

LORA BASED LONG DISTANCE TRANSMISSION WITH ALARM SYSTEM FOR OLD-AGE PEOPLE

Submitted in partial fulfilment of the requirements for the award of
**Bachelor of Engineering Degree in
Electronics and Communication Engineering**

BY

RAMA NARAYANA REDDY Y (38130246)

VENKATESHWARA RAO M (38130120)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING
SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI , CHENNAI – 600119

MAY - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **RAMANARAYANA REDDY Y (38130246)** and **VENKATESWARA RAO M (38130120)** who carried out the Project entitled “**LORA BASED LONG DISTANCE TRANSMISSION WITH ALARM SYSTEM FOR OLD-AGE PEOPLE**” under our supervision from December 2021 to May 2022.

Internal Guide

Mr. IMMANUEL RAJ KUMAR

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **RAMANARAYANA REDDY Y (38130246)** and **VENKATESHWARA RAO M (38130120)** hereby declare that the Project Report entitled “**LORA BASED LONG DISTANCE TRANSMISSION WITH ALARM SYSTEM FOR OLD-AGE PEOPLE**” done by us under the guidance of **Mr. IMMANUEL RAJ KUMAR**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1)

DATE:

2)

PLACE: Chennai

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D.** Dean, School of Electrical and Electronics Engineering and **Dr. T. RAVI, M.E., Ph.D.** Head of the Department, Department of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Mr. IMMANUEL RAJ KUMAR**, Department of Electronics and Communication Engineering for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the project work.

We wish to express our thanks to all teaching and non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

We express our gratitude to our parents for their constant encouragement and support for the completion of the project.

ABSTRACT

The pandemic declared by the World Health Organization due to the SARS-CoV-2 virus (COVID-19) awakened us to a reality that most of us were previously unaware of isolation, confinement and the massive use of information and communication technologies, as well as increased knowledge of the difficulties and limitations of their use. This project focuses on the rapid implementation of low-cost technologies, which Support for Elderly People Using Lora Communication Technologies. The system proposed here as a proof of concept uses low-cost devices for communication and data processing, supported by Long-Range (LoRa) technology.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF TABLES	viii
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	X
1	INTRODUCTION	1
2	LITERATURE REVIEW	6
	2.1 LITERATURE SURVEY	6
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	9
	3.1 PROPOSED METHODOLOGIES	9
	3.2 AREA OF PROJECT WORK AND APPLICATION	9
	3.3 EDGE AND FOG COMPUTING	9
	3.4 LoRa COMMUNICATION PROTOCOL	10
	3.5 BLOCKCHAIN AND SMART CONTRACTS	11
4	MATERIALS AND METHODS USED	13
	4.1 PROCESSOR	13
	4.2 PATIENTS AND MEDICAL STAFF REGISTRATION	14
	4.3 DATA PROCESSING	15
	4.4 DATA TRANSMISSION	16
	4.5 PATIENT MONITORING	17
	4.6 HARDWARE SETUP	16
	4.6.1 Arduino Micro-Controller	16

CHAPTER NO	TITLE	PAGE NO
4.7	LIQUID CRYSTAL DISPLAY	26
4.8	RELAY	28
5.	CONCLUSION AND FUTURE WORKS	37
	REFERENCES	38

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
4.1	Technical Specifications	21
4.2	LCD Pinout Function	27

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	Architecture workflow	13
4.2	Arduino UNO	20
4.3	ATMEGA328	24
4.4	LCD 2x16	26
4.5	LCD Pinout	27
4.6	Relay	29
4.7	Relay schematic	30
4.8	Relay coil	30
4.9	Relay circuit	31
4.10	Lora Bandwidth	34
4.11	Cloud IoT Services	35

LIST OF ABBREVIATIONS

ABP	- Activation by Personalization
BLE	- Bluetooth Low Energy
DHT	- Distributed Hash Table
DLT	- Distributed Ledger Technology
FCNs	- Fog Computational Nodes
HCS	- Healthcare Monitoring Systems
HER	- Health Record
IoT	- Internet of Things
IPFS	- Interplanetary File System
ISP	- In-System Programming
LoRa	- Long-Range
MQTT	- Message Queuing Telemetry Transport
OTAA	- Over The Air Activation
PBT	- Power Battery Terminal
RNN	- Recurrent Neural Network
TTL	- Transistor–Transistor Logic
USB	- Universal Serial Bus

CHAPTER 1

INTRODUCTION

LORA opens new opportunities to the researcher to collect and transmission data to the servers in order to improve those systems and make a fast decision. LORA applications can help many in many fields, including but not limited to smart cities, industry and healthcare. Right now, the most utilization of LORA in healthcare can be summarized as remote monitoring of the patients status in real-time. LORA networks could be used to control and manage situations, like in 2020, when coronavirus disease (COVID-19) hit the world, without putting extreme constraints on individuals and businesses. COVID-19 causes respiratory disease and has been more contagious compared to SARS in 2003 [2]. The recommended official procedure to control viruses' spread is to observe physical (or social) distancing until vaccines are available. Contagious infections would have a lower risk of spreading if improved monitoring programs, healthcare, and transportation are implemented. A standard LORA device for healthcare is made up of various sensors connecting to a microcontroller and/or a server, which provides a real-time reading of the environment or users.

In the current pandemic, sensors connected to an AI-algorithm could be helpful to diagnose whether people are infected with the COVID-19 virus based on measurements such as body temperature, blood oxygen levels and coughing patterns. A wireless heart rate and body temperature recording device that provides accurate measurement throughout the day would provide a better and a true picture of patients' health. The device could also be a research tool to understand better the effects of pulse rate and temperature on overall health. This paper discusses collecting the heart rate and body temperature data. The gathered data can then be sent to a server or a cloud using a wireless connection. These connections include but not limited to Bluetooth, WI-Fi, 4G, 3G, and GPRS. The collected data could be sent to cloud storage to perform permanent storage or monitoring in real-time [13]. A mobile application can be used to observe the patient's information easily. This

system is expected to monitor patients' Pulse Rate and Temperature, while doctors monitor the patient's condition remotely without being physically present near the patient's bed. The doctors will produce more accurate results for diagnosing a patient's medical problem, which resulted from observing abnormal Pulse Rate and Temperature readings.

Such information will help the authorities to classify the patients depends on their risk status, and this will help to give priority to the users dependently. The project main objective is to help organizations adhere to the safety protocol and guidelines related to COVID-19 to help stop the virus spreading. The real-time health monitoring of patients by connecting multiple sensors to Arduino that collects data and transmits it to a cloud server could help save both patients' and doctors' lives.

