ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1. IoT – Internet of Things.
2. LoRa Module - Long Range Module.
3. Tx - Transmitter.
4. Rx – Receiver.

ABSTRACT

There exists digital meters in every household which reads the amount of electricity consumed. In order to generate the electricity bill, a person manually reads the data and generates the bill of the house. The main purpose of the project is to eliminate this step. Smart metering also helps us to know the amount of electricity consumed by a household and this will help in utilizing the energy efficiently. We would like to do so by using the LoRa and IoT Technology. LoRa provides long-range communication, which helps in data transfer and billing process. The Arduino UNO is used to collects data from the existing meters and transmit this to the master node - ESP8266. This node uploads data into the cloud database. The database stores this information and then an application is built to generate the bill for the consumed units of electricity.

Chapter 1

INTRODUCTION

The problem statement for our project says “Smart Meter Conversion: Enhancing existing Digital meters with LoRa IoT Technology for Automated Billing Mechanism”.

Each and every household in Karnataka/India have a digital meter with a LCD Display which shows the following readings- Date, Time, Voltage, Current, Power, No. of units consumed (kWH). A person visits each house in a locality carrying a meter-reading machine to generate the electricity bill according to the power consumption and this happens on a monthly basis.

This manual billing process is very tedious. It takes lot of time and human work. We already know when a person does some work or there is a human involvement, there is always some error when compared to work done by the machines. To overcome the problems we have used the LoRa and IoT Technology. We would like to use a cost effective approach and to achieve this we do not want to replace the entire existing meter but we incorporate an additional IoT device. This device is used to collect the data of amount of electricity consumed. We transmit this data using the LoRa technology to the concentrator or the master-node. The data present in the concentrator is sent to the database and helps in generating the Bill.

1.1 Main Objectives:

1. **Reduce Human Intervention:** Using the automated billing mechanism human interaction is eliminated.
2. **Acquire accurate Data:** The errors due to the manual billing process is reduced and the Data acquired is accurate.
3. **Reduce time taken:** The manual billing process is a very time taking process. Through automated billing mechanism we can reduce the time.

4. **Cost Effective approach**: The idea involves adding an additional IoT device to the existing meters and hence does not require more budget when compared to replacing the entire meter.

5. **Optimize the Electricity Consumption**: The customers are aware of the amount of electricity consumed on a daily basis and hence can minimize the electricity usage according to their requirement.

6. **Data Security:** The chances of Data Loss is reduced and the data is secured.

Chapter 2

LITERATURE SURVEY

The advanced smart metering is shown where the gas, water and heating consumption is monitored. The electricity is monitored is using EMS for 90 minutes and with intervals of 5 minutes [1]. Monitoring the electricity is done by many low cost microprocessors using smart meter, smart plug [2]. ESP8266 Wi-Fi module, Arduino Nano, GSM module was used for smart metering. Proteus 8.0 was used to test the idea before implementing it [3]. The measurement of maximum distance and response time of ESP8266 was carried out for the smart home system for lock, lights, camera system. The modules were made of ESP8266 Wi-Fi microcontroller. The communication system was JSON and HTTP and all the modules were connected to the central main server. The maximum distance from Wi-Fi to Module was 190 meters. The response time was 1.62 seconds [4]. A model is built to smart home automation, which consisted of smart energy meter and IoT based systems. The entire system was broken down into 3 small models .The models were done using the Raspberry PI, Arduino Nano, and Adafruit IO. All the models were tested separately and then again the complete system was tested. [5].

The monitoring of energy system was done using the 8 channel relay model, Arduino mega 2560 was used as central-node, Voltage sensors, current sensors, were used with ESP8266. All the readings were collected and sent to the ESP8266 module and then the sensor data was uploaded to the ThingSpeak Cloud Platform along with the IoT based Blynk App[6]. In order to ensure the limited use of electricity to reduce the billing cost, proposed system helps in monitoring the energy consumption from any far location using the internet. An Application was developed to monitor the usage and it was connected to the Wi-Fi using ESP8266, Arduino UNO and GSM [7]. The system was developed in the laboratory which used a main controller for switching on/off the appliances based on the price and consumption. This Home Automation System is made of decision module and command module. The Home Energy Management system was open for any changes in software or the hardware structure and allowed any updating. The system collected data of current, voltage and power and communicated with the energy meter [8]. The research was on understanding and identifying problems smart homes that use LoRa WAN. The research included about federated learning, which helps in effectively managing the services provided by smart homes [9]. The system was mainly for reducing the electricity bill. The extra energy was fed back into the electricity grid. This system found that there was 64.77% cost reduced, PAR and DI was reduced by 22.67% and 21.42% respectively [10].

The energy consumption is monitored using sensors such as current and power along with the 3G module, which helped in communication. Arduino was used as the main module or controller. The current reading was 88.33% - 99.29%. The voltage reading was 99.02% - 99.96% [11]. Smart Electric meter is developed using different IoT Layers and the ARM Cortex MC. The Mean Absolute Error and Root Mean Square Error obtained was 20%, Active Power – 37 W and Voltage – 43V [12]. The Customer IoT Model is used to build the Smart meter. The current and Voltage sensors are connected to the Arduino Nano and these are then sent to the ESP board after calculations. The obtained values are displayed on the serial monitor. The results are tested by comparing the obtained values with the voltmeter and ammeter readings [13]. Smart meter is made of a Control Unit that is connected to the AC power supply, Monitoring Devices, clock and storage device and the power module. ZigBee is used as the communication medium from the control unit [14]. The Energy Management System is divided into 2 parts – communicating with the Grid and with the user. ESP is used as the Wi-Fi module for communicating with user over the ThingSpeak Cloud Platform. The costumer can get the information regarding the Load and usage through the Blynk App [15].

