGA DIVYA SUNKARA. Mioty overview: a mathematical description of the physical layer. 2022.

[33] Maarten Weyn, Glenn Ergeerts, Rafael Berkvens, Bartosz Wojciechowski, and Yordan Tabakov. Dash7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication. In *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 54–59. IEEE, 2015.

[34] Andreas Willig, Kirsten Matheus, and Adam Wolisz. Wireless technology in industrial networks. *Proceedings of the IEEE*, 93(6):1130–1151, 2005.

[35] Jun Xu, Junmei Yao, Lu Wang, Zhong Ming, Kaishun Wu, and Lei Chen. Narrowband internet of things: Evolutions, technologies, and open issues. *IEEE Internet of Things Journal*, 5(3):1449–1462, 2017.

[36] Min Xu, Jeanne M David, Suk Hi Kim, et al. The fourth industrial revolution: Opportunities and challenges. *International journal of financial research*, 9(2):90–95, 2018.

[37] Yongshun Zhang and Xin Jia. Adaptive interference suppression for dsss communications based on compressive sensing. *International Journal of Communication Systems*, 31(11):e3699, 2018.

List of Figures

List of Tables

6 Appendix

```
//ESP8266WiFi.h are the libraries install for connecting the wifi module with
    ESP8266
// PubSubClient.h it is used for connecting with MQTT protocol
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

//these are mz values
//these are the wifi module on which currently ESP is connected
const char* ssid = "Galaxy A52 5G29D7";
const char* password = "Hello123";
const char* mqtt_server = "test.mosquitto.org";

//these are the initalize the value
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (100)
char msg[MSG_BUFFER_SIZE];
int value = 0;

// setup variables for temperature sensor and connecting with MQTT Mobile
//app or MQTT lens extension
int ThermistorPin = 0;
int Vo;
float R1 = 16000;
float logR2, R2, T;
float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;
String outTopic = "Random_Topic_Rohaib_Thesis";

//this will convert the value of Resistor into the Temperature

float returnTemperatureInCelsius(){
  Vo = analogRead(ThermistorPin);
  R2 = R1 * (1023.0 / (float)Vo - 1.0);
  logR2 = log(R2);
  T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));
  T = T - 273.15;
  //T = (T * 9.0)/ 5.0 + 32.0;
  value = T;
  return T;
```

```
}
// now if there is wifi is everything is alright then connection will be
    //established
void setup_wifi() {

  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);


//if wifi is successfully connected then temperature will be received after //
    5 sec
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();

  // Switch on the LED if an 1 was received as first character
  if ((char)payload[0] == '1') {
    digitalWrite(BUILTIN_LED, LOW); // Turn the LED on (Note that LOW is the
        voltage level
    // but actually the LED is on; this is because
    // it is active low on the ESP-01)
  } else {
    digitalWrite(BUILTIN_LED, HIGH); // Turn the LED off by making the voltage
        HIGH
  }
```

```
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.publish("Random_Topic_Rohaib_Thesis", "hello world");
      // ... and resubscribe
      client.subscribe("inTopic");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

void setup() {
  pinMode(BUILTIN_LED, OUTPUT);  // Initialize the BUILTIN_LED pin as an output
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void loop() {

  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  unsigned long now = millis();
  if (now - lastMsg > 2000) {
    lastMsg = now;
    value = returnTemperatureInCelsius();
    snprintf (msg, MSG_BUFFER_SIZE, "current temperature is: #%ld", value);
    Serial.print("Publish message: ");
    Serial.println(msg);
```

6 Appendix

```
    client.publish("Random_Topic_Rohaib_Thesis", msg);
  }
}
```