Over the past few years, different healthcare monitoring systems (HCS) have been developed and proposed; aiming to collect, process, and analyze data acquired from various sensors and devices. These systems are also responsible for monitoring and observing patient vital signs. The advent of Internet of Things (IoT) technologies has also facilitated such progress. In this context, the Internet of Health Things is an area of uncompelling change and expansion in the new age of smart houses, smart cities, and all digital things. The cloud computing paradigm has emerged as a solution to develop HCS systems for patients' monitoring. Such systems include many types of medical sensors such as ECG, blood pressure, and oxygen monitor linked via a network to the cloud for storage and processing in order to allow medical professionals to observe, diagnose and give treatment to patients. In fact, apart the health legislation and patient rights and responsibilities, many countries have made efforts to establish HCS systems and make health services accessible for all individuals of society. As an example, we cite the Kingdom's efforts to provide comprehensive health care in an accessible and simple manner.

Cloud-based solution faces many design constraints related to network bottlenecks, in particular bandwidth and energy [4], are shaping the design HCS. These constraints are even more serious over the last years, when the amount of data is huge and QoS demands in healthcare systems are increasing. In this

context, according to industrial report [5], the connected medical devices market was valued at USD 28.24 billion in 2020, and it is expected to reach USD 94.32 billion by 2026. Edge and Fog computing are considered the key enabling paradigms to solve the network problems of the cloud centralized computational model to more decentralized models. In these models, the edge layer collects medical data from sensors, and transmits them to the fog layer, which in turn performs the treatments and sends the results and observations to the cloud layer via internet. However, both internet and cloud data centers are exposed to security threats. As medical data have an important role in disease analysis and in diagnostic results, securing patients' health documents is of paramount interest, targeting more than one privacy measures including access control and data integrity.

Recently, a wide range of IoT health applications and frameworks have been developed in the one side to automate and make health services accessible to individuals, and in the other side to facilitate medical data exchange between different actors in a secure and a rapid way. It is to be noted that personal medical systems have many requirements, such as data sharing, data security and consistency, data reliability, and convenience]. Traditional healthcare systems are not able to meet these crucial requirements because they cannot ensure the safe use and sharing of data in a secure and a controlled way. Hence, to face reliability, fault tolerance and privacy challenges, Blockchain technology is recently adopted in the medical and e-health domain. This technology is a secure and transparent distributed ledger, and it paves the way for a revolution in existing healthcare systems by integrating its unique features. However adopting such a computationally intensive-based mechanism to a resource-constrained IoT devices and ecosystems is a challenging task. In fact, multiple problems arise such as real-time processing, data security and scalability and architectural based problems. Currently, to the best of our knowledge, there are no studies that propose a real implementation of HCS and integrates Fog and Blockchain technologies with a focus on the architectural and scalability issues. For example, in the proposed framework studies theoretically the transmission of health record (HER) between Fog layer and Blockchain and no focus on the architectural and scalability issues is presented, nor a proof of concept and framework evaluation have been made. In this context, we propose LoRaChainCare, a HCS prototype model

integrating Fog and Blockchain to allow close to real-time data processing and scalable and secure patient's health. To mitigate real-time requirement, LoRa ChainCare integrates Edge and Fog devices supporting LoRa communication protocol which is a relevant communication technology for healthcare systems by which energy and security issue are reliable in addition to a long bandwidth transmission. In addition, LoRa chaincare integrates a secure and scalable storage model based on IPFS and blockchain technology.

Blockchain is a decentralized data management platform that provides immutability. Thus, it can perfectly support file traceability metadata on a distributed file system like IPFS, due to the similarity in their structure.

In this project, we propose a novel IoT architecture integrating Blockchain and LoRa network to monitor personal health care data in a secure and efficient process. Our proposed system is based on a multi-layer architecture to collect data from IoT sensing devices, share it and store it to be analysed and accessible among different actors. The Edge/Sensor computing layer collects data from different sources and send relevant ones to the upper layer which is the Fog layer. This latter is based on the LoRa network. Then, the cloud layer contains all the resources that will manage the recovered data from the lower layer. A single main private permissioned Ethereum is also implemented to assure traceability and security issues of the proposed HCS system. In this context, IPFS is combined with the Blockchain technology to improve the storage of medical records. Finally, a Data monitoring/Analytic layer allows to control and monitor the patients' health data through mobile or web applications.

This article first aims to review all related background in the context of fog computing, Blockchain technology and IoT healthcare systems. Then, it discusses related work with a particular focus on IoT-Blockchain platforms proposed in the healthcare domain. Finally, a multi-layer IoT-based Blockchain framework integrating the LoRa network, named LoRa ChainCare, is proposed. A real case-study implementation is also presented and evaluated in terms of Blockchain resources, data transmission time and power consumption.

In summary, major contributions of this article can be summarized as follows:

1. To present a background of the most prominent technologies related to Home Care Systems mainly: Fog and Edge computing, Blockchain and smart contracts;
2. To discuss and compare the related work dealing with IoT-Blockchain platforms in the context of e-healthcare;
- 3 To implement a multi-layer architecture based on IoT and Blockchain aiming to collect and secure patients' data access by healthcare service providers and stakeholders while satisfying security, reliability and energy constraints.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE SURVEY

Different state-of-the-art eHealth architectures have incorporated Fog and Edge computing to make the processing and access of health medical data faster.

Authors considered hierarchical infrastructures composed of four layers: IoT layer, Edge layer, Fog layer, and Cloud layer. The mobile device receives health data from user's sensors and provides the environmental and geolocation information to send them along to the Fog node. The authors proposed Mobi-IoST, to address the problem of a seamless connectivity between IoT devices and cloud servers due to devices mobility.

The authors propose fall detection system architecture integrating smart LoRa-based Edge gateways (Edge layer) and LoRa access points (fog layer). In their work, health and contextual data are sent using Bluetooth Low Energy (BLE) to a LoRa gateway for compression and forwarded later to the LoRa-based access point to RNN processing (recurrent neural network) and distributed storing. Although system implementation is presented, the authors only evaluated the precision of implemented RNN with no evaluation considering Edge and Fog performance.

Although the aforementioned works have explored Fog and Edge computing for rapid access and processing of data in healthcare systems, a weak interest is accorded to security and privacy of patient's data. Indeed, the distributed topology of Edge/Fog computing increases the need for securing the network against the attacks.

