AUTOBILL

An SDC (IoT) mini project report submitted in partial fulfillment of therequirement for the Award of the Degree of: AUTOBILL

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

by

Mustafa Mohammed Meraj 160722733076
Mir Ayan Ali 160722733083
Motassim Meraj Khan 160722733098

Under the Guidance of
Dr. Shaik Khaleel Ahamed, Associate Professor, Dept. of CSE

Department of Computer Science and Engineering
Methodist College of Engineering and Technology,
King Koti, Abids, Hyderabad-500001.

2023-2024

METHODIST
College of Engineering & Technology
(Approved by AICTE, New-Delhi & Affiliated to Osmania University)
College Code : 1607

Dr. Prabhu G Benakop
B.E., M.E., Ph.D.
SM., IEEE, LMISTE, LMISOI
Principal

VISION

To produce ethical, socially conscious and innovative professionals who would contribute to sustainable technological development of the society.

MISSION

- To impart quality engineering education with latest technological developments and interdisciplinary skills to make students succeed in professional practice.
- To encourage research culture among faculty and students by establishing state of art laboratories and exposing them to modern industrial and organizational practices.
- To inculcate humane qualities like environmental consciousness, leadership, social values, professional ethics and engage in independent and lifelong learning for sustainable contribution to the society.

PRINCIPAL

PRINCIPAL
METHODIST COLLEGE OF ENGG.& TECH.
King Koti Road, Abids, Hyderabad.

King Koti Road, Abids
Hyderabad - 500 001. T.S. India
Ph : 040-24753445/24755999
www.methodist.edu.in

METHODIST COLLEGE OF ENGINEERING & TECHNOLOGY

Affiliated to Osmania University, - College Code – 1607

Department of Computer Science & Engineering

VISION

To become a leader in providing Computer Science & Engineering education with emphasis on knowledge and innovation.

MISSION

- To offer flexible programs of study with collaborations to suit industry needs
- To provide quality education and training through novel pedagogical practices
- To expedite high performance of excellence in teaching, research and innovations.
- To impart moral, ethical valued education with social responsibility.

Head of the Department
Department of CSE
Methodist College of Engg & Tech
Abids, Hyderabad.

Department of Computer Science & Engineering

PROGRAM OUTCOMES

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in

Program Specific Outcomes

At the end of 4 years, Compute Science and Engineering graduates at MCET will be able to:

PSO1: Apply the knowledge of Computer Science and Engineering in various domains like networking and data mining to manage projects in multidisciplinary environments.

PSO2: Develop software applications with open-ended programming environments.

PSO3: Design and develop solutions by following standard software engineering principles and implement by using suitable programming languages and platforms

METHODIST COLLEGE OF ENGINEERING & TECHNOLOGY

Affiliated to Osmania University, - College Code – 1607

Department of Computer Science & Engineering

Program Educational Objectives

Graduates of Compute Science and Engineering at Methodist College of Engineering and Technology will be able to:

PEO1: Apply technical concepts, Analyze, Synthesize data to Design and create novel products and solutions for the real life problems.

PEO2: Apply the knowledge of Computer Science Engineering to pursue higher education with due consideration to environment and society.

PEO3: Promote collaborative learning and spirit of team work through multidisciplinary projects

PEO4: Engage in life-long learning and develop entrepreneurial skills.

Head of the Department
Department of CSE
Methodist College of Engg. & Tech
Abids, Hyderabad.

METHODIST
COLLEGE OF ENGINEERING & TECHNOLOGY
(An UGC-AUTONOMOUS INSTITUTION)
Accredited by NAAC with A+ and NBA
Affiliated to Osmania University & Approved by AICTE
Estd : 2008

Department of Computer Science and Engineering

Skill Development Course IoT MINI PROJECT

(3PW354CS)

A.Y 2023-2024

This is to certify that this SDC(IoT) Mini project report entitled **"AUTOBILL",** being submitted *by:*

Mustafa Mohammed Meraj 160722733076
Mir Ayan Ali 160722733083
Motassim Meraj Khan 160722733098

submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering, during the academic year 2023-2024, is a bonafide record of work carried out by them.

INTERNAL **EXTERNAL** **HOD**

METHODIST

COLLEGE OF ENGINEERING & TECHNOLOGY

(An UGC-AUTONOMOUS INSTITUTION)

Accredited by NAAC with A+ and NBA

Estd : 2008

Affliated to Osmania University & Approved by AICTE

Department of Computer Science and Engineering

DECLARATION BY THE CANDIDATES

We, **MUSTAFA MOHAMMED MERAJ (160722733076), MIR AYAN ALI (160722733083), MOTASSIM MOHAMMED MERAJ KHAN (160722733098), students** of Methodist College of Engineering and Technology, pursuing Bachelor's degree in Computer Science and Engineering, hereby declare that SDC(IoT) Mini project report entitled "**AUTOBILL**" carried out under the guidance of **Dr. Shaik Khaleel Ahamed** submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science. This work is carried out by us and the references have been taking from various digital resources for report preparation.

METHODIST
COLLEGE OF ENGINEERING & TECHNOLOGY
(An UGC-AUTONOMOUS INSTITUTION)
Accredited by NAAC with A+ and NBA
Affliated to Osmania University & Approved by AICTE
Estd : 2008

Department of Computer Science and Engineering

CERTIFICATE BY THE SDC(IOT) LAB INCHARGE

This is to certify that this SDC (IoT) Mini project report entitled **"AUTOBILL"** being submitted *by* **MUSTAFA MOHAMMED MERAJ (160722733076), MIR AYAN ALI (160722733083), MOTASSIM MOHAMMED MERAJ KHAN (160722733098)** submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering, during the academic year 2023-2024, is a bonafide record of work carried out by them.

Dr. Shaik Khaleel Ahamed
Associate Professor,
Dept. of CSE

METHODIST
COLLEGE OF ENGINEERING & TECHNOLOGY
(An UGC-AUTONOMOUS INSTITUTION)
Accredited by NAAC with A+ and NBA
Estd : 2008
Affliated to Osmania University & Approved by AICTE

Department of Computer Science and Engineering

CERTIFICATE BY THE HEAD OF THE DEPARTMENT

This is to certify that this SDC (IoT) Mini project report entitled AUTOBILL *by*

Mustafa Mohammed Meraj	**160722733076**
Mir Ayan Ali	**160722733083**
Motassim Meraj Khan	**160722733098**

submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering of the Osmania University, Hyderabad, during the academic year 2023-2024, is a bonafide record of work carried out by them.

Dr. P. Lavanya,
Professor &
Head of the Department

ACKNOWLEDGEMENT

We would like to express a deep sense of gratitude towards the **Dr. Lakshmipathi Rao, Director, Methodist College of Engineering and Technology,** for always being an inspiration and for always encouraging us in every possible way.

We would like to express a deep sense of gratitude towards the **Dr. Prabhu G Benakop, Principal, Methodist College of Engineering and Technology,** for always being an inspiration and for always encouraging us in every possible way.

Our sincere thanks to **Dr. P. Lavanya, Professor** and **Head of the Department of Computer Science and Engineering,** for her valuable guidance and encouragement which has played a major role in the completion of the project and for helping us by being an example of high vision and pushing towards greater limits of achievement.

We would like to express our sincere gratitude to our project guide **Dr. Shaik Khaleel Ahmed, Associate Professor,** for giving us the opportunity to work on this topic. It would never be possible for us to take this project to this level without his innovative ideas and his relentless support and encouragement. Who helped us by being an example of high vision and pushing towards greater limits of achievement.

We are indebted to the Department of Computer Science & Engineering and Methodist College of Engineering and Technology for providing us with all the required facility to carry our work in a congenial environment. We extend our gratitude to the CSE Department staff for providing us to the needful time to time whenever requested.

We would like to thank our parents for allowing us to realize our potential, all the support they have provided us over the years was the greatest gift anyone has ever given us and also for teaching us the value of hard work and education. Our parents have offered us with tremendous support and encouragement, thanks to our parents for all the moral support and the amazing opportunities they have given us over the years.

Internet of Things (IoT):

Introduction:

The Internet of Things, commonly known as IoT, refers to the network of interconnected devices, objects, and systems that communicate and share data over the internet. These devices, equipped with sensors, actuators, and connectivity features, enable them to collect, exchange, and act upon information, creating a smart and interconnected ecosystem.