PAPER	METHEDOLOGY USED	RESULTS
Paper 1	Smart meters, data collectors, database system. Communication – Radio Mesh Topology Data concentrator is capable of connecting up to 500 meters to it. The Data Concentrator is send the data to the database over Ethernet.	The Active and Reactive Power, Voltage is monitored for 90 minutes i.e for every 5 seconds and a graph is plotted.
Paper 2	Smart Meter, Smart Plug and the ZigBee Communication. Smart Meter sends data of the Power consumption to the data center. The Wi-Fi Module transmits all the Power parameters to the user's mobile or can upload data through ZigBee.	The Smart Meter shows the values of Vrms, Irms, Power, and Power Factor. Smart Plug – Energy Consumption of different regions in the house. The data is also shone in the User's Mobile when the Smart Plug is connected to Wi-Fi
Paper 3	Arduino Nano, ESP 8266, GSM Module. Arduino Nano acts as the brain of the system. It calculates the amount of energy and cost. ESP is the Wi-Fi Module to send and receive data. GSM is used to send SMS, Connect to the internet etc.	Simulation is done on the Proteus 8.0 Platform, LCD shows the Voltage, Current and Power readings. GSM sends the message about energy and the bill. Blynk App is used to monitor the energy readings.
Paper 5	Arduino Nano, R Pi, Adafruit IO Software. The amount of current consumed is stored in the current sensor and sent to Arduino Nano and then to R Pi.	The energy Meter data is uploaded to the cloud. After every 15 seconds, the data is sent and refreshed. The data is uploaded to the Adafruit software.

Paper 6	Arduino Mega 2560, Voltage and Current Sensors. The sensors are connected to Arduino Mega. It measures the current and Voltage Readings. The ThingSpeak Cloud stores the data from the Wi-Fi Module ESP 8266.	ThingSpeak loud shows the graphical representation of sensor data every 3 seconds. Blynk App is also used for monitoring. All these are done over Wi-Fi. This system consumes Less Power.
Paper 7	ESP 8266, GSM Module, Arduino UNO and Current sensor. The current sensor sends data to the Arduino Uno. AU calculates the data of energy consumed. This calculated data is sent to the mobile Application through Wi-Fi using the ESP8266 module.	The Current vs Power characteristic Graph is obtained. The highest and Lowest Power values were seen at 0.14A. The android app shows the amount of electricity consumed and the electricity bill.
Paper 11	Arduino UNO, RTC Module. AU acts as the main central system. It is connected to the GPRS module in the software. The RTC module is used for communicating with the MC using I2C Protocol. Arduino IDE Software is used to Program the AU Board and data is saved in the db.	The current and Voltage readings are recorded. The obtained results were checked for accuracy. The current accuracy - 97 to 98%. Voltage accuracy – 99.4%.

Table 1: Comparison Table

Chapter 3

PROPOSED WORK

The Project focuses on Converting the existing Digital Electric Meter into a Smart Electric Meter. The Smart Electric meter eliminates the need of human involvement in the Billing Process. The members of a house and pay their Electricity Bill from anywhere can generate the Bill. This is done by using IoT Technology and LoRa for Communication. LoRa helps in transferring the data collected by the IoT device. IoT is used to design a smart device, which collects data from the existing meter, and in turn help in Billing Process. The complete model is explained below in this section.

3.1 Hardware Requirements:

1.	Digital Electric Meter
2.	Arduino UNO
3.	ESP 8266 Node MCU
4.	LoRa Module – Sx1278
5.	TTL Converter
6.	RJ10 Cable

Table 2: Hardware Components

1. **Digital Electric Meter:**

   A digital electric meter, also known as an electric meter, is a device that measures the energy consumption of a building over a period. It is installed by electrical utility companies on a property and charges occupants based on the power usage displayed on the meter.

   It can record energy consumption in kilowatt-hours (kWh) and display the data in a digital format, making it easy for users to read and understand. Using a radio frequency signal emitted by the electrical meter, the utility department has a direct, real-time reading of the electricity consumed.

Figure 3.1: Digital Meter

2. **ESP8266 Node MCU:**

   The NodeMCU (Node Microcontroller Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. It has also Wi-Fi capabilities, so we can control it wirelessly and make it work on a remote installation easily! We can tell our board what to do by sending a set of instructions to the microcontroller on the board. To do so we can use the Arduino Software (IDE).

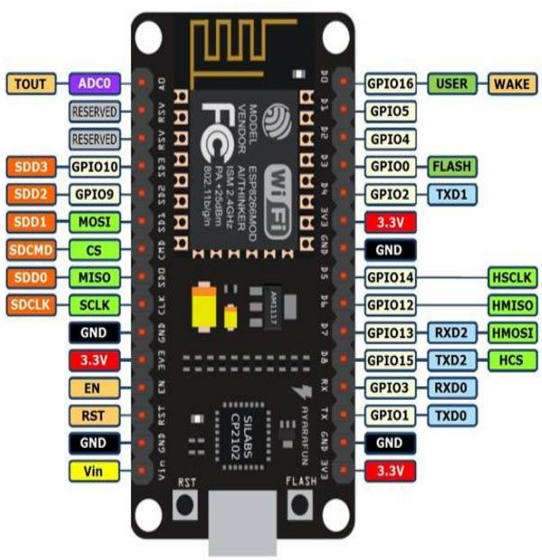


Figure 3.2: ESP8266

3. **Arduino UNO:**

   Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. It can be powered by a USB cable that accepts voltages between 7 and 20 volts, such as a 9-volt battery.

Figure 3.3: Arduino

4. **MAX232-TTL Converter:**

   The MAX232 translates a TTL logic 0 input to between +3 and +15 V, and changes TTL logic 1 input to between −3 and −15 V, and vice versa for converting from TIA-232 to TTL. (The TIA-232 uses opposite voltages for data and control lines)

   TTL also refers to transistor–transistor logic, a logic family made from bipolar junction transistors.