Blockchain is a cutting-edge technology, which is gaining considerable interest, and integrated in many works to cope with securing stored data and communication within network. The authors propose a three-layer architecture where a Fog layer is integrated between the healthcare sensing layer and cloud data centres. They integrate a Blockchain model at the edge of the network that offers trust access and control over the network. Each Fog node is a collection of blocks, smart contracts and ledgers. In addition to endorse security, Blockchain technology allows for storing the records of transactions between the different entities. The authors proposed FogBus, a platform independent framework, and presented a cost-efficient prototype for Sleep Apnea patients. The framework integrates both blockchain and Fog computing. Fog Gateway Nodes are connected to sensing layer via wireless/wired communication protocols such as Zigbee, Bluetooth and NFC. Fog Computational nodes (FCNs) are provided with processing cores, memory and bandwidth so that they are able to execute the back-end program of the applications. Some FCNs are equipped with adequate security features and fault tolerant techniques such as Blockchain and replication, patients' vital signs are remotely being monitored by an immutable history log, providing thus a global access to medical information. The focus of this work is the conceptual design of health data sharing systems based on Hyperledger Fabric Blockchain-based smart contracts. Four layers compose the developed system: connectivity layer (routing management, security management, message brokers and network management), IoT physical layer of health (health devices), Blockchain IoT service layer (Blockchain-related service), and application layer (user interaction). Sensor data are sent by the gateway to the IoT server, which in turn, routes them to the Blockchain where they will be saved in the World State (off chain). Ray et al. investigated the integration of IoT and Blockchain for e-healthcare and discussed their features to harness the IoT-centric health services. They proposed an IoBHealth data-flow architecture for integrating Blockchain and IoT sensory data collected from patients to be securely accessed and managed by healthcare service providers and stakeholders. The proposed IoBHealth system is developed as an improved version of the existing Medrek architecture. Its ultimate aim is to transmit and manage the electronic health record (HER) in a decentralized way and to guarantee its security, immutability and transparency between the different stakeholders of healthcare industry. However, no experimental evaluation, nor a

comparison to prior work have been made. Authors proposed a Blockchain leveraged decentralized eHealth architecture integrating Edge and Fog devices and using Bluetooth or ZigBee protocol for data transmission. The Fog devices run a Blockchain protocol. To cope with the problem of large data storage on Blockchain, some works merge smart contracts with decentralized peer-to-peer Interplanetary File System (IPFS), which relies on a global Distributed Hash Table (DHT) to provide HCS records sharing in IoT scenarios. These works leverage IPFS technology to improve data sharing and IoT communication in untrusted environments.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 PROPOSED METHODOLOGIES

1. to present a background of the most prominent technologies related to Home Care Systems mainly: Fog and Edge computing, Blockchain and smart contracts;
2. to discuss and compare the related work dealing with IoT-Blockchain platforms in the context of e-healthcare;
3. to implement a multi-layer architecture based on IoT and Blockchain aiming to collect and secure patients' data access by healthcare service providers and stakeholders while satisfying security, reliability and energy constraints.

3.2 AREA OF PROJECT WORK AND APPLICATION

The focus of this section is to depict the Edge and Fog computing paradigms. It also presents LoRa communication protocol. Moreover, it gives definition and functionality of Blockchain technology.

3.3 EDGE AND FOG COMPUTING

Edge computing is the deployment of processing and storage resources near the location where data are produced. It alleviates the central clouds work by distributing some computations to processing devices located too close to IoT devices. Such tasks can be data filtering, analysis, or signal converting. The local data processing vastly reduces the transmission rate, occurring far less bandwidth

and service delay than might otherwise. The Edge computing integration becomes more important in healthcare systems by empowering it to react as fast as possible in emergency situations. Fog is first introduced by CISCO, and it refers to the distributed infrastructure between the Edge device and the cloud. It includes Fog nodes and the necessary network connections for transporting the data. Any device with computing, storage and network connectivity come under the definition of fog nodes. Fog nodes allow data to be collected and processed locally and reduce recurring transmissions over the cloud between Cloud and Edge devices. The proximity of Fog nodes to both IoT devices and end users ensures real-time computation and improved QoS.

3.4 LoRa COMMUNICATION PROTOCOL

LoRaWAN technology offers a long-range transmission of low data rate with optimized power consumption and a reduced cost. Indeed, data is transferred from the end device (sensor node) to the LoRa gateway using free unlicensed frequency within the ISM band so that it can be used, safely, for the e-health applications. Frequencies are ranged from 868 MHz to 900 MHz. LoRaWAN is able to transmit data up to three miles (five kilometers) in urban areas, and up to 10 miles (15 km) or more in rural areas. The other main feature of the LoRaWAN is the low power consumption measured in milliwatts (mW) so that some battery-operated devices can last for up to 10 years. LoRaWAN is deployed in a star topology which is adapted for applications that require long-range communication between a large number of devices that exchange small amounts of data (Data rate depends mainly on the spreading factor). All the LoRaWAN features are detailed in.

The components of a typical LoRa-based architecture and are as follow:

- The end devices: are the LoRa nodes which send the data to the Gateways and they are classified into 3 classes (A, B or C) depending on the node availability and its power consumption.
- The LoRa server: handles the entire network including the up-link and down-link communications coming respectively from the end device

to the application server and from the application server for the end devices. In the other hand, in this server, several encryption / decryption processes are performed to ensure the data security.

- Gateway: is the bridge between the end devices and the LoRa server. Three frequency channels and Internet protocol are supported to communicate respectively with end devices and LoRa server.
- Join server: checks for the end device authentication to send encrypted data through the LoRa network. The authorization is made through an activation process; Over-The-Air-Activation (OTAA) or the Activation by Personalization (ABP). The encryption is performed by the join server generating two session keys (NwKSkey and AppSkey). Using the ABP technology, the session keys are set statically in the end device and the network server, and they are not modified at any time. In the OTAA technology, the generation of the session keys is done dynamically at each device re-connect or reset. Consequently, the OTAA technique guaranties a better security in the data transmission between the end device and the network server.

3.5 BLOCKCHAIN AND SMART CONTRACTS

Blockchain, referred to as Distributed Ledger Technology (DLT) for transactions, enabling all participating nodes to validate transactions related data, ensuring thus unalterable and transparent history of any digital asset. It eliminates the need for third-party verification solving thus the problem of trust for sensitive data. All nodes of the network agree at the ledger to put each transaction into a block and validate that block with public key cryptography to append it to a chain. This validation process is referred as mining. All miners maintain a copy of the entire chain of blocks. The resulting list of blocks is chained using hash pointers and creates thus a tamper-evident data structure or log that helps to track data provenance and traceability abilities. Indeed, the potential of Blockchain is due to Smart contracts, which are algorithmic program codes, that can execute autonomously to perform transactions in Blockchain. They replace third parties that

centralize and manage the distributed ledger. All smart contract executions are recorded on the Blockchain, and made therefore publicly visible by all nodes. Additionally, Blockchain can help to decrease time and costs of information systems. The security of data in Blockchain is ensured by its distributed data storage, secure protocols, and mainly its consensus mechanism.

CHAPTER 4

MATERIALS AND METHODS USED

4.1 PROCESSOR

In this section, we describe the architecture workflow and the different functionalities of the proposed system. Like illustrated in Figure 4.1, the architecture workflow could be modeled with 4 parts. All these parts communicate together using exchanged data.