Fig 1.1: iot architecture

Key Components:

Devices and Sensors: IoT devices vary widely, spanning from consumer electronics like smart thermostats and wearables to industrial equipment such as sensors in manufacturing plants. These devices employ an assortment of sensors, including temperature, humidity, motion, and proximity sensors. For instance, in a smart agriculture setup, soil moisture sensors and drones equipped with imaging sensors can provide real-time data for precision farming.

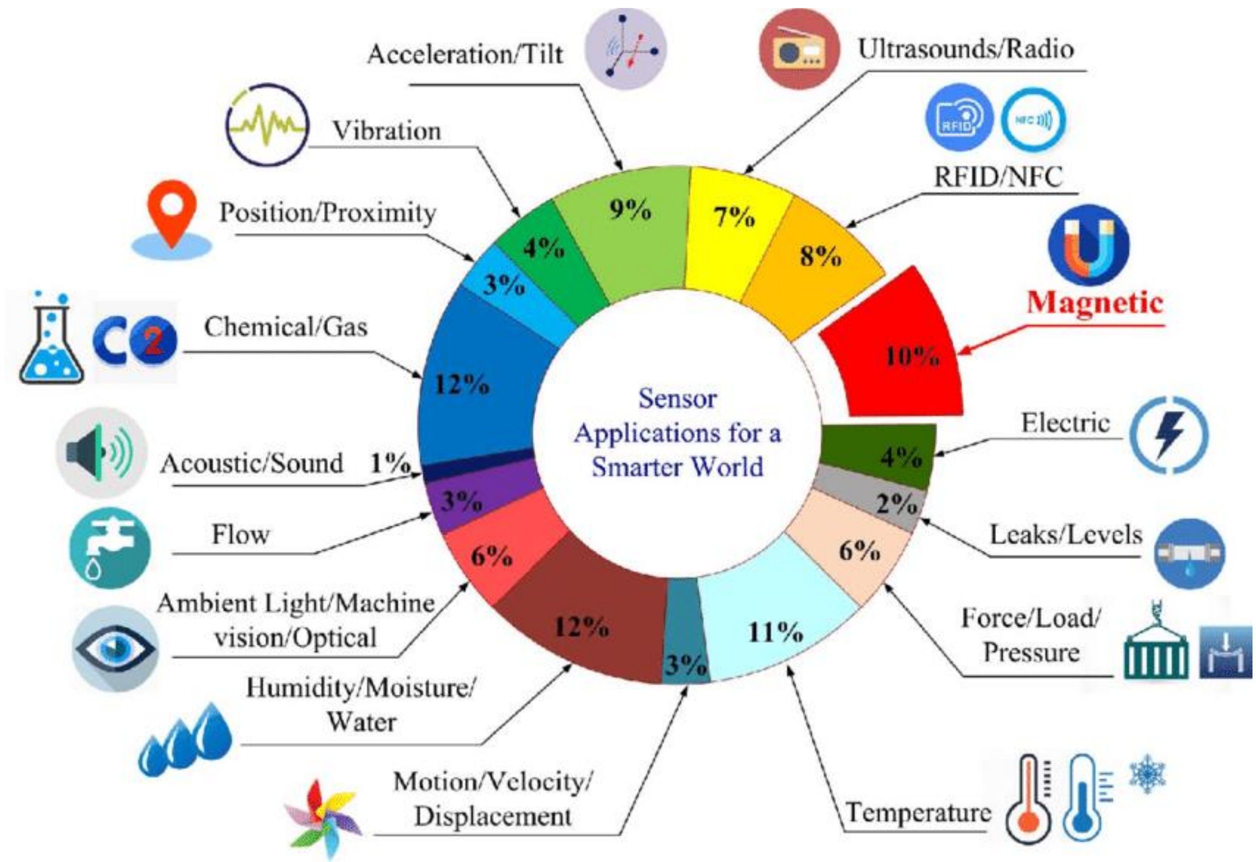


Fig 1.2: sensor applications for a smarter world

Connectivity: The choice of connectivity depends on the application's requirements. Wi-Fi and Ethernet are common in homes and offices, while Low Power Wide Area Networks (LPWANs) like LoRa and NB-IoT are suitable for long-range, low-power applications. In industrial settings, Wired HART and Wireless HART are employed for process automation, ensuring robust and reliable connectivity in challenging environments.

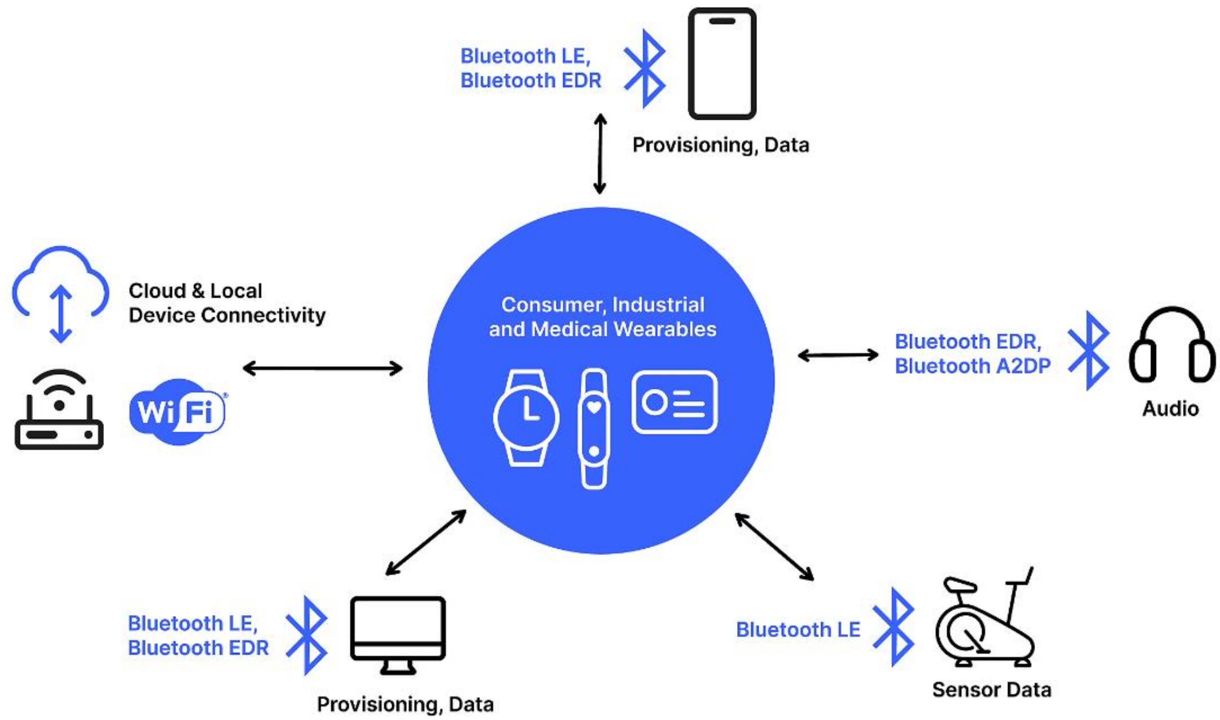


Fig 1.3: Consumer, Industrial and medical wearables showcasing connectivity

Data Processing and Analytics: The enormous volume of data generated by IoT devices necessitates sophisticated processing and analytics. Cloud platforms like AWS IoT, Azure IoT, and Google Cloud IoT offer scalable solutions for storing and analyzing data. Edge computing, on the other hand, involves processing data closer to the source, reducing latency and enabling real-time decision-making. This is particularly critical in applications like autonomous vehicles, where split-second decisions are essential for safety.

