

Unified Smart Three in One Meter For Water, Gas and Energy

Mr. Nagesh H B
Assistant Professor
Department of Electronics and
Communications ACS College
of Engineering,
Bangalore, India
nagesh.murthy@gmail.com

Ankitha c
UG Scholar
Department of Electronics and
Communication
ACS College of Engineering,
Bangalore, India
ankithac88@gmail.com

Rekha ck
UG Scholar
Department of Electronics and
Communication
ACS College of Engineering,
Bangalore, India
rekhack22@gmail.com

Yashodha MS
UG Scholar
Department of Electronics and Communication
ACS College of Engineering, Bangalore, India
yashuashu2004@gmail.com

Abstract—The proposed solution uses IOT technology to enable real-time monitoring and automated data collection. Current and voltage sensors, water flow sensors, and gas sensors are integrated with a micro-controller such as ESP32or Arduino to measure respective utilities. Data is transmitted to the cloud using Wi-Fi or GSM communication. The system is programmed using Embedded C/C++ are used for data storage and visualization. A web or mobile application displays consumption details, billing information, and alerts. This system improves billing accuracy, reduces manual effort, and supports efficient resource management. Software side, Embedded C/C++ is used for programming the micro-controller. A web-based or mobile application, developed using HTML, CSS, JavaScript, and Python, displays real-time consumption data, historical usage patterns, billing information, and alert notifications for leakage or abnormal consumption.

I. INTRODUCTION

A unified smart 3-in-1 meter is an innovative utility measurement device that integrates the monitoring and reporting of gas, water, and electrical energy

consumption into a single smart unit, revolutionizing how residential, commercial, and industrial utilities are tracked and managed. Traditionally, separate meters have been installed for each utility, leading to higher installation costs, increased maintenance efforts, and fragmented data collection. In contrast, a unified smart meter consolidates all three into a compact digital platform equipped with advanced sensors and communication technologies. This integration not only simplifies infrastructure but also enhances accuracy by replacing mechanical components with digital measurement systems that reduce errors, improve reliability, and extend device longevity

At the heart of the unified smart 3-in-1 meter is its ability to collect and transmit real-time consumption data for gas, water, and electricity. Using communication protocols such as NB-IoT, LoRaWAN, or cellular connectivity, the meter can automatically send usage information to utility providers or centralized management systems without requiring manual reading. This enables remote monitoring and automated billing, which significantly reduces operational costs for providers and minimizes billing

centralized management systems without requiring manual reading. This enables remote monitoring and automated billing, which significantly reduces operational costs for providers and minimizes billing .

II. LITERATURE SURVEY

The rapid growth of Internet of Things (IoT) and smart infrastructure has accelerated research in unified metering technologies that combine multiple utility measurements within a single integrated system. Early work in multivariate resource monitoring focused on independent smart meters for electricity, gas, and water. As individual systems matured, researchers recognized the need for integrated metering to improve efficiency, data coherence, and cost effectiveness.

In traditional metering research, the focus was on overcoming limitations of mechanical meters — such as inaccuracy, high maintenance, and manual data collection — by digitalizing individual meters. Studies by Güngör et al. (2011) demonstrated that smart electricity meters using digital communication protocols significantly improved data quality and enabled remote monitoring. Similarly, work on smart gas meters by Suryadevara and Mukhopadhyay (2012) highlighted reliable ultrasonic sensing techniques and low-power communication for remote gas usage reporting. Research on smart water metering by Nguyen et al. (2015) explored acoustic and electromagnetic flow sensor integration to detect consumption patterns and leakage with high accuracy. To improve accuracy and speed of accident detection using advanced image processing and deep learning models.: Intelligent roadside cameras, YOLO-CA neural network, Multiscale Feature Fusion (MSFF), Dynamic loss- weight adjustment. The model achieved a high detection speed of 21.6 FPS and an average precision of 90%, showing promising results for smart-city integration. Requires advanced infrastructure, high-cost cameras, and IoT gateways — not suitable for low-cost vehicle-mounted systems like ESP32-based solutions.