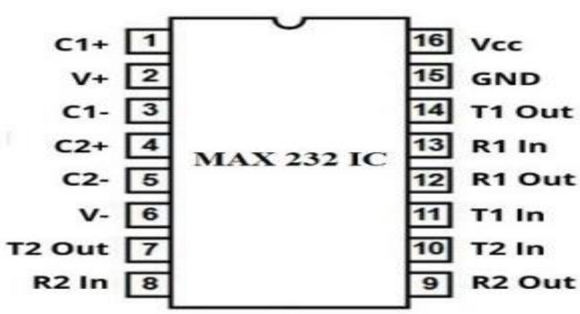


Figure 3.4: MAX232-TTL Converter

5. **LoRa Module- Sx1278:**

This module is a type of low cost RF front-end transceiver module based on SX1278 from Semtech Corporation. It keeps the advantages of RFIC SX1278 but simplifies the circuit design. The module is suitable for low range and low data rate applications.

The module consists of RFIC SX1278, thin SMD crystal and antenna matching circuit. The antenna port is well matched to standard 50 Ohm impedance. Users don't need to spend time in RF circuit design and choose suitable antennas for different applications. The module operates at 1.8~3.6V with extra low standby current which makes it suitable for battery powered-up applications.

Figure 3.5: LoRa Module - SX1278

3.2 Software Requirements:

1.	Arduino IDE
2.	Node JS
3.	Mongo DB

Table 3: Software Requirements

1. **Arduino IDE:**

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

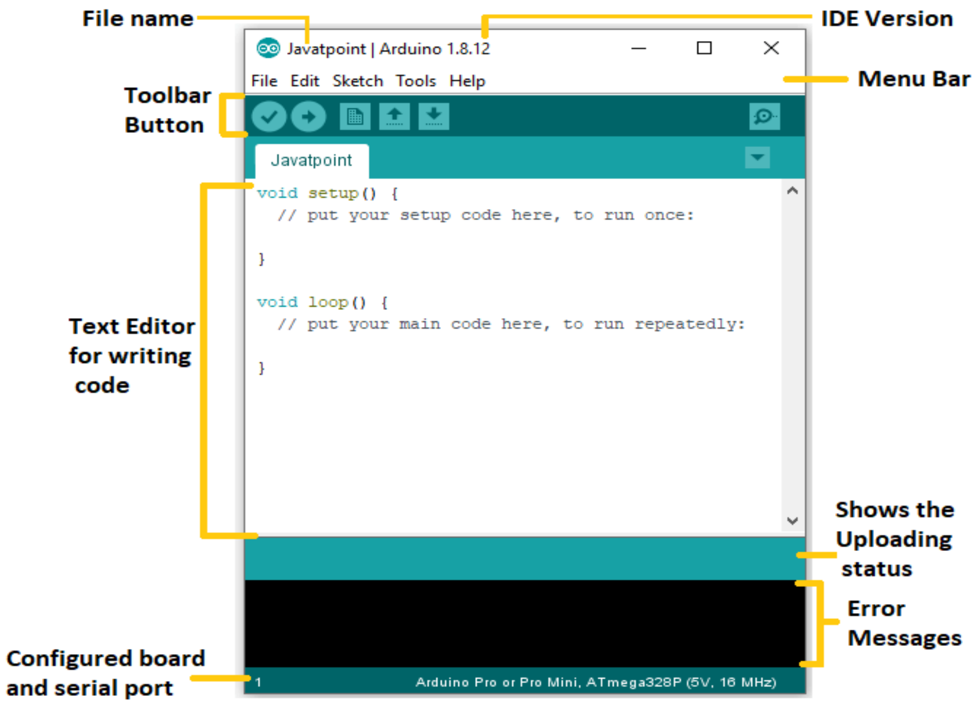


Figure 3.6: Arduino IDE

2. Node JS:

Node.js is a powerful and versatile platform that can be used to build a wide variety of applications. It is a good choice for developers who want to build fast, scalable, and reliable applications.

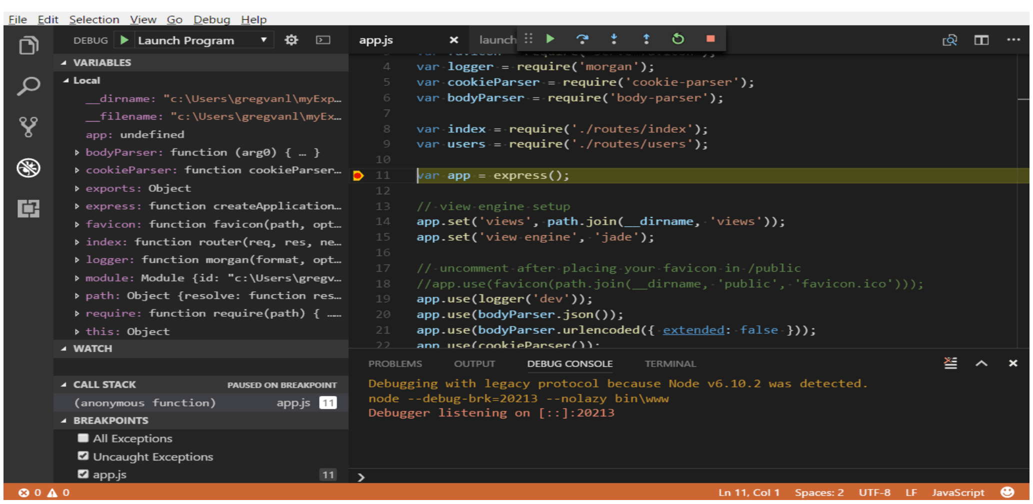


Figure 3.7: Node JS

3. Mongo DB:

MongoDB is a scalable, flexible NoSQL document database platform designed to overcome the relational databases approach and the limitations of other NoSQL solutions. MongoDB is well known for its horizontal scaling and load balancing capabilities, which has given application developers an unprecedented level of flexibility and scalability. MongoDB provides developers with a number of useful out-of-the-box capabilities, whether you need to run privately on site or in the public cloud.