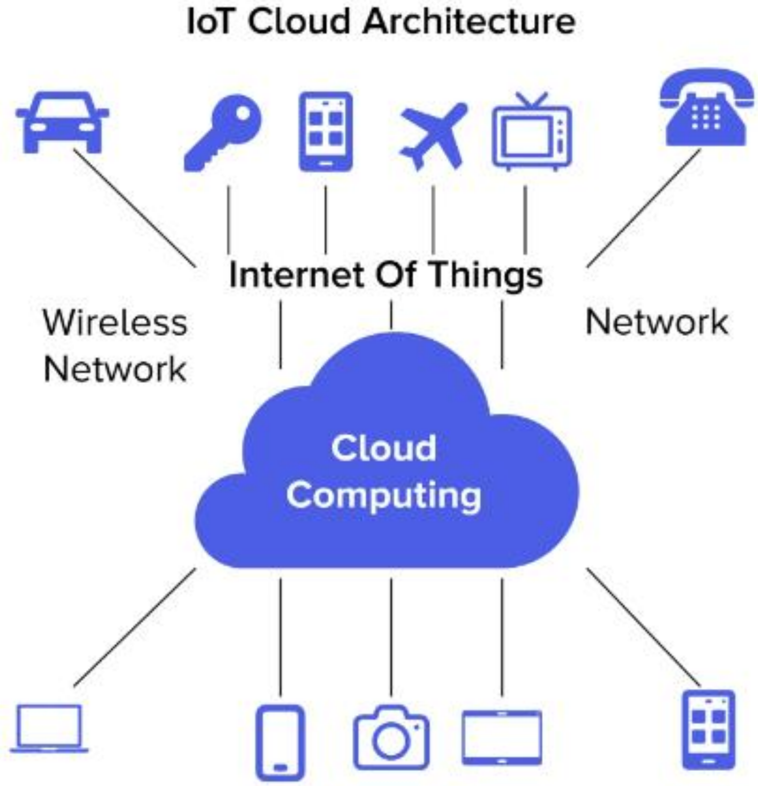


Fig 1.4: Cloud computing in iot

Actuators: Actuators in IoT devices enable tangible responses to data insights. In smart buildings, actuators control HVAC systems, adjusting temperature and airflow based on occupancy and environmental conditions. In industrial automation, actuators are integral to robotic systems, ensuring precise movements and responses to changing conditions.

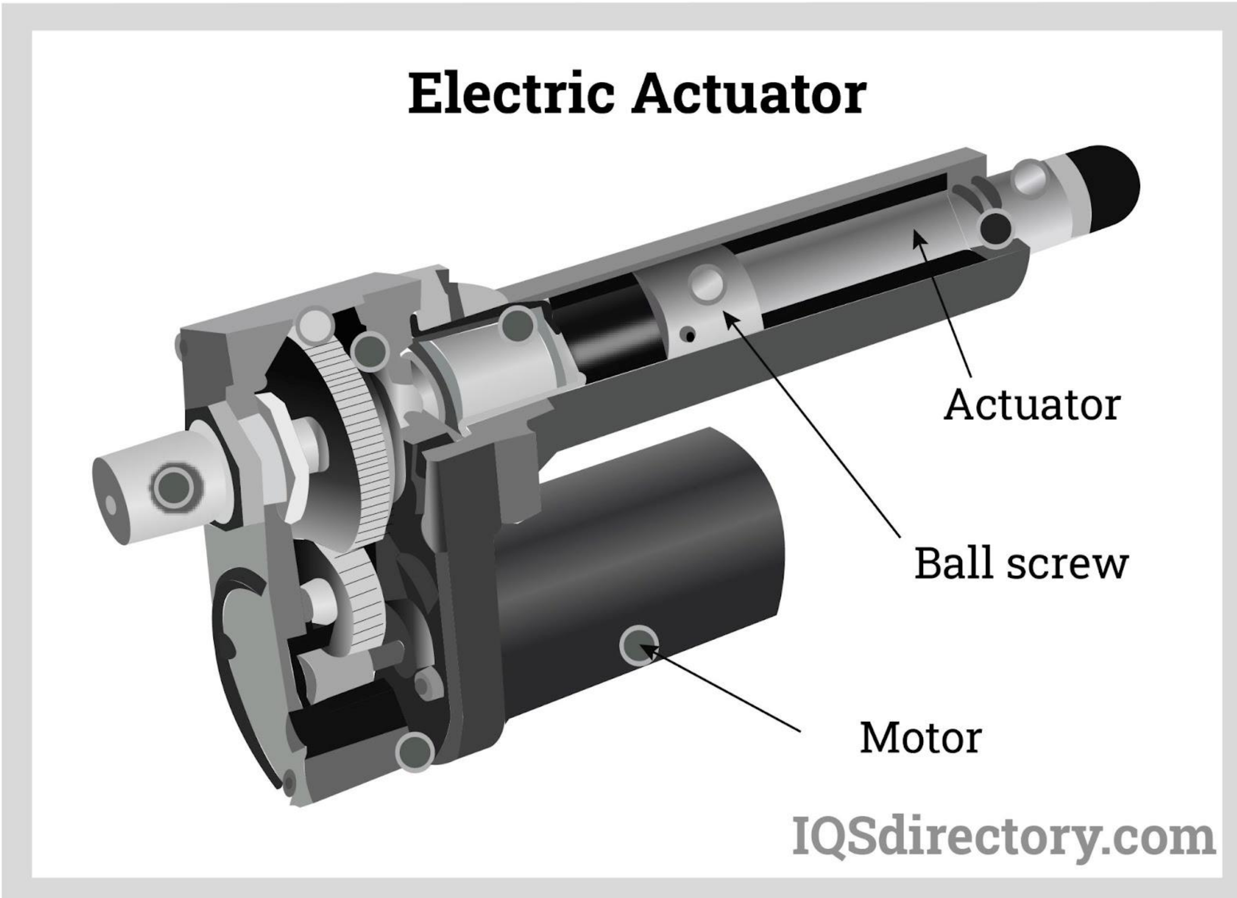


Fig 1.5: Electric Actuator

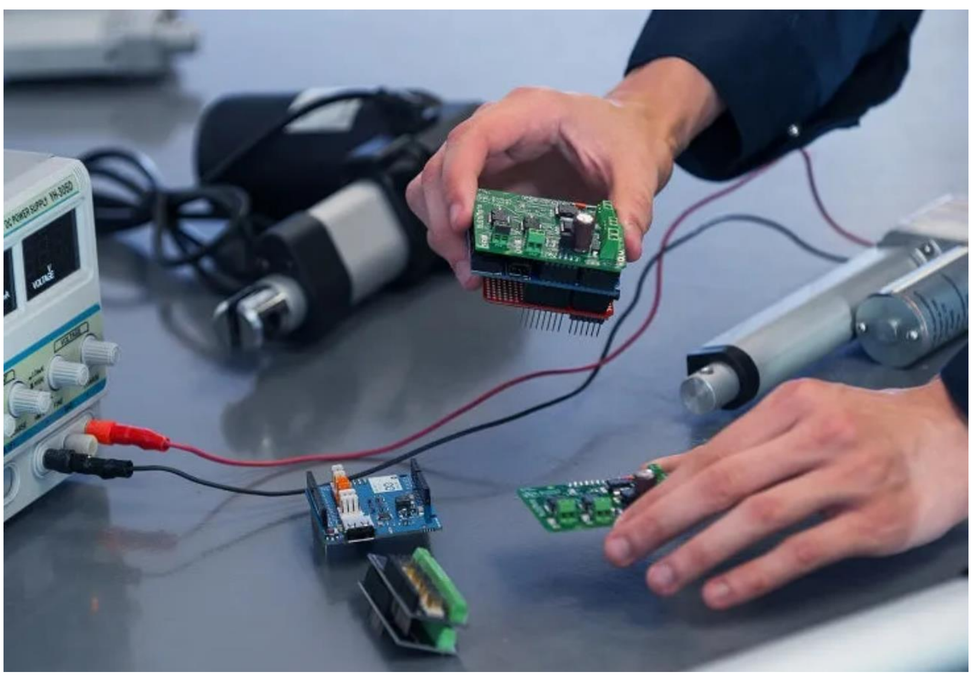


Fig 1.6: Actuator Circuit

Applications:

Smart Homes: The evolution of smart homes involves interconnected ecosystems. Smart thermostats not only regulate temperature but also learn user preferences over time. Integration with voice assistants like Amazon Alexa or Google Assistant enables seamless control of various devices. Smart security systems utilize cameras and motion sensors, sending alerts and providing remote monitoring capabilities.