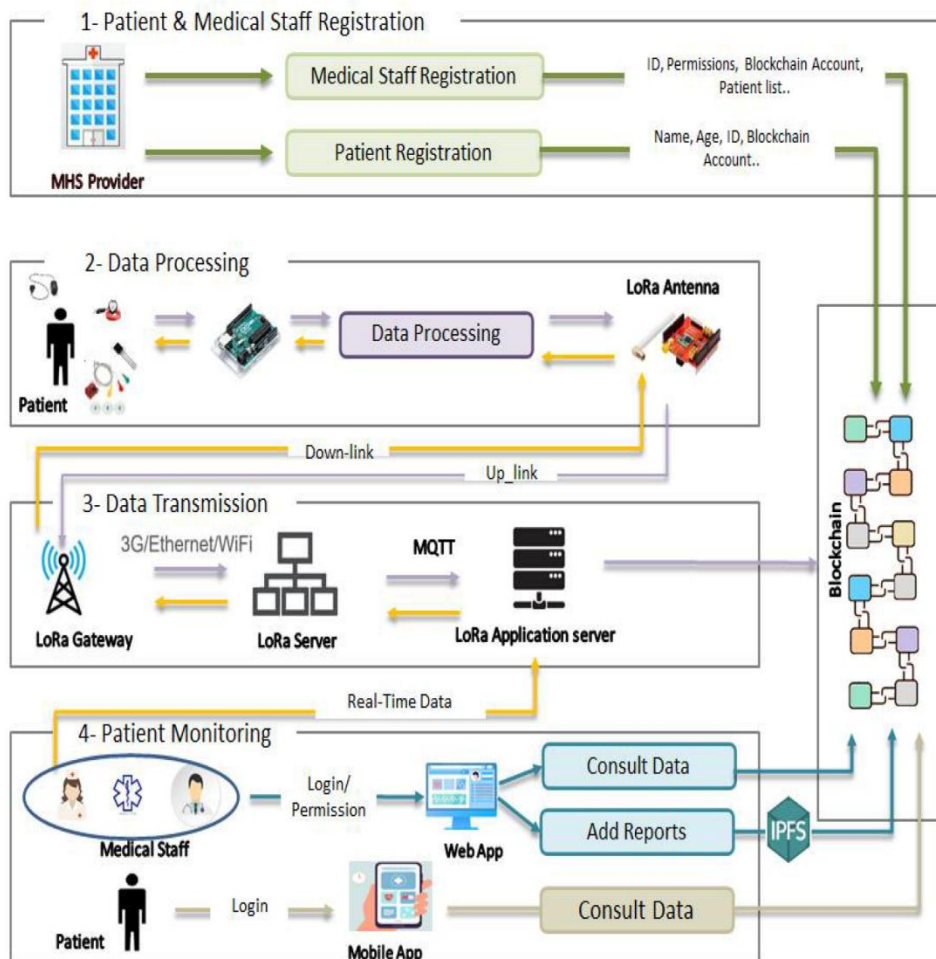


Fig. 4.1: Architecture workflow

4.2 PATIENTS AND MEDICAL STAFF REGISTRATION

Before starting the data monitoring and storing processes, all the active actors should be registered in the main system. The MHSP (Main Healthcare Service Provider) is responsible to register all the patients who benefit from the hospital services. The MS (Medical Staff) is also registered to ensure the different hospital services. Patients Registration: It includes the registration of the hospitalized patients, but also the elderly persons staying home and whose health state is monitored by the hospital's medical staff. All the patients parameters are stored in the Blockchain such as the Ethereum address which is used as an ID, Name, Age, phone number, etc. Each patient is affected to a corresponding MS to manage all the patient requirements.

For example, one or many doctors should monitor his health state, give the corresponding diagnosis and set reports describing the proper treatment. Patients are able to authenticate in the proposed system and they are authorized to consult their own data. Medical Staff Registration: The MS includes doctors, nurses and paramedical staff. The MHSP registers each one of them and sets their personal parameters like Ethereum address as an ID, Name, Role, Licence ID, etc. The most important thing is to affect the right permissions to each one of the MS since the proposed system is an access based control. Doctors are able to consult the historical data of the patients to which they are assigned. They are also authorised to ask for real time data before writing a report describing the health state and prescribing the right treatment. Nurses are allowed to consult the historical and real time data of patient to which they should carry on. They are able to request for a doctor intervention in case of critical condition through a web interface. Finally, the paramedical staff are allowed to view all the MHSP requests related to a given patient, such as preparing a specific meal to a patient, transferring a patient to the radiology department, taking samples from a patient to the analysis laboratory, etc. Each patient is selected with his ID so that all the needed parameters are shown to the requested Medical Staff.

4.3 DATA PROCESSING

After registering the patients, specific sensors related to their disease are used to measure all the needed parameters such as temperature, SPO2, ECG, blood pressure, sugar, etc. These sensors are connected to a processing platform which performs three actions:

Data Acquisition: The processing platform recovers data from several heterogeneous sensors (digital, analog, serial...) so that it converts the data to a comprehensive information. **Data Processing:** Classical systems proposed in the state of the art save all the recovered data in the off-chain data storage or in the cloud or Blockchain platforms. For the off-chain servers, data could be modified or deleted due to malicious actions (for example in the case of death that is due to medical malpractice). For the Cloud/Blockchain platforms, data storage requires resources fees. Thus, if the data is stored periodically (Every 1 or 2 min...), it may cost huge expenses especially when some data is insignificant (For example: The same “normal” temperature all the day). Therefore, in the proposed architecture, several processing tasks are performed. Each parameter is analysed and compared to given thresholds. Only values that exceed the thresholds will be stored in the Blockchain platform. Algorithm 1 shows a heart rate processing pseudo code. In this algorithm we studied the variation related to the patient’s age. To do so, we used the traditional formula proposed by Fox et al in 1971. However the maximum heart rate can depend on other parameters like the patient’s sex, diseases, activities, the drugs he takes.

Data Modulation: After analyzing data and decide which one will be stored in the Blockchain platform, the processing platform integrates a LoRa shield to ensure the modulation then the transmission of the data with the LoRa protocol.

Algorithm 1 Heart rate Analysis.

```
heart_R ← Read heart rate from sensor
age ← Read age from Blockchain
▷ Using the patient ID
Function heart_rate (heart_R: Integer, Age: Integer): Boolean
Store ← False
if (heart_R ≥ 120 - age) or (heart_R ≤ 60) then
    Store ← True
end if
heart_rate ← Store
EndFunction
```

4.4 DATA TRANSMISSION

The LoRa Shield sends the modulated data through the integrated antenna to the gateways whose ranges cover the considered end device. Indeed, the LoRa device is not registered in a specific gateway but into a given LoRa server, so that the patient data can reach several gateways in order to find an available channel. This data is demodulated in the received gateway then sent to the LoRa server using the internet protocol. The transmitted frame includes mainly the encrypted data using the AppSKey, the MIC field for the end device authentication and the frame counter which is used for security purposes and data duplicate elimination.

The data sent from the end device to the LoRa server is called up-link message. The data sent from the Application/LoRa server to the end device is called down-link message.