III. PROPOSED SYSTEM

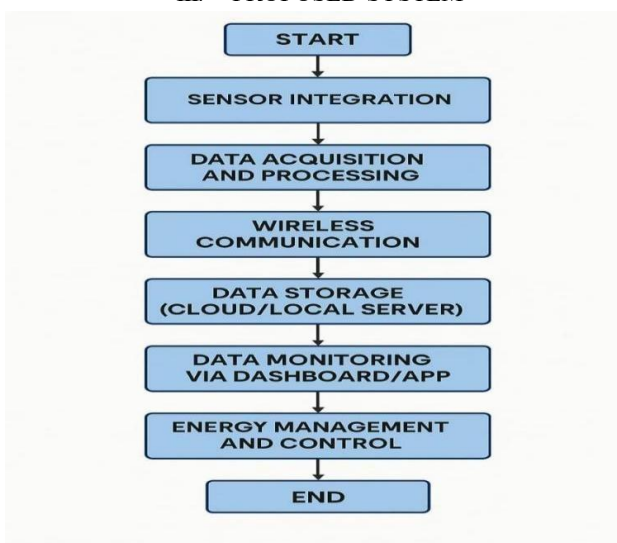


Fig 3.1: Proposed System

This block diagram represents an Unified Smart 3-in-1 Meter for Gas, Water, and Energy. The proposed system introduces a Unified Smart 3-in-1 Meter designed to measure, monitor, and manage gas, water, and electrical energy consumption through a single intelligent device. The main objective of this system is to replace multiple conventional meters with a compact, digital, and IoT-enabled solution that ensures accurate measurement, real-time monitoring, and efficient utility management. By integrating three utilities into one platform, the system reduces installation complexity, maintenance cost, and manual intervention while improving data reliability and accessibility.

IV. HARDWARE REQUIREMENTS

ESP32: The ESP32 is a powerful, low-cost microcontroller developed by Espressif Systems, widely used in IoT (Internet of Things) applications because of its high processing speed, built-in communication features, and rich set of peripherals. It is an upgraded version of the ESP8266 and includes a dual-core Tensilica processor, which allows it to handle multiple tasks simultaneously, making it suitable for real-time monitoring systems like accident detection and health monitoring. The ESP32 supports both 3.3V logic and offers



Fig 4.1: ESP32

excellent power-saving modes, making it suitable for battery-operated systems. It also has built-in timers, watchdogs, touch sensors, hall sensors, and a hardware encryption engine, which ensures secure communication. The device can run Arduino code, Micro Python, or the native ESP-IDF framework, giving developers flexibility in programming. This fig shown in figure 4.1.

LCD Display: A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and 7-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements. The figure 4.2 shows the LCD Display.



Fig 4.2: LCD Display

Push Button: A Pushbutton Switch is a switch designed so that its contacts are opened and closed by depressing and releasing a pushbutton on the Switch in the direction of its axis. Used to arise the emergency condition.this fig 4.3 shows push button.



Fig 4.3: Push Button

Gas Sensor: An alcohol sensor detects the attentiveness of alcohol gas in the air and an analog voltage is an output reading. The sensor can activate at temperatures ranging from -10 to 50° C with a power supply is less than 150 Ma to 5V. The sensing range is from 0.04 mg/L to 4 mg/L, which is suitable for breathalyzers. The passive alcohol sensor (PAS) is a device developed to assist police in identifying drinking drivers. The PAS draws in mixed expired and environmental air from in front of the subject's face and passes it into a fuel cell sensor that can detect very small amounts of gas. The fig 4.3 shows Gas Sensor detection.



Fig 4.4: Gas Sensor

Node MCU : it also supports OTA (Over-The-Air) updates and works efficiently with cloud-based energy management systems. Its compact size, built-in connectivity, and ease of programming make it a highly suitable choice for developing affordable and scalable smart meter solutions.The fig 4.5 shows Node MCU.



Fig 4.5 : Node MCU

LDR Sensor: An LDR (Light Dependent Resistor), also known as a photo resistor, is a sensor that detects light intensity. In the context of a unified smart energy meter, the LDR is commonly used to measure electricity usage by detecting the blinking of an LED on a digital energy meter. Most modern digital electricity meters have a small Blinking LED that flashes once for every unit of power consumed.this fig 4.6 shows LDR Sensor.



Fig 4.6 : LDR Sensor

3-in-1 smart meter : It is a unified device designed to monitor electricity, water, and gas consumption simultaneously using a single embedded system. It integrates three different types of sensors-one for each utility-into a central microcontroller platform like Arduino Uno, ESP32, or Node MCU.