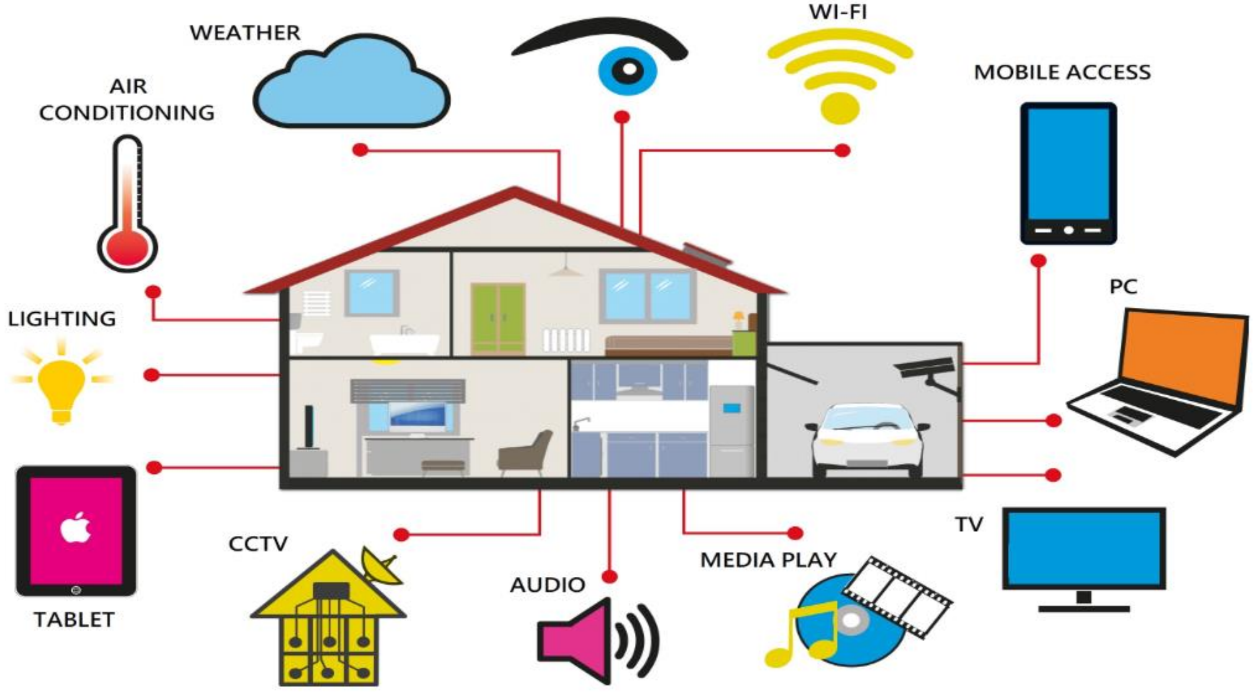


Fig 1.7: Smart Home

Industrial IoT (IIoT): IIoT transforms traditional industries through predictive maintenance and process optimization. In manufacturing, sensors on machinery monitor performance metrics in real-time. Predictive analytics forecast potential equipment failures, allowing for proactive maintenance and minimizing downtime. Supply chain visibility is enhanced, with RFID and GPS tracking ensuring efficient logistics.

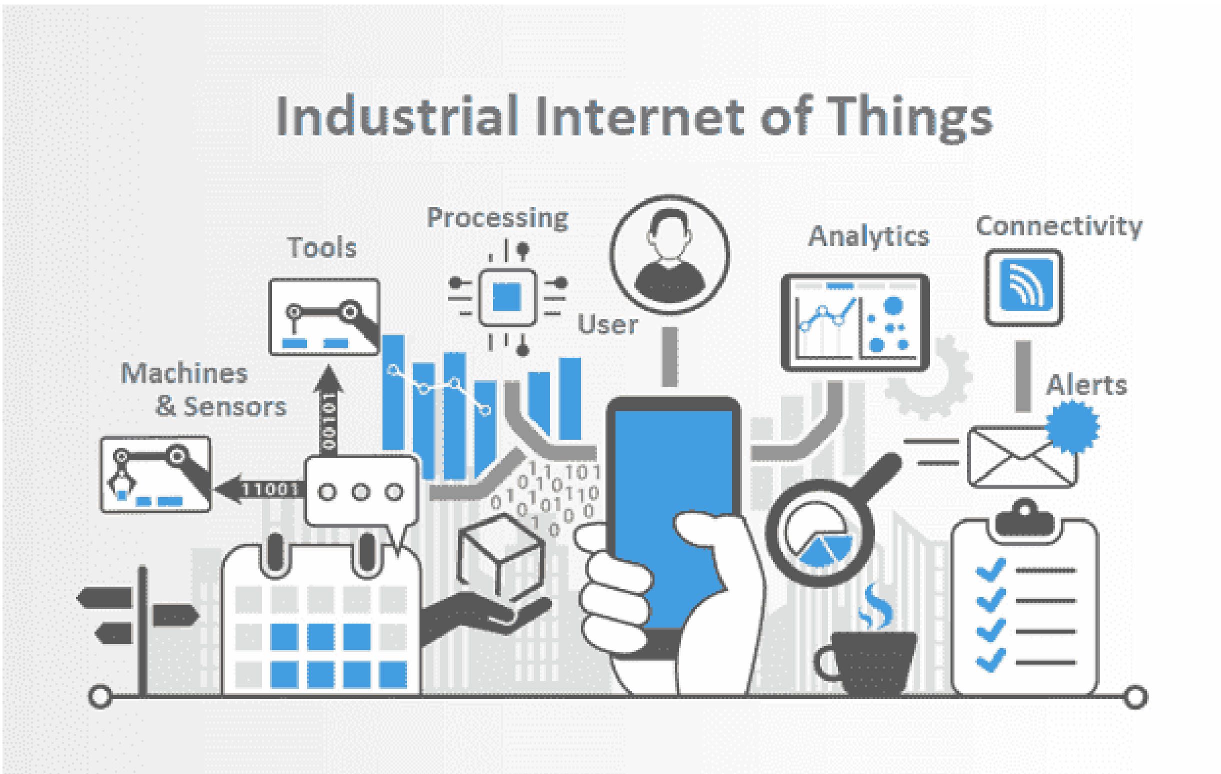


Fig 1.8: Industrial Internet Of Things (iiot)

Healthcare: The healthcare sector benefits from wearables monitoring vital signs and tracking health metrics. Continuous glucose monitors for diabetics, smart inhalers for respiratory conditions, and remote patient monitoring devices enable personalized and proactive healthcare. IoT also facilitates the integration of Electronic Health Records (EHRs) for streamlined and accessible patient information.

Fig 1.9: iot in healthcare

Smart Cities: The concept of smart cities incorporates diverse applications. Intelligent traffic management systems use IoT sensors to monitor traffic flow and optimize signal timings, reducing congestion. Waste management systems leverage sensors in bins to optimize collection routes, reducing costs and environmental impact. Public safety is enhanced through smart street lighting and video surveillance.

Fig 1.10: smart city

Challenges and Considerations:

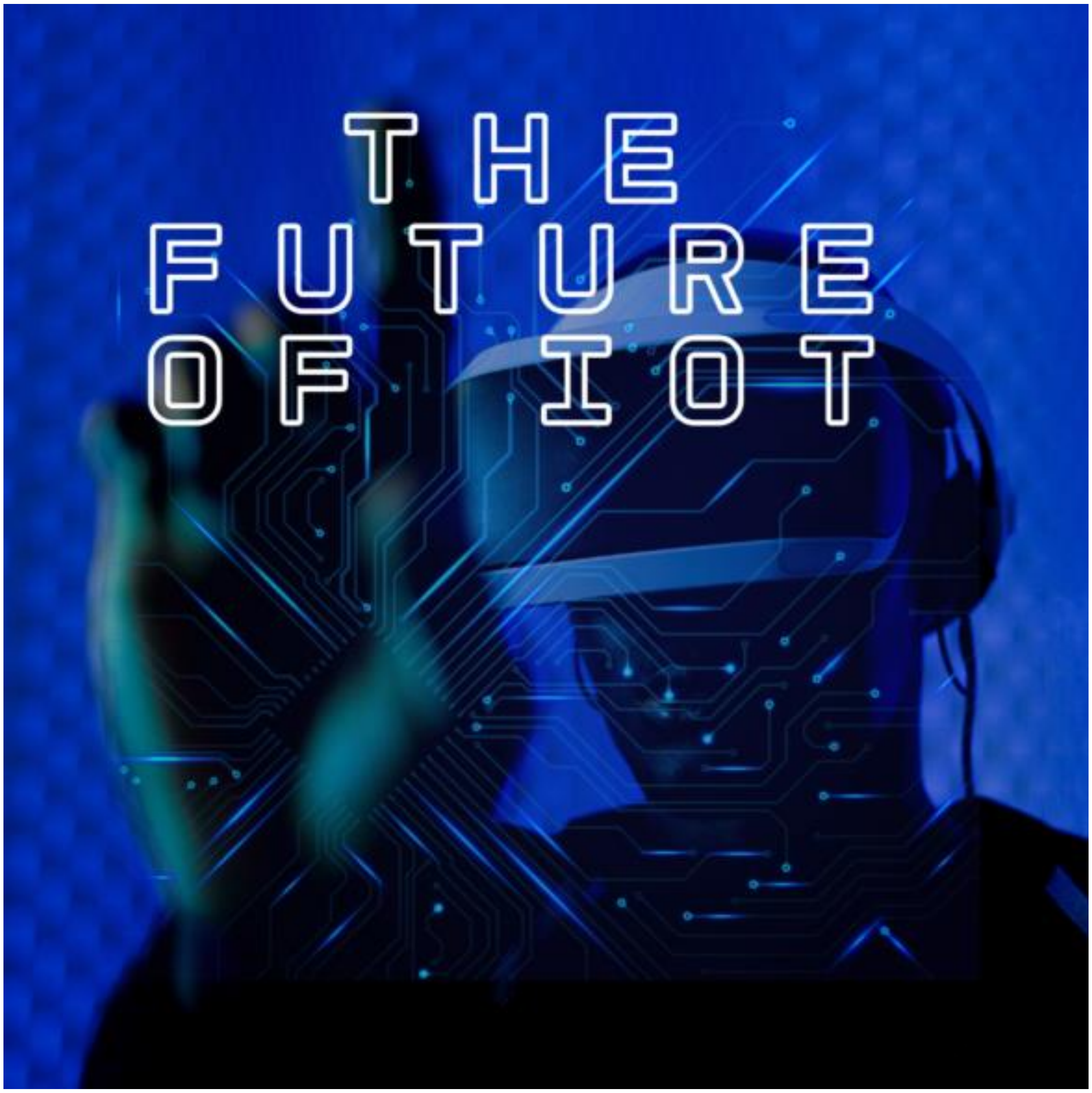
Security: IoT security involves safeguarding devices, networks, and data. Implementing strong encryption protocols, securing device firmware, and regularly updating software are essential. Additionally, ensuring secure device onboarding and authentication prevents unauthorized access.

Interoperability: Interoperability challenges arise due to the proliferation of diverse devices and communication protocols. Standardization efforts, such as those by industry consortia and standards organizations, aim to establish common frameworks, ensuring seamless communication between devices from different manufacturers.

Scalability: Scalability is crucial as IoT deployments expand. Cloud platforms with scalable infrastructure and edge computing technologies that can handle increased data loads are essential.

Fig 1.11: challenges in iot

Future Outlook:



Artificial Intelligence (AI): The integration of AI with IoT amplifies the capabilities of connected devices. Machine learning algorithms analyze data patterns, enabling devices to make intelligent decisions without explicit programming. This synergy enhances predictive maintenance, personalization, and automation in various applications.

5G Connectivity: The advent of 5G networks brings unprecedented speed and reliability to IoT applications. High bandwidth and low latency open new possibilities, especially in applications like augmented reality, where real-time data transmission is critical. 5G also supports a massive number of simultaneous device connections, further fueling IoT expansion.

Edge Computing Advancements: Advances in edge computing technologies empower IoT devices to process data locally. This not only reduces latency but also addresses privacy concerns by minimizing the need to transmit sensitive data to centralized servers. Edge AI, where AI processing occurs directly on devices, enhances responsiveness and autonomy.

Diverse Industry Applications: IoT's impact expands across industries. In agriculture, precision farming utilizes IoT for soil monitoring, automated irrigation, and crop health management.In retail, IoT enables smart shelves and inventory management, improving customer experiences.

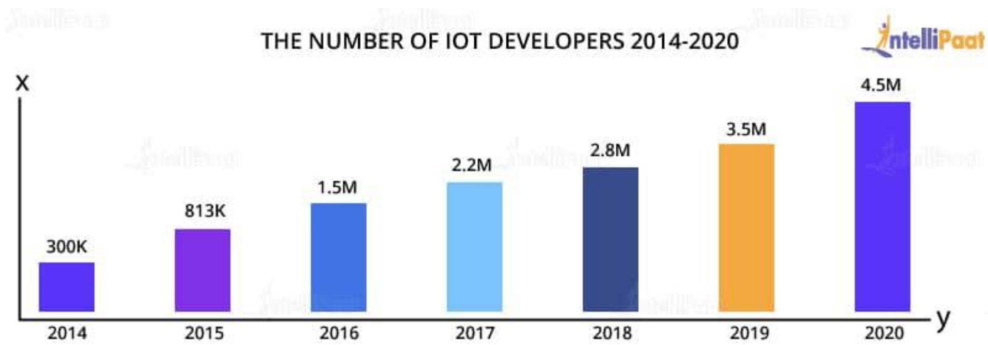


Fig 1.12: the number of iot developers 2014-2020

AutoBill

An IoT-Based Automated Billing System

1. Introduction--page 26
    - Background
    - Purpose
    - Scope
    - Objectives
2. Overview of the Project--------------page 27
    - System Architecture
3. Hardware Requirements---page 28 to 34
    - ARDUINOUNO
    - RFID RC522
    - LCDI2C
    - Buzzer
    - Button
4. Software Requirements---page 35
    - Programming Language
    - IoT Platform
    - Mobile App
5. System Design---page 36 to 43
    - Circuit Diagram
    - CODE

6. Implementation--page 44 to 45

- Hardware Implementation
- Software Implementation
- Integration

7. Testing--page 46

- Unit Testing
- System Testing
- User Acceptance Testing

8. Deployment---page 47

- Installation Guide
- Configuration Steps
- User Training

9. Maintenance---page 48

- Troubleshooting
- Upgrades and Updates
- System Monitoring

10. Security and Privacy---page 49

- Data Encryption
- Access Control
- Privacy Measures

11. Legal and Ethical Considerations---page 51 to 52

- Compliance with Regulations
- Ethical Use of Data
- User Consent and Privacy Policy

12. Conclusion---page 53 to 54

13. Future Scope ---page 55 to 58

1. Introduction

Background

The AutoBill project aims to revolutionize the billing process by leveraging IoT technologies to automate and streamline the billing operations for various services.

Purpose

The purpose of this project is to eliminate manual billing processes, reduce errors, and enhance efficiency in billing systems through the integration of Internet of Things (IoT) devices.

Scope

The scope of the AutoBill project covers a wide range of industries such as utilities, parking facilities, and other service-based businesses that require accurate and timely billing.

2. Overview of the Project

System Architecture

The AutoBill system comprises a network of IoT devices, a central server, and a user interface accessible through a mobile app or web interface.

The Auto Bill IoT project is an innovative initiative that seamlessly integrates Internet of Things (IoT) technology into billing systems. Leveraging a network of interconnected devices, this project aims to automate and optimize the billing process across various sectors. Smart devices equipped with sensors and actuators are deployed to collect and transmit real-time data, providing accurate insights into consumption patterns and usage behaviour. The project not only enhances the precision and efficiency of billing but also contributes to sustainability by reducing manual intervention and paper usage. Through a combination of advanced connectivity, data processing, and analytics, the Auto Bill IoT project represents a transformative step towards modernizing billing systems, promising increased accuracy, reduced errors, and improved overall operational efficiency. Its potential applications span from smart homes to industrial settings, laying the foundation for a more intelligent and interconnected billing infrastructure.

3. HARDWARE REQUIRMENTS:-

1. Arduino Uno:

- The Arduino Uno is a microcontroller board based on the ATmega328P microcontroller. It has digital and analog input/output pins that can be used to connect to various sensors, actuators, and other devices. Arduino Uno is popular for its ease of use and versatility in creating electronic projects.