Finally, the data reaches the application server. Many communication protocols could be used such as the MQTT (Message Queuing Telemetry Transport) and the HTTP POST. In the proposed architecture we used the MQTT protocol. Indeed, MQTT is a light protocol based on the publisher-subscriber relationship

instead of the client-server one. Consequently, the application server has no longer to request data when it has no idea when it will arrive. Especially in our proposed systems, the data are not sent periodically, but it depends on the data processing performed in the end device. Patient data will be transmitted to the subscriber client (Application server) as soon as it arrives to the Broker.

This application server could be used to:

Store the patient data into the Blockchain platform.

Recover patient parameters from the Blockchain (for example the Patient Age).

(Down-link Stream).

4.5 PATIENT MONITORING

One of the main purpose of the proposed system is to ensure the patient data monitoring. In Fact, this data can be monitored through mobile application or Web application.

Mobile application: is used by the patient himself to consult his own data. After the patient authentication, he is able to discover all about his health state (list of assigned doctors, the diagnosis, drugs, doctor's reports...) according to the permission given by the MHSP.

Web application: is used by the Medical Staff to monitor the patient information. Indeed, after the authentication, each one of the Medical Staff is able to handle all the application functionalities according to the permission given by the MHSP. For example, doctor can consult all the patient data which are assigned to him and whose parameters are stored in the Blockchain platform. He can ask for realtime parameters using the LoRa down-link communication through the LoRa application server. Finally, he is able to write a report about the patient health state. This file will be uploaded by the doctor and stored into the IPFS platform. The Hash of the uploaded file is then added to the Blockchain platform.

Sensors are connected to the Arduino Uno platform which processes the data including data acquisition, data conversion, checking whether measurements are in normal or abnormal range and data modulation. Data modulation is performed to send patient data to the LoRa Gateway using the Arduino connected to LoRa Shield.

4.6 HARDWARE SETUP

Component Specification

- ❖ Arduino UNO
- ❖ Liquid Crystal Display
- ❖ Relay module
- ❖ Voltage sensor
- ❖ 12 Voltage regulator

4.6.1 Arduino Micro-Controller

Introduction to Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Why Arduino?

The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive
- Cross-platform
- Simple, clear programming environment
- Open source and extensible software
- Open source and extensible hardware.

Arduino UNO

The UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.



Fig. 4.2: Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Table 4.1: Technical Specifications

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 ma
DC Current for 3.3V Pin	50 ma
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Hardware

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The

official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in Arduino. An early Arduino board with an RS-232 serial interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are at the top, the 6 analog input pins at the lower right, and the power connector at the lower left.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino UNO is the Opti boot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Bo Arduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used. An official Arduino Uno R2 with descriptions of the I/O locations.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

Power

The Arduino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

Vin. The input voltage to the Arduino/Genuine board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7- 12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3. A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.

IOREF. This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can

read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.7

Memory

The ATmega328 has 32 KB (with 0.5 KB occupied by the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical.

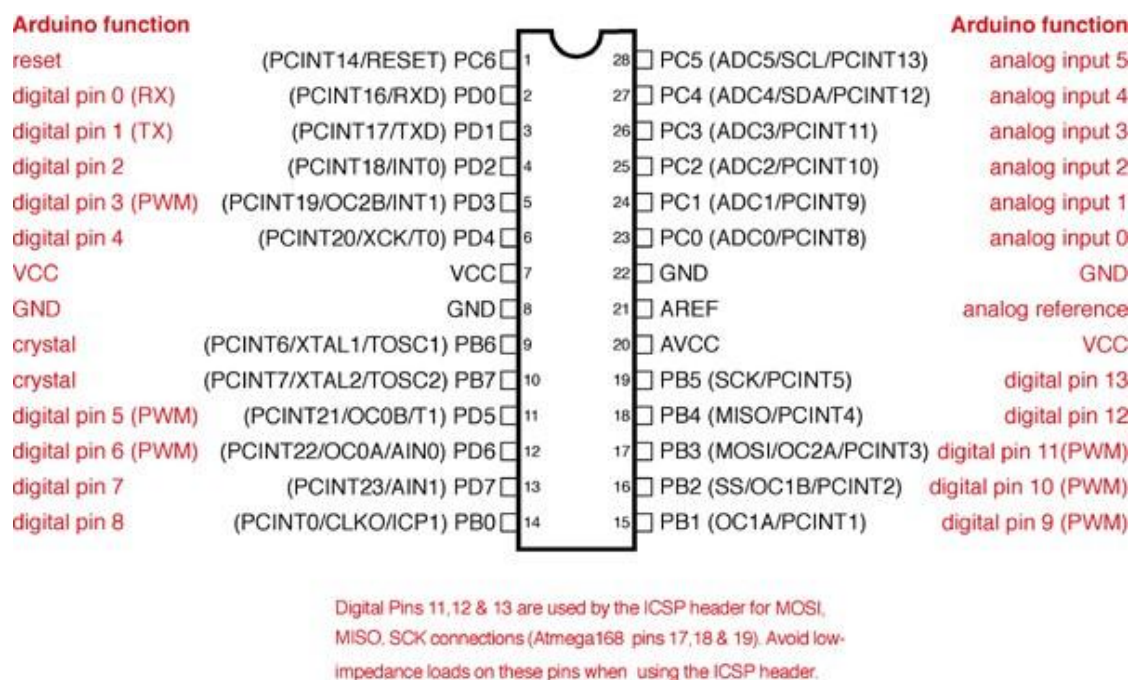


Fig. 4.3: ATMEGA328

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

I²C: A4 or SDA pin and A5 or SCL pin. Support I²C communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with `analogReference()`.

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

4.7 LIQUID CRYSTAL DISPLAY

LCD screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

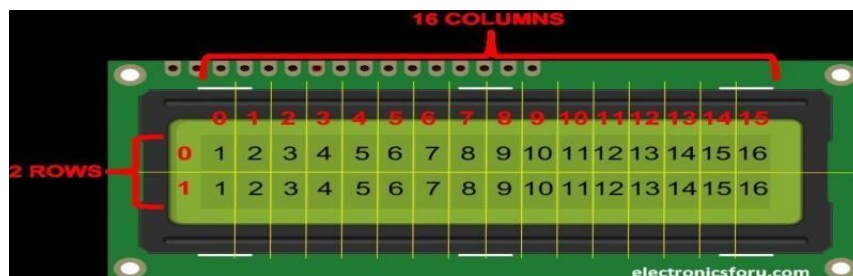


Fig. 4.4: LCD 2x16

We come across LCD displays everywhere around us. Computers, calculators, television sets, mobile phones, digital watches use some kind of display to display the time. An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16x2 LCD display is a very basic module commonly used in projects. The 16x2 translates to a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5x7-pixel matrix.