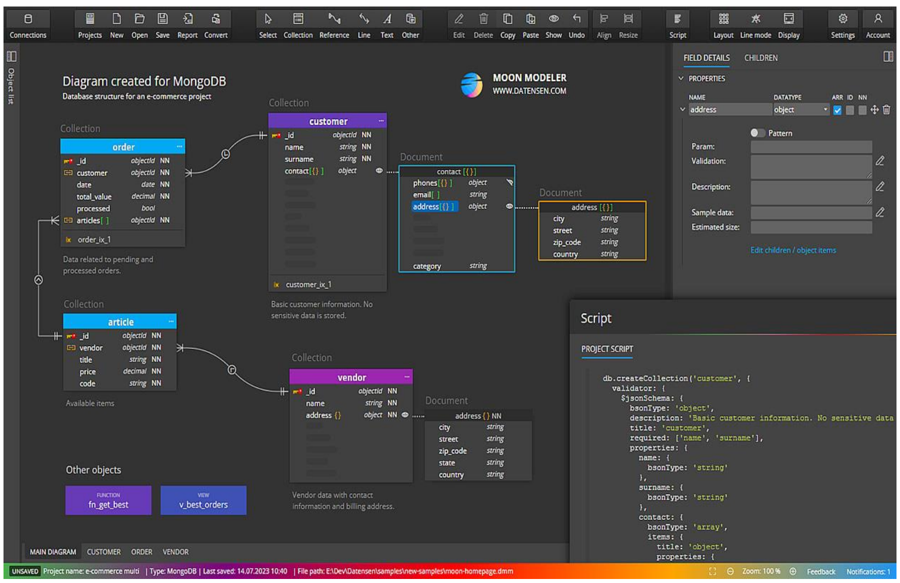


Figure 3.8: Mongo DB

3.3 Construction and Working:

The Digital Electric Meter is connected to Arduino UNO. This collects the data of number of units consumed by a household. Arduino UNO is connected to LoRa Module, it transmits the data collected by Arduino to ESP8266 which is connected to LoRa. ESP8266 is the Master Node or the Concentrator. ESP stores the data received, in the Database. The Telegram Bot fetches this data from the database when a user requests for a bill. The below Block diagram Figure 9 shows this basic overview.

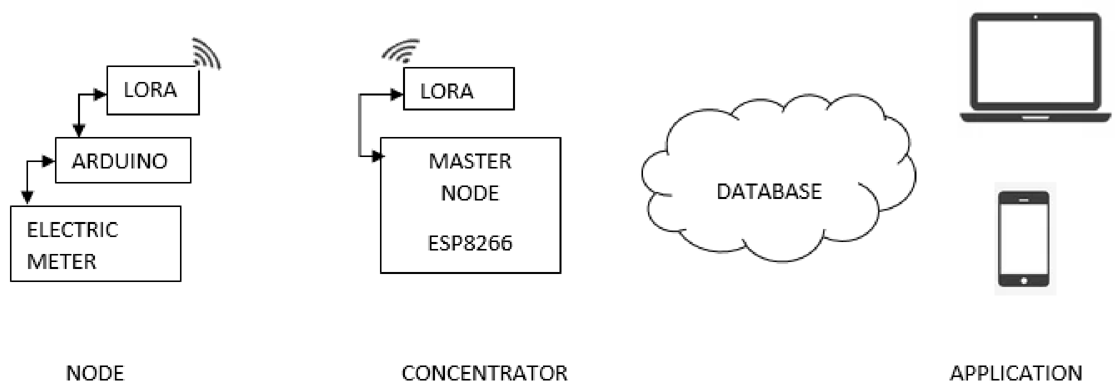


Figure 3.9: Block Diagram (Overview)

The System is divided into 3 Parts:

A. Acquiring Data from the Digital Meter.

B. Transmitting and Storing Data in Concentrator.

C. Billing Process.

A. Acquiring Data from the Digital Meter:

The Figure 10 explains the connection. The Digital Meter is connected to TTL Converter-The Tx of RJ10 cable is connected to the Rx pin of the TTL Converter and the Rx of RJ10 cable is connected to the Tx pin of the TTL Converter. TTL Converter is connected to Arduino UNO - The Tx of Arduino is connected to the Rx of TTL and The Rx of Arduino is connected to the Tx of TTL.

Figure 3.10: Block Diagram - A.

The present Digital meters work using the RS232 Protocol. Arduino UNO sends an acknowledgement signal to Electric Meter via TTL Converter. TTL Converter amplifies this signal and sends it to the Meter. When Meter recognizes the acknowledgement, it sends the data to the TTL Converter, which de-amplifies the signal and sends it to Arduino UNO. **Figure 11 shows this logic.**

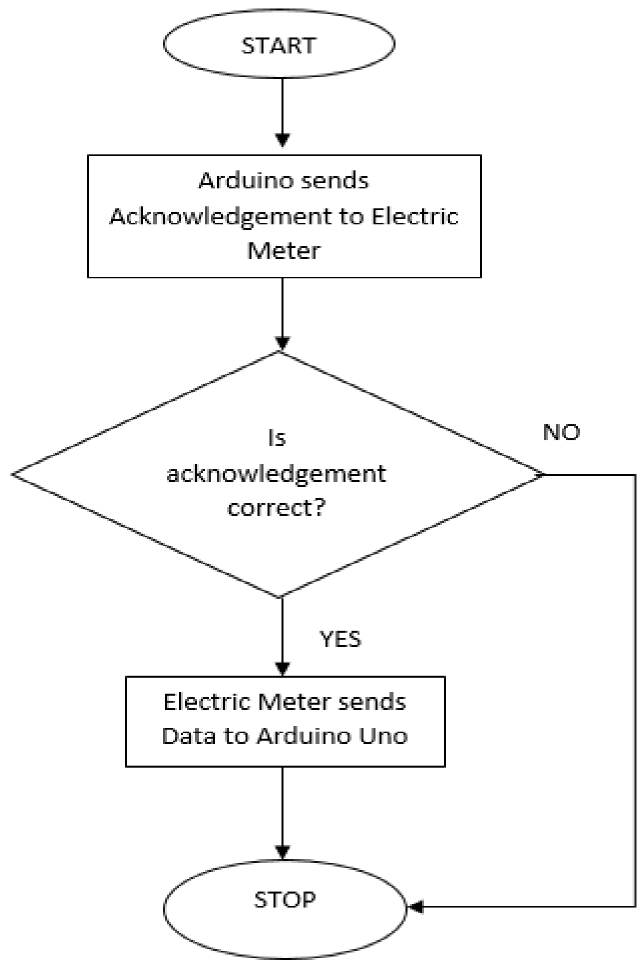


Figure 3.11: Flowchart – A.