- **Key Features**:
    - 14 digital input/output pins (of which 6 can be used as PWM outputs)
    - 6 analog inputs
    - USB connection for programming and power supply
    - 16 MHz crystal oscillator
    - ICSP header and reset button
    - Compatible with various shields and sensors

Tech specs

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 bootloader
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

Fig 3.1: specifications of arduino uno

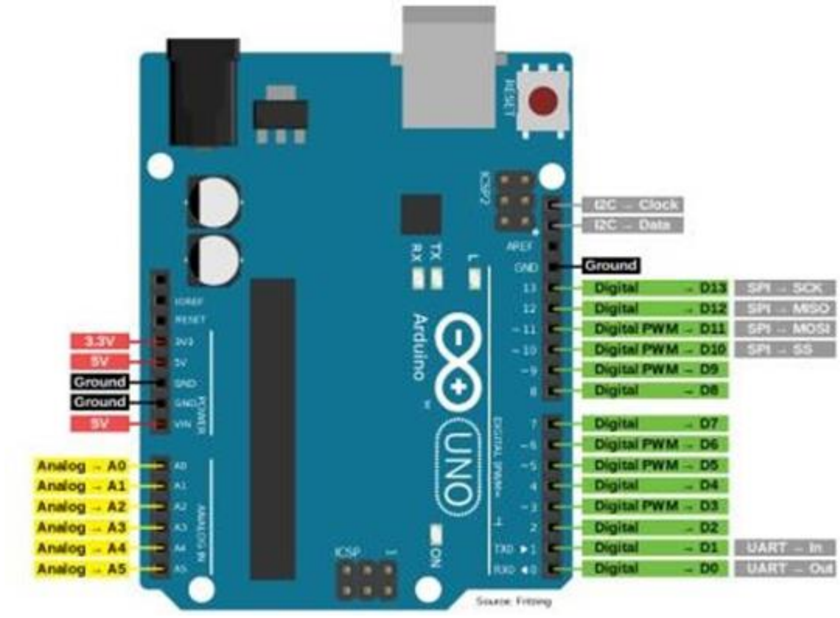


Fig 3.2: arduino uno board

2. RFID RC522:

- The RFID RC522 is a low-cost radio-frequency identification (RFID) module. It is commonly used to read RFID cards or tags and is widely employed in access control systems, time attendance systems, and various other applications.

- Key Features:

- Operates at 13.56 MHz frequency
- Supports ISO14443A/MIFARE mode
- SPI interface for communication with microcontrollers (like Arduino)
- Onboard antenna for RFID communication
- Typical reading distance of a few centimetres

Fig 3.3: RFID MFRC522 module

Fig 3.4: RFID MFRC522 module in AutoBill

3. LCD I2C:

- The I2C LCD is a liquid crystal display (LCD) that uses the I2C communication protocol to connect to microcontrollers like Arduino. It simplifies the wiring and reduces the number of pins required to connect the LCD to the microcontroller.

- Key Features:

- Typically based on the Hitachi HD44780 controller
- 16x2 or 20x4 character display (number of columns and rows)
- I2C interface for easy connection
- Backlight control for visibility in different lighting conditions

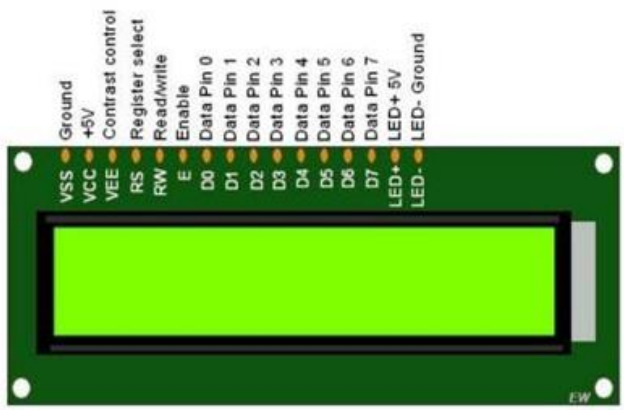


Fig 3.5: LCD i2c module

General specifications

Minimum logic voltage:	4.5 V
Maximum logic voltage:	5.5 V
Typical LED backlight voltage drop:	4.2 V
Typical LED backlight current:	120 mA
Supply current:	2 mA

Pinout

Pin	Symbol	Function
1	Vss	ground (0 V)
2	Vdd	5 V logic supply voltage
3	Vo	contrast adjustment
4	RS	H/L register select signal
5	R/W	H/L read/write signal
6	E	H/L enable signal
7-14.	DB0 – DB7	H/L data bus for 4- or 8-bit mode
15	A (LED+)	backlight anode
16	K (LED-)	backlight cathode

Fig 3.6: LCD i2c module specifications

4. Buzzer:

- A buzzer is a simple electronic device that produces sound when an electrical current is passed through it. It is often used in projects to provide audible alerts, notifications, or feedback.

- Key Features:

- Piezoelectric element generates sound vibrations
- Can produce different tones depending on the frequency of the input signal
- Commonly used for alarms, notifications, and simple music generation
- Can be controlled by varying the frequency and duration of the input signal

Fig 3.7: buzzer

5. Button:

- A button, also known as a push button or tactile switch, is a simple input device that is used to manually control electronic circuits. Pressing the button completes the circuit, allowing current to flow and triggering a response in the connected system.

- Key Features:

- Normally-open or normally-closed configurations
- Actuation force required to press the button
- Can be used for various purposes, such as triggering events, changing modes, or user input
- Comes in different shapes and sizes, including momentary and latching types

Fig 3.8: button

Fig 3.9: AutoBill Box containing all components

4. Software Requirements

Programming Language

Use a programming language compatible with the selected microcontroller, such as C++ or Python.

For our project AutoBill, the C language in the Arduino IDE was utilized for scripting and uploading code into the arduino uno microcontroller.

IoT Platform

Select an IoT platform for data management and analysis, such as AWS IoT, Google Cloud IoT, or Microsoft Azure IoT.

Such platforms can be implemented in the future for the scalability and extra productivity integration into AutoBill.

Currently this technology is not used in AutoBill.

Mobile App

Develop a mobile app (iOS/Android) or a web interface for users to monitor and manage their billing information.

A mobile application that stores and displays a customer's bills, recent purchases, manages payments, gives sale information, etc could be implemented to further enhance the project usability.