16X2 LCD Pinout Diagram

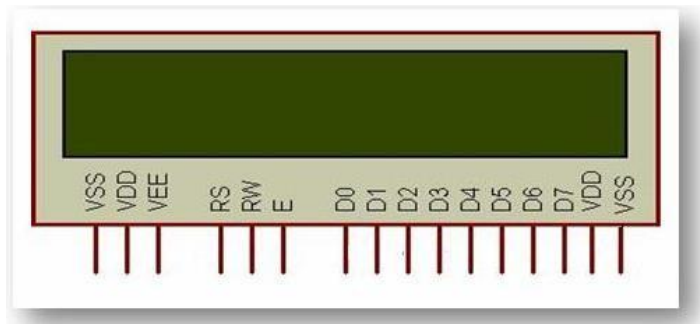


Fig. 4.5: LCD Pinout

Table 4.2: LCD Pinout Function

PIN NO.	FUNCTION	NAME
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	VCC
3	Contrast adjustment; the best way is to use a variable resistor such as a potentiometer. The output of the potentiometer is connected to this pin. Rotate the potentiometer knob forward and backwards to adjust the LCD contrast.	Vo / VEE
4	Selects command register when low, and data register when high	RS (Register Select)
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given; Extra voltage push is required to execute the instruction and EN (enable) signal is used for this purpose. Usually, we make it EN=0 and when we want to execute the instruction, we make it high EN=1 for some milliseconds. After this we again make it ground that is, EN=0.	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight VCC (5V)	Led+
16	Backlight Ground (0V)	Led-

RS (Register Select)

A 16X2 LCD has two registers, namely, command and data. The register select is used to switch from one register to other. RS=0 for command register, whereas RS=1 for data register.

Command register

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. Processing for commands happens in the command register.

Data register

The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. When we send data to LCD it goes to the data register and is processed there. When RS=1, data register is selected.

4.8 RELAY

The relay is the device that open or closes the contacts to cause the operation of the other electric control. It detects the intolerable or undesirable condition with an assigned area and gives the commands to the circuit breaker to disconnect the affected area. Thus, protects the system from damage.

Relay module

This is a 5V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high- current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller.



Fig. 4.6: Relay

Features

- 4.1.1** Input supply 12 VDC @ 170 mA
- 4.1.2** Output four SPDT relay
- 4.1.3** Relay specification 5 A @ 230 VAC
- 4.1.4** Trigger level 2 ~ 5 VDC
- 4.1.5** Berg pins for connecting power and trigger voltage
- 4.1.6** LED on each channel indicates relay status
- 4.1.7** Power Battery Terminal (PBT) for easy relay output and aux power connection
- 4.1.8** Four mounting holes of 3.2 mm each
- 4.1.9** PCB dimensions 88 mm x 68 mm

Pin description input

VCC: Positive supply voltage

GND: Ground

IN1--IN4: Relay control port

Output

Connect a load, DC 30V/10A, AC 250V/10A

Working principle of relay

It works on the principle of an electromagnetic attraction. When the circuit of the relay senses the fault current, it energizes the electromagnetic field which produces the temporary magnetic field.

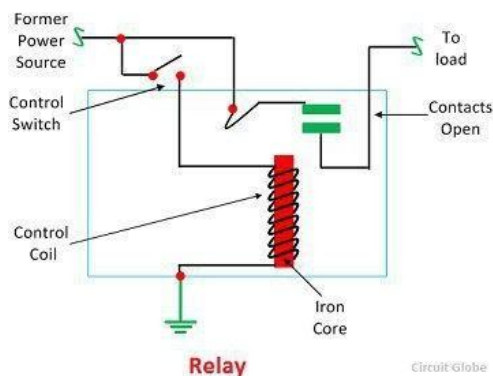


Fig. 4.7: Relay schematic

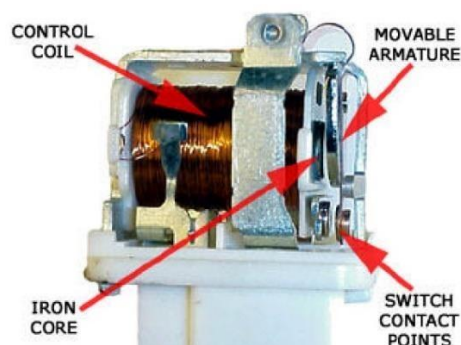


Fig. 4.8: Relay coil

This magnetic field moves the relay armature for opening or closing the connections. The small power relay has only one contacts, and the high-power relay has two contacts for opening the switch.

The inner section of the relay is shown in the figure below. It has an iron core which is wound by a control coil. The power supply is given to the coil through the contacts of the load and the control switch. The current flows through the coil produces the magnetic field around it.

Due to this magnetic field, the upper arm of the magnet attracts the lower arm. Hence close the circuit, which makes the current flow through the load. If the contact is already closed, then it moves oppositely and hence open the contacts.

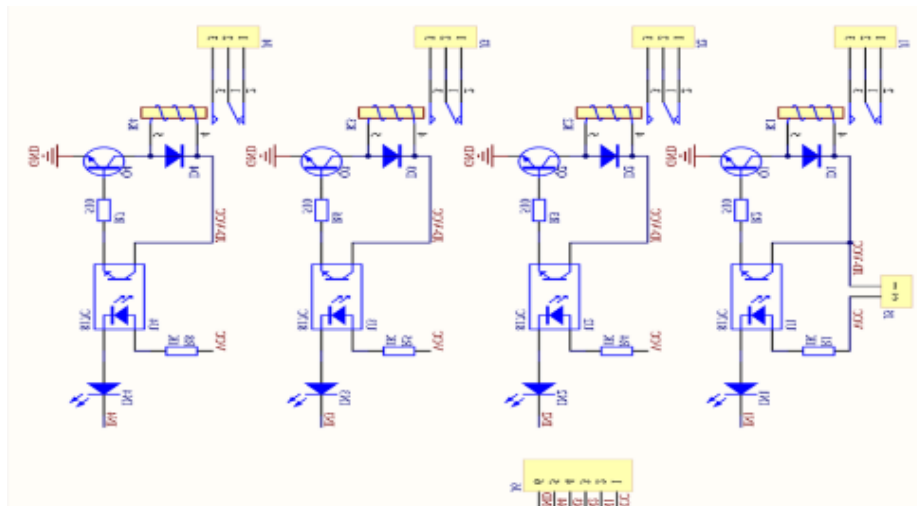


Fig. 4.9: Relay circuit

Pole and Throw

The pole and throws are the configurations of the relay, where the pole is the switch, and the throw is the number of connections. The single pole the single throw is the simplest type of relay which has only one switch and only one possible connection. Similarly, the single pole double throw relay has a one switch and two possible connections.

Construction of Relay

The relay operates both electrically and mechanically. It consists electromagnetic and sets of contacts which perform the operation of the switching. The construction of relay is mainly classified into four groups. They are the contacts, bearings, electromechanical design, terminations and housing.

Contacts – The contacts are the most important part of the relay that affects the reliability. The good contact gives limited contact resistance and reduced contact wear. The selection of the contact material depends upon the several factors like nature of the current to be interrupted, the magnitude of the current to be interrupted, frequency and voltage of operation.