B. Transmitting and Storing Data in Concentrator:

The data is present in Arduino UNO. Arduino UNO is the slave Node. It is connected to the LoRa Module - Tx of Arduino is connected to Rx of LoRa and vice versa. ESP8266 is the Wi-Fi module and is the Master Node. ESP8266 is connected to LoRa Module to Receive the Data sent by Slave Node. Tx of ESP is connected to Rx of LoRa and vice versa**. Figure 12 shows the block diagram.**

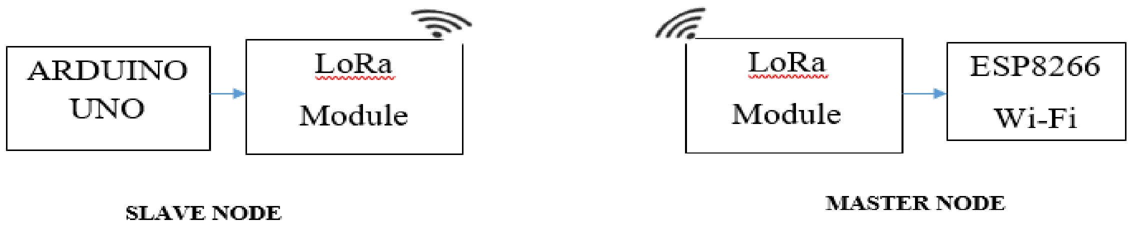


Figure 3.12: Block Diagram - B.

ESP8266 is connected to the API End Point or server through Wi-Fi. It receives the data i.e Home I.D, Mobile Number and the Units Consumed. Server checks if the Home I.D and Mobile Number sent is present in the Database. If it exists, the value of number of units is updated in the database as **shown below in Figure 13.**

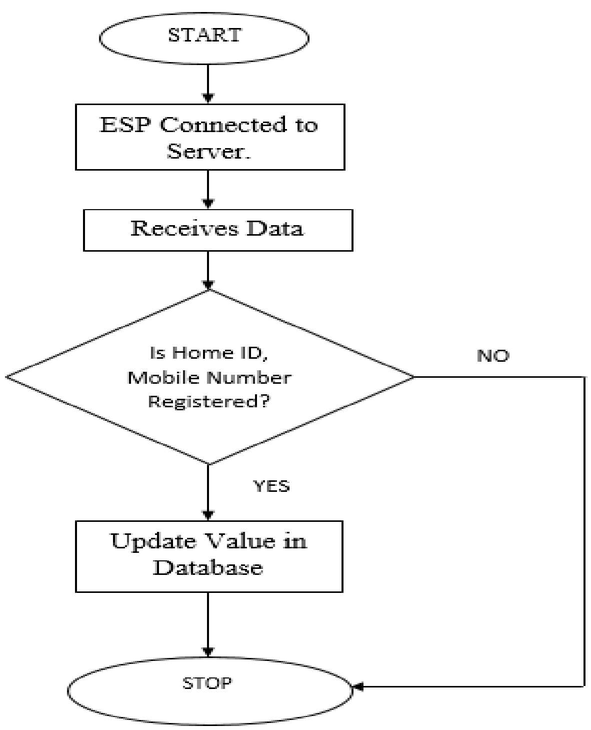


Figure 3.13: Flow chart - B.

The Master Node sends five requests to the Slave. Five requests sent in order to eliminate data loss or errors. The Slave responds to these requests by sending the Data. Once the Master receives the data, it updates the value in the Database.

C. Billing Process:

The Data is stored is now stored in the database by ESP8266. In order to ease the billing process we have created a Telegram Bot, which provides us the bill according to the amount of electricity consumed by a household. Telegram Bot is created using Node JS. Telegram Bot created fetches the data of electricity consumed from the Database.

- **Bill calculation:**
  0-50 units: 4.15 rupees/unit
  51-100 units: 5.60 rupees /unit
  101-200 units: 7.15 rupees /unit
  Above 200 units: 8.20 rupees /unit

Chapter 4

RESULT ANALYSIS

The results obtained for every stage of the project is shown here.

A. Acquiring Data from the Digital Meter:

The Arduino collects data from the Digital Meter through the RJ10 Cable.

B. Transmitting and Storing Data in Concentrator:

The Data collected by the Arduino is sent to ESP8266. The below Figure 14 shows the response of Arduino to the Master. Arduino sends the data to Master once it receives a request from the Master.