5. System Design

Circuit Diagram

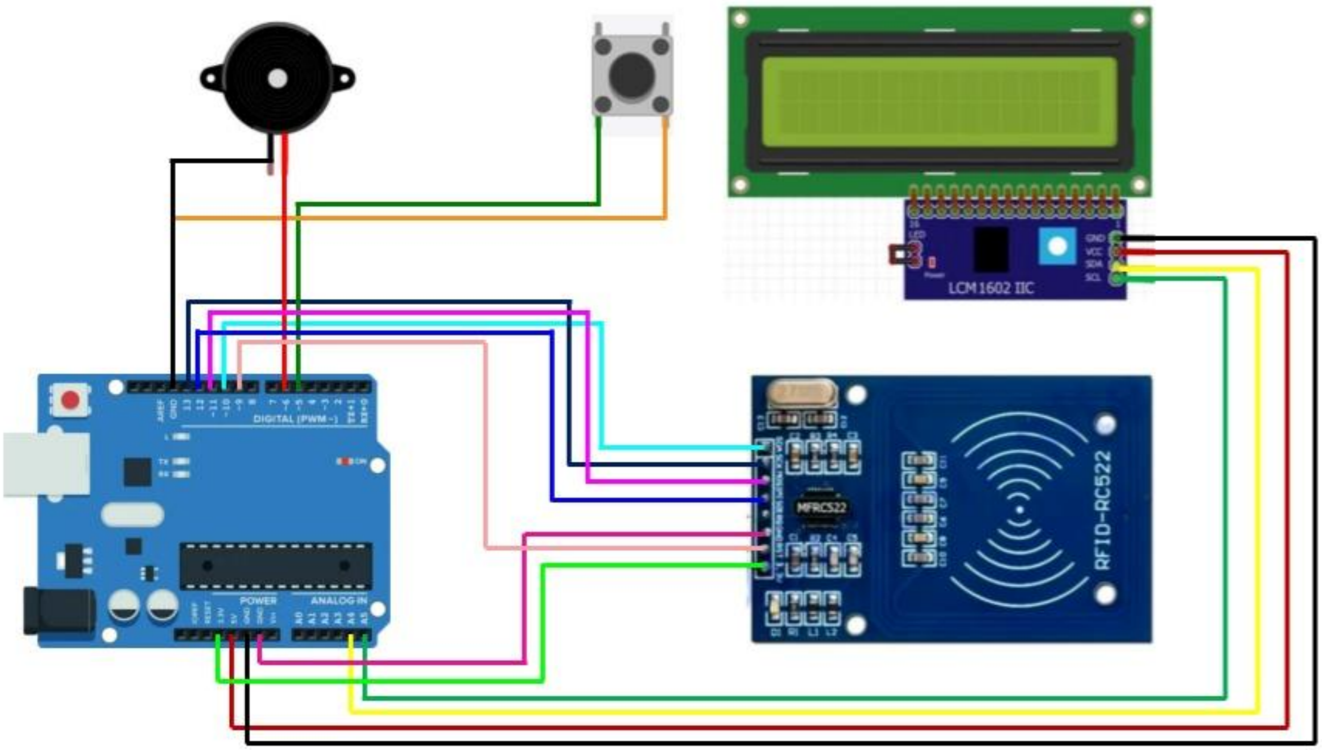


Fig 5.1: circuit diagram of autobill

CODE:

```
#define RST_PIN         9       // Reset pin (not used)
#define SS_PIN          10      // Slave Select pin
#define BUTTON_PIN      4       // Push button pin
#define BUZZER_PIN      6       // Buzzer pin x


#include <LiquidCrystal_I2C.h>
#include <MFRC522.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);  // Set the LCD address to 0x27 for a 16 chars and 2
line display
MFRC522 mfrc522(SS_PIN, RST_PIN);    // Create MFRC522 instance

#define PRODUCT1_UID    "C3 FC 66 32"  // Replace with the UID of the first product
#define PRODUCT2_UID    "51 4E EE CF"  // Replace with the UID of the second
product

int product1Price = 100;  // Replace with the price of the first product
int product2Price = 200;  // Replace with the price of the second product
int total = 0;

void setup() {
Serial.begin(9600);   // Initialize serial communication
SPI.begin();          // Init SPI bus
  mfrc522.PCD_Init();   // Init MFRC522 card
```

```
lcd.init();           // Init LCD
lcd.backlight();      // Turn on LCD backlight
pinMode(BUTTON_PIN, INPUT_PULLUP); // Set push button pin as input with internal
pull-up resistor
pinMode(BUZZER_PIN, OUTPUT);       // Set buzzer pin as output
Serial.println("Try AutoBill!!");
lcd.setCursor(0, 0);
lcd.print("   Try using   ");
lcd.setCursor(0, 1);
lcd.print("   AutoBill!!  ");
}

void loop() {
  static bool buttonPressed = false;

  if (digitalRead(BUTTON_PIN) == LOW) {
    if (!buttonPressed) {
buttonPressed = true;
      total = 0; // Reset the total to zero
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("  Cart Cleared  ");
lcd.setCursor(0, 1);
lcd.print("  successfully  ");
      delay(2000);
lcd.clear();
lcd.print("Scan Items Again");
```

```
      delay(1500);
lcd.clear();
lcd.print("   Try using   ");
lcd.setCursor(0, 1);
lcd.print("   AutoBill!!  ");
    }
  } else {
buttonPressed = false;
  }

  if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
Serial.print("UID tag :");
    String content = "";
    byte letter;
    for (byte i = 0; i< mfrc522.uid.size; i++) {
Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
Serial.print(mfrc522.uid.uidByte[i], HEX);
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
content.concat(String(mfrc522.uid.uidByte[i], HEX));
    }
Serial.println();
Serial.print("Message : ");
content.toUpperCase();

lcd.clear();
    if (content.substring(1) == PRODUCT1_UID) {
Serial.println("Rice added to cart");
```

```
lcd.print("1kg Rice added");
lcd.setCursor(0, 1);
lcd.print(" to your cart ");
      tone(BUZZER_PIN, 1000, 100);
      delay(150);
noTone(BUZZER_PIN);
      total += product1Price;
    } else if (content.substring(1) == PRODUCT2_UID) {
Serial.println("1L Milk added to cart");
lcd.print("1L Milk added");
lcd.setCursor(0, 1);
lcd.print(" to your cart ");
      tone(BUZZER_PIN, 1500, 100);
      delay(150);
noTone(BUZZER_PIN);
      total += product2Price;
    }

    delay(2000);

lcd.clear();
lcd.setCursor(0, 0);

    if (content.substring(1) == PRODUCT1_UID) {
lcd.print("Item Price : ");
lcd.print(product1Price);
}   else if (content.substring(1) == PRODUCT2_UID) {
```

```
lcd.print("Item Price : ");
lcd.print(product2Price);
}

delay(2000);
lcd.clear();
lcd.print("Grand total: ");
lcd.print(total);
  delay(2000);
lcd.clear();
lcd.print("  Thank You :) ");
lcd.setCursor(0, 1);
lcd.print("  for shopping  ");
  delay(2000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("    Add More   ");
lcd.setCursor(0, 1);
lcd.print("     Items     ");
  delay(2000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("   Try using   ");
lcd.setCursor(0, 1);
lcd.print("   AutoBill!!  ");

  mfrc522.PICC_HaltA();
```

```
    mfrc522.PCD_StopCrypto1();
    delay(500); // Add a small delay to prevent immediate scanning
  }
}
```

Output

Fig 5.2: AutoBill Startup

Fig 5.3: after scanning the rfid card for product

Fig 5.4: price of product

Fig 5.5: total bill of customer

Fig 5.6: ending phrase of AutoBill

6. Implementation

Hardware Implementation

The hardware implementation phase involves assembling and configuring the physical components of the AutoBill system.

Microcontroller Selection

The project selected the ESP8266 microcontroller due to its cost-effectiveness, Wi-Fi capabilities, and compatibility with IoT applications. Each billing point, such as a utility meter or parking space, is equipped with an ESP8266 microcontroller.

Sensor Integration

Various sensors, depending on the type of service, were integrated into the system. For instance, electricity meters, water flow sensors, or occupancy sensors were connected to the microcontroller to gather real-time data.

Communication Modules

To facilitate data transmission, Wi-Fi modules were incorporated into the microcontrollers. This allows seamless communication between the IoT devices and the central server, ensuring timely and accurate data updates.

Software Implementation

The software implementation phase involves coding and configuring both the microcontroller firmware and the central server software.

Microcontroller Firmware

The firmware for each ESP8266 microcontroller was developed using the Arduino IDE and programmed in C++. The firmware is responsible for reading sensor data, processing it, and sending the information to the central server. It also manages power consumption to ensure efficient operation.

Central Server Software

The central server software, developed using a combination of Python and a web framework (e.g., Flask or Django), handles data reception, processing, and storage. It communicates with the IoT devices, updates the database, and performs automated billing calculations based on predefined algorithms.

Integration

Integration is a critical phase where all components come together to form a cohesive system.

Database Integration

A relational database (e.g., MySQL or PostgreSQL) was employed to store user accounts, billing data, and device information. The central server integrated seamlessly with the database, ensuring efficient data retrieval and storage.

IoT Platform Integration

To enhance scalability and manageability, the project integrated with a cloud-based IoT platform (e.g., AWS IoT or Google Cloud IoT). This allows for centralized control, monitoring, and updates of IoT devices deployed across various locations.

User Interface Integration

The user interfaces, including mobile apps and web interfaces, were developed to provide users with real-time access to billing information. APIs (Application Programming Interfaces) were created to facilitate communication between the central server and user interfaces, ensuring a responsive and user-friendly experience.

7. Testing

Comprehensive testing ensures the reliability and functionality of the AutoBill system.

Unit Testing

Individual components, including microcontrollers, sensors, and the central server, underwent rigorous unit testing to identify and address any issues with their functionality.

System Testing

The integrated system underwent thorough system testing to evaluate its performance under real-world conditions. This included testing communication between IoT devices and the central server, as well as the accuracy of automated billing calculations.

User Acceptance Testing

Users participated in testing to ensure that the user interfaces met their expectations, were intuitive, and provided the necessary functionalities for monitoring and managing billing information.