Bearing – The bearing may be a single ball, multi-ball, pivot-ball and jewel bearing. The single ball bearing is used for high sensitivity and low friction. The multi-ball bearing provides low friction and greater resistance to shock.

Electromechanical Design – The electromechanical design includes the design of the magnetic circuit and the mechanical attachment of core, yoke and armature. The reluctance of the magnetic path is kept minimum for making the circuit more efficient. The electromagnet is made up of soft iron, and the coil current is usually restricted to 5A and the coil voltage to 220V.

Terminations and Housing – The assembly of an armature with the magnet and the base is made with the help of spring. The spring is insulated from the armature by moulded blocks which provide dimensional stability. The fixed contacts are usually spot welded on the terminal link.

What is LoRa?

The term **LoRa stands for Long Range**. It is a wireless Radio frequency technology introduced by a company called **Semtech**. This LoRa technology can be used to transmit bi-directional information to long distance without consuming much power. This property can be used by remote sensors which have to transmit its data by just operating on a small battery.

Typically **Lora can achieve a distance of 15-20km** (will talk more on this later) and can work on battery for years. Remember that LoRa, LoRaWAN and LPWAN are three different terminologies and should not be confused with one another. We will discuss them briefly later in this article.

Understanding LoRa Technology

In any typical **IoT solution** provided for warehouse management or field monitoring, there will be hundreds of Sensors nodes deployed on the field which will monitor the vital parameters and send it to the cloud for processing. But these sensors should be wireless and should operate on a small battery so that it is portable. Wireless solutions like RF can send data to long distance but requires more power to do so thus cannot be battery operated, while BLE on the other hand can work with very little power but cannot send data to long distance. So this is what brings in the need for LoRa.

In LoRa we can achieve high distance communication without using much power, thus overcoming the drawback of Wi-Fi and BLE communication. But how is it possible? If that is the case why do BLE and RF still exist?

That is because **LoRa comes with its own drawbacks**. In order to achieve high distance with Low power LoRa compromises on Bandwidth, **it operates on very low bandwidth**. The maximum bandwidth for Lora is around 5.5 kbps, this means that you will be able to send only small amount of data through LoRa. So, you cannot send Audio or Video through this technology, it works great only for transmitting less information like sensor values. The below chart shows where **LoRa lies compared with Wi-Fi, Bluetooth and Cellular devices**.

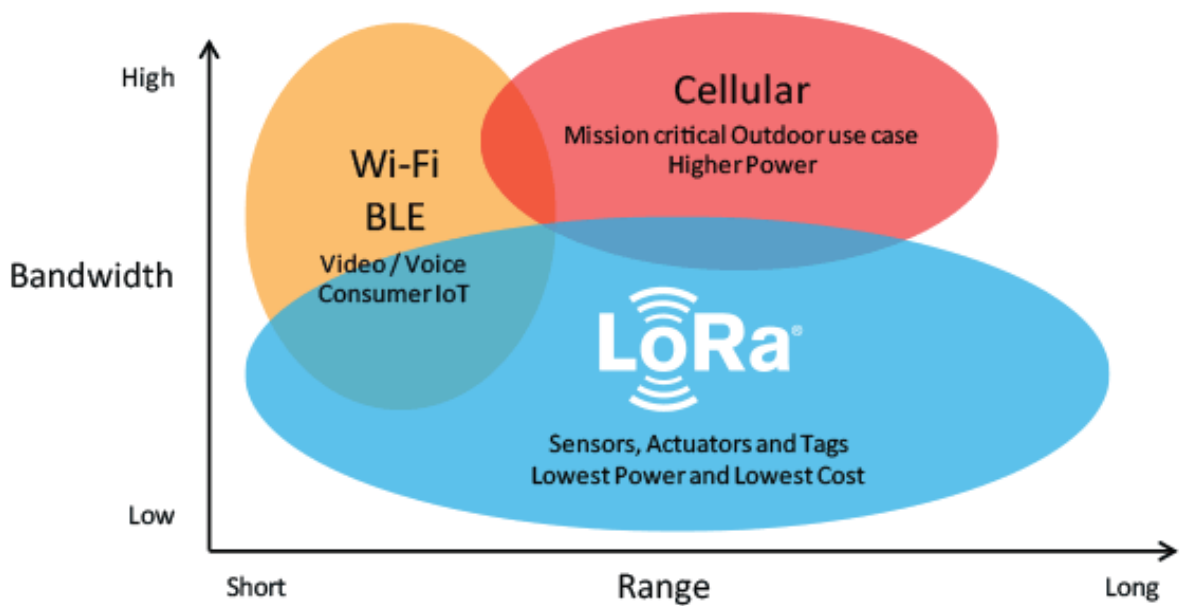


Fig. 4.10: Lora Bandwidth

Many people compare LoRa with Wi-Fi or Bluetooth, but these two do not stand anywhere near LoRa. Bluetooth is used to transfer information between two Bluetooth devices and Wi-Fi is used to transfer information between an Access Point (Router) and Station (Mobile). But LoRa technology was primarily not invented to transmit data between two LoRa modules.

You can think of LoRa to be more like cellular communication. Signal from one **LoRa Node** reaches another Node through a **LoRa Gateway** as shown in the image below.