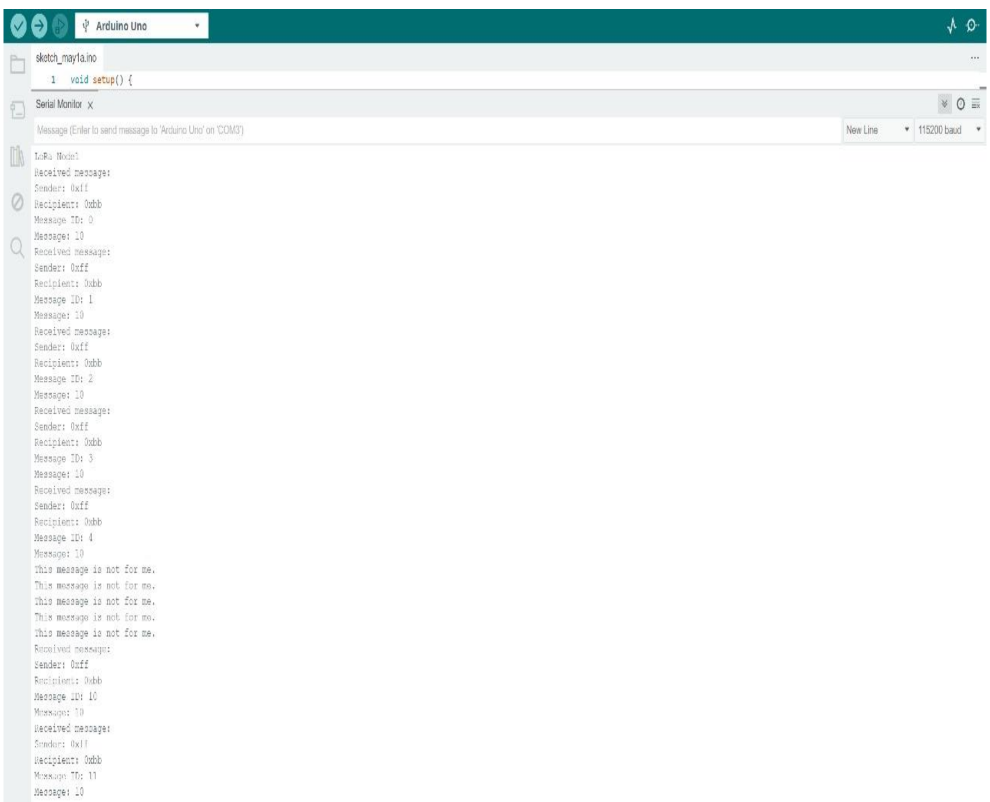


Figure 4.14: Arduino Response

A Webpage for Master Node is created using Node JS. This webpage stores the information of every user or node – i.e Home ID, Mobile Number, Payment Status and the Units Consumed. These are the nodes connected to that particular Master Node. We can add or delete nodes and update the details of the users in the webpage. **Figure 15 shows the Webpage created.**

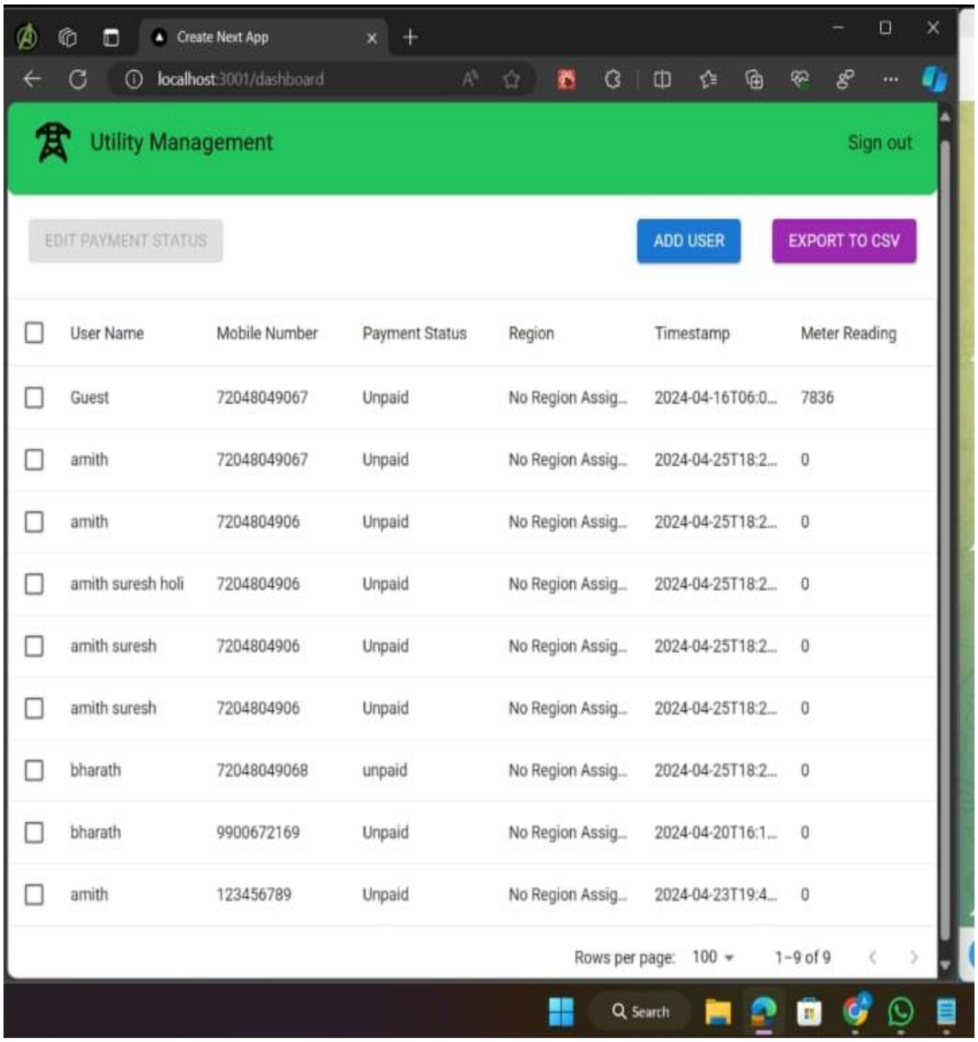


Figure 4.15: Webpage for Concentrator

C. Billing Process:

Telegram Bot is created. This Bot fetches data from the database and responds by sending the Meter Reading, Date and bill. The person is first asked to enter the registered Mobile Number. After the Mobile number is entered, Bot checks if the mobile Number is correct i.e stored in the database. If the Mobile Number is present then the Bot Fetches the data of Meter Reading and Calculates the Bill for it. The Invoice for the Bill is generated and sent to the User or Costumer. **Figure 16 shows the Telegram Bot working.**