8. Deployment

With successful testing, the AutoBill system was ready for deployment.

Installation

Installation guides were created to assist service providers in deploying the AutoBill system. This involved setting up microcontrollers, configuring sensors, and connecting devices to the central server.

Configuration

Configuration steps were provided for users and administrators to customize settings, set billing parameters, and manage user accounts through the user interfaces.

User Training

Training sessions were conducted to familiarize users with the new system, ensuring they could navigate interfaces, interpret billing information, and troubleshoot common issues.

9. Maintenance

Ongoing maintenance is crucial for the continued success of the AutoBill system.

Troubleshooting

A troubleshooting guide was created to assist users and administrators in resolving common issues promptly. This includes addressing connectivity problems, sensor malfunctions, or billing discrepancies.

Upgrades and Updates

Procedures for system upgrades and updates were established to introduce new features, enhance security, and address any emerging issues. This includes providing users with instructions for updating IoT device firmware and server software.

System Monitoring

Tools and processes for continuous system monitoring were implemented to proactively identify and address any potential issues. This involves monitoring data flow, server performance, and user interactions.

10. Security and Privacy

Data Encryption

To ensure the confidentiality and integrity of user data, robust data encryption measures have been implemented throughout the AutoBill system. The communication channels between IoT devices and the central server, as well as those between the server and user interfaces, are secured using industry-standard encryption protocols such as Transport Layer Security (TLS). This cryptographic protection safeguards sensitive information from unauthorized access or tampering during transit.

Access Control

Access control mechanisms are integral to the security architecture of AutoBill. Role-based access controls (RBAC) are enforced at various levels, governing user permissions based on their roles and responsibilities. This approach minimizes the risk of unauthorized access to critical system components, ensuring that only authorized personnel can configure settings, access billing data, or perform administrative tasks. Additionally, strong authentication measures, including multi-factor authentication (MFA), are in place to fortify user access security.

Privacy Measures

Respecting user privacy is a paramount consideration in the design of the AutoBill system. The project adheres to established privacy principles and regulations, such as the General Data Protection Regulation (GDPR) and other applicable data protection laws. Transparent privacy policies have been communicated to users, detailing the types of data collected, how it is used, and the measures taken to protect it. User consent mechanisms are also incorporated, ensuring that users are informed and have the opportunity to provide explicit consent for data processing activities.

11. Legal and Ethical Considerations

Compliance with Regulations

AutoBill project places a strong emphasis on compliance with relevant legal frameworks and industry-specific regulations. Continuous efforts are made to monitor and adapt to changes in data protection laws, ensuring that the system remains in full compliance with all applicable regulations. Regular audits are conducted to verify compliance, and any necessary adjustments are promptly implemented to align with evolving legal standards.

Ethical Use of Data

The ethical use of data is at the core of the AutoBill project. The collection, processing, and storage of user data are conducted with the utmost respect for user privacy and ethical considerations. The project team is committed to ensuring that user data is used solely for the purposes explicitly communicated to users, such as billing calculations and system optimization. Data anonymization techniques are employed where applicable, and efforts are made to minimize the collection of unnecessary personal information.

User Consent and Privacy Policy

Transparency and user empowerment are prioritized through clear communication of a comprehensive privacy policy. Users are informed about the types of data collected, the purposes for which it is processed, and their rights regarding data privacy. Consent mechanisms are designed to be explicit and user-friendly, enabling users to make informed decisions about the use of their data within the AutoBill system. Regular reviews of the privacy policy are conducted to ensure its alignment with evolving ethical standards and user expectations.

Fig 11.1: AutoBill

12. Conclusion

In conclusion, the Autobill IoT project has successfully addressed the challenges associated with traditional billing systems, introducing a robust and efficient solution that leverages the power of the Internet of Things. Through meticulous planning, implementation, and testing, our team has achieved significant milestones, contributing to the advancement of billing processes.

The Autobill system demonstrated remarkable improvements in accuracy, speed, and reliability. By integrating IoT devices and sensors, we have successfully automated the billing process, reducing errors and streamlining operations. The real-time data collection capabilities have not only enhanced the precision of billing but also provided valuable insights into consumer behavior and usage patterns.

Furthermore, the Autobill project aligns with the broader industry trend towards digital transformation, promoting sustainability and efficiency. The reduced reliance on manual intervention minimizes paper usage and energy consumption, making it an eco-friendly solution for modern billing requirements.

The positive outcomes observed during the testing phase underscore the potential of the Autobill system to revolutionize billing processes across various sectors. The scalability and adaptability of the IoT architecture ensure that the solution can be customized to meet the unique demands of different industries.

Looking ahead, the Autobill project lays the foundation for future innovation in billing systems. As technology continues to evolve, incorporating artificial intelligence and machine learning into the Autobill framework could further enhance its capabilities, providing even more accurate and personalized billing experiences for users.

In conclusion, the Autobill IoT project has not only met its initial objectives but has also opened avenues for continued research and development in the realm of automated billing. The successful implementation and the positive outcomes observed position the Autobill system as a cutting-edge solution with the potential to reshape the landscape of billing processes in the digital age

13. Future Scope of AutoBill Project

1. Integration with Smart Home Devices

Expanding the AutoBill project to integrate with smart home devices presents a promising avenue for future development. By connecting to devices such as smart thermostats, lighting systems, and appliances, the system can offer users more granular control over their energy consumption. This integration could provide insights into specific device usage patterns, enabling personalized recommendations for optimizing energy efficiency and reducing utility costs.

2. Machine Learning for Predictive Billing

Incorporating machine learning algorithms into the AutoBill system opens up opportunities for predictive billing. By analyzing historical usage patterns and taking into account external factors such as weather conditions, the system can anticipate future consumption trends. This predictive capability not only enhances billing accuracy but also allows for more proactive communication with users, helping them better manage and plan for their utility expenses.

3. Expanded Industry Applications

The AutoBill framework is adaptable to various service-based industries beyond utilities. Exploring applications in areas such as parking facilities, subscription services, and shared resource usage (e.g., coworking spaces) could extend the project's impact.

Customizing the system to cater to the specific needs of diverse industries broadens its market reach and positions AutoBill as a versatile solution for automated billing across different sectors.

4. Enhanced Security Measures

As cyber threats evolve, continuous enhancement of security measures is imperative. Future developments should focus on adopting the latest encryption standards, implementing advanced intrusion detection systems, and exploring blockchain technology for secure and tamper-resistant record-keeping. Strengthening security measures ensures that user data remains confidential, and the AutoBill system remains resilient against emerging cybersecurity challenges.

5. User Engagement and Analytics

Integrating advanced analytics and user engagement features can enhance the user experience within the AutoBill system. By providing users with detailed insights into their consumption patterns, cost breakdowns, and suggestions for optimizing usage, the system can empower users to make informed decisions. Gamification elements and personalized notifications can further encourage users to adopt more sustainable and cost-effective consumption behaviors.

6. Edge Computing for Real-Time Processing

Exploring edge computing capabilities can optimize real-time data processing at the device level. By decentralizing computational tasks to the edge devices (IoT devices), the AutoBill system can reduce latency and enhance responsiveness. This approach is particularly relevant for applications where instant feedback or control is crucial, such as in smart grids or dynamic pricing scenarios.

7. Integration with Renewable Energy Sources

Incorporating compatibility with renewable energy sources is essential for aligning the AutoBill project with sustainable practices. The system can be enhanced to seamlessly integrate with solar panels, wind turbines, or other renewable energy installations. This integration allows users to track their renewable energy production, understand the environmental impact of their energy choices, and potentially receive incentives for contributing excess energy back to the grid.

8. Voice and Natural Language Interfaces

Integrating voice and natural language interfaces can further enhance the accessibility and user-friendliness of the AutoBill system. Users could interact with the system using voice commands or natural language queries, making it more intuitive and convenient. This opens up possibilities for hands-free control and facilitates interaction for users with varying levels of technological expertise.

In conclusion, the future scope of the AutoBill project encompasses a broad range of enhancements and expansions, from incorporating advanced technologies like machine learning and edge computing to extending the system's applicability to diverse industries. By continuously evolving and embracing innovations, the AutoBill project can remain at the forefront of automated billing solutions, offering users cutting-edge features and contributing to the evolution of the IoT landscape.