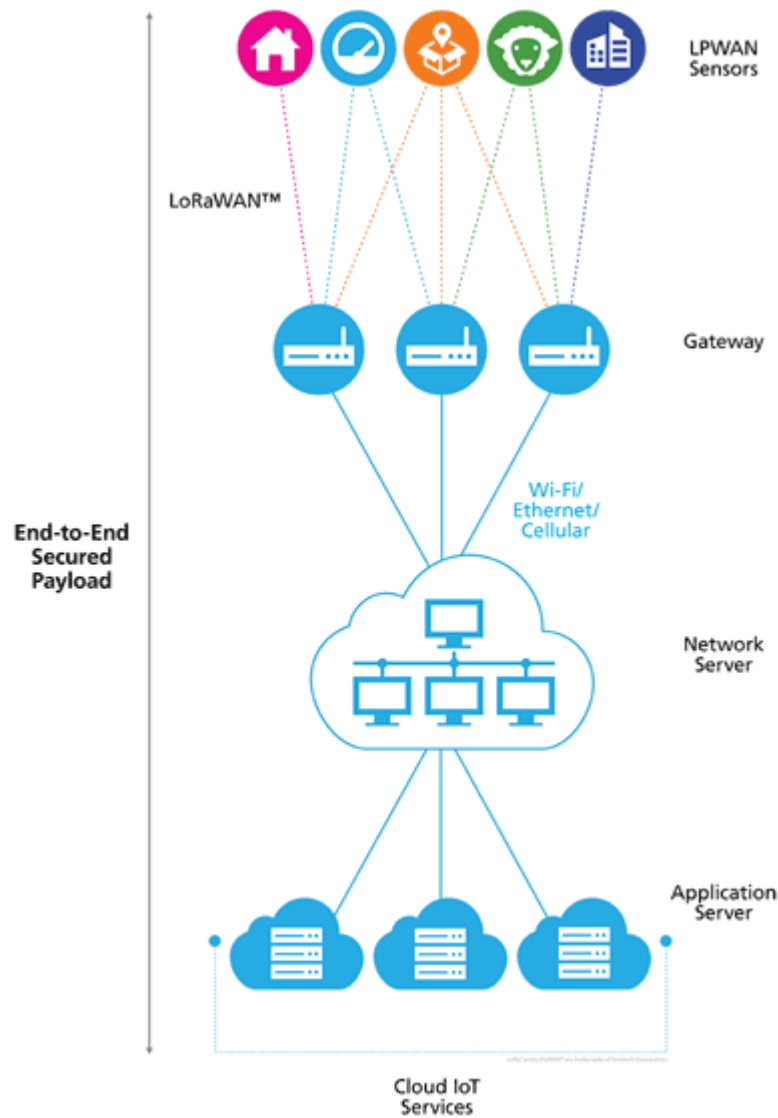


Fig. 4.11: Cloud IoT Services

These Gateways then take the information to the internet and finally to the end user through an application interface. Similarly the **data from the user will also reach the node through the network server and the Gateway.**

A LoRa Node usually operates on a Battery and consists of a Radio Module and Microprocessor. The Microprocessor is used to read the data from the sensor and send it in the air through the Radio module which will then be picked up by a LoRa Gateway. The LoRa Gateway also has a Radio Module and a Microprocessor but is normally operated over AC mains since they require more power. **A single LoRa Gateway could listen to multiple LoRa nodes, while a**

single LoRa node could also send information to multiple gateways, this way the information from the node will be picked up gateway without it being lost. **When information is sent from the node to the gateway it is called as uplink and when it is sent from gateway to node it is called as down link.**

LoRa falls under the category of LPWAN, where **LPWAN stands for Low Power Wide Area Network**. It is not just LoRa that can operate on LPWAN, but we also have other technologies like Narrow Band IoT (NB-IOT), Sigfox etc. which are capable of operating in the same LPWAN. Once the technology of LoRa was introduced, it needed certain set of protocols to be followed by all manufactures, so the LoRa alliance was formed which then introduced the LoRaWAN. LoRaWAN is a modified form of LPWAN which specifies the protocol on how LoRa in a physical layer should be used to send and receive data among the nodes, gateways and to the internet.

How much distances can I communicate with LoRa?

The Official Information on **LoRa claims that it could achieve a distance of 715km line of sight**. That is when there is no obstacle between the Node and Gateway. Few people have even practically achieved communication between 212km Ground to Ground and even upto 702km using weather Balloon.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

In this project, we proposed LoRaCare an IoT Architecture based on Blockchain technology for secure and authorized health data sharing, including patients vital signs and medical reports, by leveraging qualitative technologies. Blockchain, Edge/Fog computing and LoRaWAN were resorted to cope with the QoS requirements of HCS, comprising primarily low cost, security, scalability and reliable performance. Our proposed system allows health practitioner to monitor patients, ensuring health and medical safety are protected. The private-permissioned blockchain storage using decentralized file sharing system based on IPFS preserves security and solves Blockchain cost and scalability problem to share and store large data. Furthermore, our proposed HCS integrates Edge and Fog layers between IoT and Cloud layers. An Arduino Uno board acts as an edge device and communicates with the Fog device through a LoRa communication protocol which preserves reliable performance. As a proof of concept, we implemented a full LoRaChainCare prototype. The implemented system integrates environmental and health sensors connected to Arduino Uno board that uses an ATmega328P microcontroller is embedded and equipped with LoRa shield. Also, a web-user application is developed to allow medical staff to explore Blockchain services or to upload high-storage report into IPFS. We integrate the IPFS storage along with Ethereum to store large data that would be cost heavy if stored on Blockchain.

REFERENCES

- [1] Alka, N.U., Salihu, A.A., Haruna, Y.S. and Dalyop I.A., "A Voice Controlled Pick and Place Robotic Arm Vehicle using Android Application", American Journal of Engineering Research (AJER), Vol. 6, Issue-7, 2017.
- [2] Bishwajit Pal, and Samitha Khaiyum, "Autonomous Planetary Vehicle Development Platform", International Journal of Engineering and Advanced Technology, Vol. 9 Issue 1, 2019.
- [3] Chenguang Yang, Huaiwei Wu, Zhijun Li, Wei He, Ning Wang, Member, and Chun-Yi Su, "Mind Control of a Robotic Arm with Visual Fusion Technology", Transactions on Industrial Informatics, Vol. 3, Issue 4, 2017.
- [4] Daniel Anojan Atputharuban, Mobarakol Islam, Ravikiran Ramesh, and Hongliang Ren, "Real-Time Instrument Segmentation in Robotic Surgery Using Auxiliary Supervised Deep Adversarial Learning", IEEE Robotics And Automation Letters, Vol. 4, Issue 2, 2019.
- [5] Edward Tunstel, "Operational Performance Metrics for Mars Exploration Rovers", Journal of Field Robotics, Vol. 24, Issue 8, 2017.
- [6] Garima Aggarwal, Kevin Yu, and Debdeep Banerjee, "Object Tracking Test Automation Using a Robotic Arm ", IEEE Access Journal, Vol. 6, Issue 5, 2017.
- [7] Harmeet Singh, Sangeeta and Harpreet Kaur, "Design and Analysis of Robotic Rover with Gripper Arm using Embedded C", International Journal of Computer Applications Technology and Research, Vol. 4, Issue 2.
- [8] Kyohei Otsu, Masahiro Ono, Thomas J. Fuchs, Ian Baldwin, and Takashi Kubota, "Autonomous Terrain Classification With Co- and Self-Training Approach, IEEE Robotics And Automation Letters, Vol. 1, Issue 2, 2016.
- [9] Mark Maimone, Ashitey Trebi-Ollennu, Edward Tunstel, Jeng Yen, Rich Petras, and Reg Willson, "Mars Exploration Rover Mobility and Robotic Arm Operational Performance", IEEE Access Journal, Vol. 2, Issue 10.
- [10] Piotr Czaplicki, Maciej Recko, and Justyna Tołstoj-Sienkiewicz, "Robotic Arm Control System for Mars Rover Analogue", IEEE Robotics And Automation Letters, Vol. 1, No. 16.

- [11] Tian Wang, *Member*, Yuting Yao, Yang Chen, Mengyi Zhang, Fei Tao, Hichem Snoussi, "Auto-sorting System Towards Smart Factory based on Deep learning for Image Segmentation", IEEE Sensors Journal, Vol. 7, Issue 6, 2018.
- [12] Yasir ali memon, Imaaduddin motan, Muhammad ali akbar, Sarmad hameed, Moez ul hasan, "Speech Recognition System for a Voice Controlled Robot with real time Obstacle Detection and Avoidance", International Journal Of Electrical, Electronics And Data Communication, Vol. 4, Issue 9, 2016.