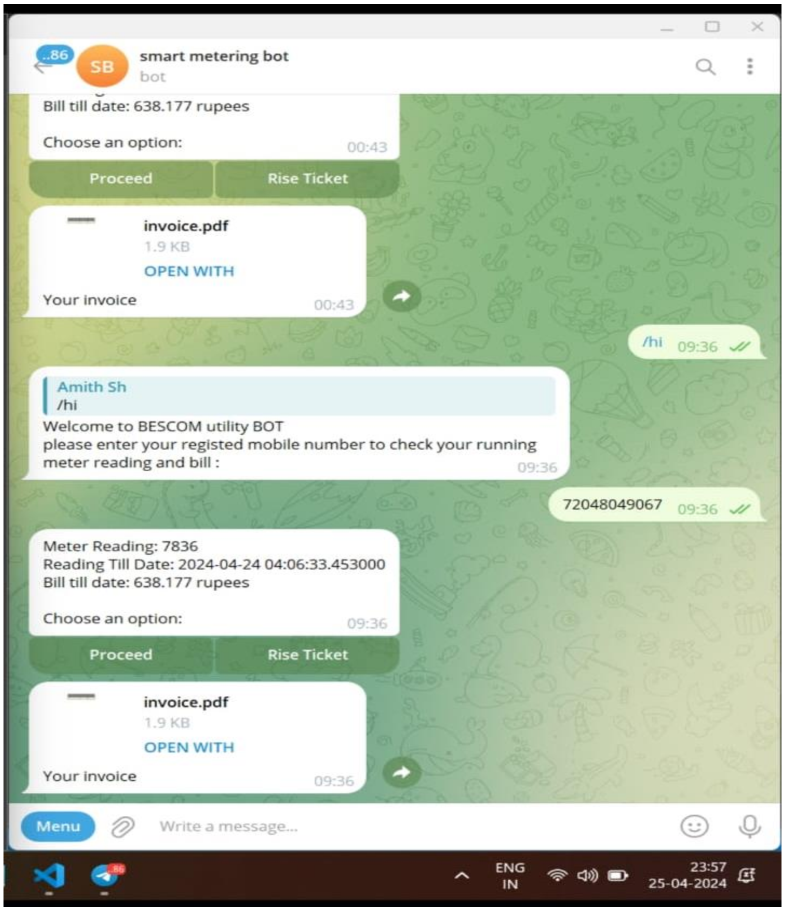


Figure 4.16: Telegram Bot for Billing

Chapter 5

CONCLUSION & FUTURE SCOPE

This is a real time use of Internet of Things and LoRa Technology to ease the billing process of Electric Meter. The project is relevant to society and the Government's Electricity Board. It helps in reducing the budget for the electricity board as we reduce human intervention during the billing process. Through this process, Costumers can optimize the consumption of electricity. It helps in making our society or city Smart with the use of new technologies.

REFERENCES

1. Enrique Rodriguez-Diaz, Emilio J. Palacios-Garc´ıa et all , “Advanced Smart Metering Infrastructure for Future Smart Homes”, IEEE 5th International Conference on Consumer Electronics Berlin (ICCE-Berlin), 2015, Pg No 29-31,

2. C.M. Lin, M.T. Chen, “Design and Implementation of a Smart Home Energy Saving System with Active Loading Feature Identification and Power Management” , IEEE , 2017 , Pg No : 739-742

3. Md Abul Hayat Shahjee Midul et all, “Design and Implementation of IoT-Based Smart Energy Meter to Augment Residential Energy Consumption ” , 3rd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2023, Page 106-110

4. Lailis Syafa’ah, Agus Eko Minarno, Fauzi Dwi Setiawan SumadiDwi Anggraini Puspita Rahayu, “ESP 8266 For Control And Monitoring In Smart Home Application”, 2019 IEEE , Pg 123-128

5. Shruti Dash, Pallavi Choudekar, “Home Automation using Smart Devices and IoT” , 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021

6. Vishnukant V. Gavhane, Mayuri R. Kshirsagar et all, “IoT based Energy Meter with Smart Monitoring of Home Appliance” , 6th International Conference for Convergence in Technology (I2CT), 2021 , Pg 1-5

7. Ayodeji Olalekan Salau, Lekhika Chettri, Tshering Kiden Bhutia, Mayalmit Lepcha, “IoT Based Smart Digital Electric Meter for Home Appliances”, International Conference on Decision Aid Sciences and Application (DASA),2020,Pg 708-713

8. Ciprian Ionut PAUNESCU, Lucian TOMA, Mircea EREMIA, “Laboratory Smart Home Energy Management System”

9. Dr.Sabout Nagaraju, "LORAWAN PROTOTYPE FOR SMART HOME VULNERABILITIES AND THREATS INVESTIGATION" , Winter Summit on Smart Computing and Networks (WiSSCoN) , 2023

10. Tanmay Arya, D. Saxena, "Optimizing Electricity Bill, Grid Performance, and User Satisfaction in a Comprehensive Home Energy Management System with Electric Vehicles and Home-to-Grid Capabilitie" , International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM) , 2023

11. Soeharwinto,M.Fadhlan Ariska,Baihaqi Siregar, Fahmi Fahmi, " Power Meter Monitoring for Home Appliances Based on Mobile Data Communication" , International Conference on Smart Cities, Automation & Intelligent Computing Systems,Yogyakarta, Indonesia, November 08-10, 2017,Pg 118-120.

12. Selvam. P , Ramakrishnaprabu G et all, "Intrusive Energy Management with advanced smart metering and monitoring using IoT", Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA),2022, Pg 359-364

13. Sadık Yıldız, Mustafa Burunkaya, "Web Based Smart Meter for General Purpose Smart Home Systems with ESP8266" , 2019 IEEE.

14. Yong Zhao, Wanxing Sheng, Junping Sun, Weijun Shi , "Research and Thinking of Friendly Smart Home Energy System Based on Smart Power" , IEEE, 2011

15. Seema P N, M P Suresh, Dr. Manjula G Nair, "Smart Energy Management System for the Optimal Control of Loads to Reduce the Burden on The Grid" , IEEE 3rd Global Conference for Advancement in Technology (GCAT) , 2022

APPENDIX -A

ARDUINO UNO

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Inout 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 bootloader
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Clock Speed: 16 MHz
- LED_BUILTIN: 13
- Length: 68.6 mm
- Width: 58.4 mm
- Weight: 25 g

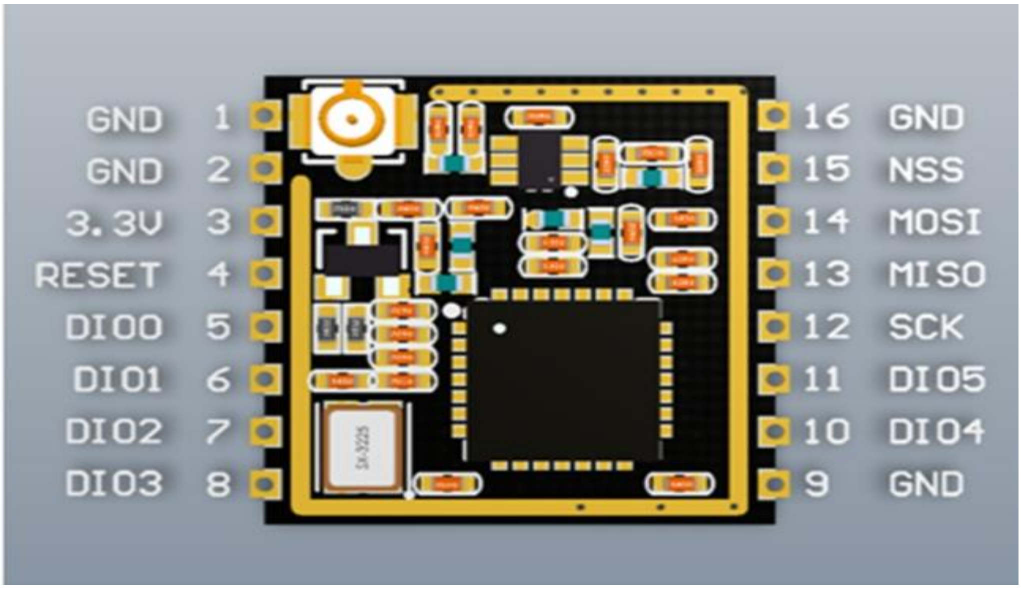
APPENDIX-B

ESP8266

	Official NodeMCU	NodeMCU Carrier Board	LoLin NodeMCU
Microcontroller	ESP-8266 32-bit	ESP-8266 32-bit	ESP-8266 32-bit
NodeMCU Model	Amica	**Amica**	Clone LoLin
NodeMCU Size	49mm x 26mm	49mm x 26mm	58mm x 32mm
Carrier Board Size	n/a	102mm x 51mm	n/a
Pin Spacing	**0.9" (22.86mm)**	**0.9" (22.86mm)**	1.1" (27.94mm)
Clock Speed	80 MHz	80 MHz	80 MHz
USB **to Serial**	CP2102	**CP2102**	CH340G
USB Connector	Micro USB	Micro USB	Micro USB
Operating Voltage	3.3V	3.3V	3.3V
Input Voltage	4.5V-10V	4.5V-10V	4.5V-10V
Flash Memory/SRAM	4 MB / 64 KB	4 MB / 64 KB	4 MB / 64 KB
Digital I/O Pins	11	11	11
Analog In Pins	1	1	1
ADC Range	0-3.3V	0-3.3V	0-3.3V
UART/SPI/I2C	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1
WiFi Built-In	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n
Temperature Range	-40C - 125C	-40C - 125C	-40C - 125C
Product Link		NodeMCU	NodeMCU

APPENDIX-C

LoRa Module – SX1278



Features

- LoRa ™ Spread Spectrum modulation technology
- Constant RF power output at + 20dBm-100mW voltage change
- Half-duplex SPI communication
- Supports FSK, GFSK, MSK, GMSK, LoRa ™ and OOK modulation modes
- Automatic RF signal detection, CAD mode and very high speed AFC
- Packet engine with CRC up to 256 bytes
- Small footprint dual-row stamp-hole patch package
- Shielded housing
- Spring Antenna

Specifications

- Communication distance: 15KM
- Sensitivity: down to -148dBm
- Programmable bit rates: up to 300kbps
- RSSI dynamic range: 127dB
- Wireless frequency: 433MHz
- Working voltage: 1.8-3.7v
- Working temperature: -40-+80 ℃

APPENDIX-D

MAX232 IC

4 Pin Configuration and Functions

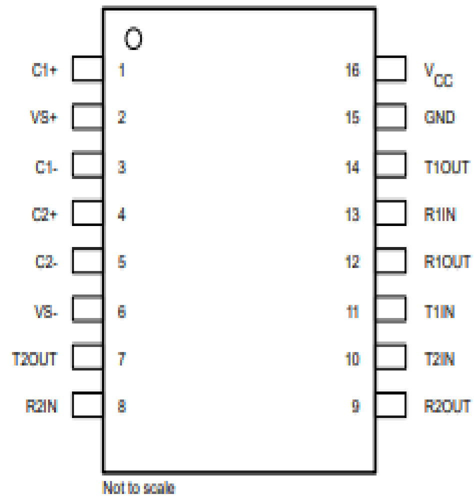


Figure 4-1. MAX232: D, DW, N or NS Package
MAX232I: D, DW, or N Package
(Top View)

Table 4-1. Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
C1+	1	—	Positive lead of C1 capacitor
VS+	2	O	Positive charge pump output for storage capacitor only
C1-	3	—	Negative lead of C1 capacitor
C2+	4	—	Positive lead of C2 capacitor
C2-	5	—	Negative lead of C2 capacitor
VS-	6	O	Negative charge pump output for storage capacitor only
T2OUT	7	O	RS232 line data output (to remote RS232 system)
R2IN	8	I	RS232 line data input (from remote RS232 system)
R2OUT	9	O	Logic data output (to UART)
T2IN	10	I	Logic data input (from UART)
T1IN	11	I	Logic data input (from UART)
R1IN	13	I	RS232 line data input (from remote RS232 system)
T1OUT	14	O	RS232 line data output (to remote RS232 system)
GND	15	—	Ground
V_{CC}	16	—	Supply Voltage, Connect to external 5V power supply

ANNEXURE

BUDGET OF THE PROJECT:

Sl.NO	COMPONENTS / ITEMS	COST
1.	Arduino UNO	400
2.	ESP8266	350
3.	LoRa Module-Sx1278 (2)	600
4.	TTL Converter	150
5.	Foam Board	200
6.	Cardboard	150
7.	Miscellaneous	250
	TOTAL	**2,100**