Fig 4.7 : 3-in-1 smart meter .

This setup allows for real-time, centralized data collection, processing, and transmission for all utilities from one compact unit. Each utility is monitored by its corresponding sensor: Electricity is typically measured using a current sensor (e.g., ACS712) or an LDR that detects LED pulses from a digital energy meter. The fig 4.7 shows 3-in-1 smart meter .

Relay : A relay is an electrically operated switch used to control the connection or disconnection of electricity, water, or gas supply based on specific conditions. In a unified smart metering system, a relay allows the embedded controller (like Arduino or Node MCU) to automatically turn ON or OFF the utility supply when necessary-for example, during overuse, non-payment, or in prepaid metering scenarios.Fig 4.8 shows relay.



Fig 4.8 :Relay .

V. SOFTWARE REQUIREMENTS

Arduino IDE:



Fig 5.1: Arduino IDE

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio. The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consist of only two functions: setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. loop(): After setup() has been called, function loop() is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

Arduino IDE software is an open-source development environment used to write, compile, and upload programs to Arduino boards. It provides a simple and user-friendly interface, making it suitable for beginners as well as advanced developers. The Arduino IDE supports programming in a simplified version of C and C++, commonly referred to as Arduino programming. It allows users to create sketches (programs) that control hardware components such as LEDs, motors, sensors, displays, and communication modules.

The Arduino IDE is mainly used to program boards like Arduino Uno, Arduino Nano, and Arduino Mega 2560. These boards are based on microcontrollers such as the ATmega328P. The IDE provides built-in libraries and example codes that make hardware interfacing easy. Users only need to select the correct board and port, write the code,

and click the upload button to transfer the program to the board through a USB cable.

The interface of the Arduino IDE includes a code editor, a message area, a text console, and toolbar buttons for verifying and uploading programs. It also supports serial communication through the Serial Monitor, which helps in debugging and viewing real-time data from the microcontroller. Additionally, the IDE allows installation of extra libraries and board packages to support advanced modules and third-party hardware.

One of the main advantages of Arduino IDE is its cross-platform compatibility, as it runs on Windows, macOS, and Linux. It simplifies embedded system development by handling complex configurations in the background. Overall, Arduino IDE software plays a major role in electronics prototyping, IoT development, robotics, and educational projects by providing an easy and efficient programming platform.

Embedded C:

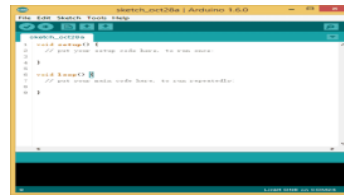


Fig 5.2: Overview of Arduino IDE

Embedded C software is a specialized form of the C programming language used to develop applications for embedded systems. An embedded system is a combination of hardware and software designed to perform a specific function within a device. Embedded C enables programmers to write efficient and reliable code that directly interacts with hardware components such as microcontrollers, sensors, actuators, timers, and communication modules. Unlike general-purpose C programs that run on computers with operating systems, Embedded C programs run on dedicated hardware with limited memory, processing power, and storage.

Embedded C is commonly used with microcontrollers such as ATmega328P, STM32F103C8T6, and ESP8266. These microcontrollers are widely used in applications like home automation, robotics, automotive control systems, medical devices, consumer electronics, and Internet of Things (IoT) projects. The software written in Embedded C directly accesses hardware registers to control input/output ports, manage communication protocols like UART, SPI, and I2C, and handle real-time operations using interrupts.

One important feature of Embedded C is efficient memory management, as embedded systems typically have limited RAM and ROM. Programmers often use keywords such as volatile to ensure proper hardware communication and implement interrupt service routines (ISRs) for fast response to external events. The program structure generally includes hardware initialization followed by an infinite loop that continuously executes system tasks.

The development process involves writing code in an integrated development environment (IDE), compiling it to generate a HEX file, and flashing it into the microcontroller. Embedded C software plays a vital role in modern electronics by enabling smart, automated, and reliable device control.

V1.RESULT AND DISCUSSIONS

The development and testing of the unified smart meter system for water, gas, and energy yielded promising results, confirming its potential as a reliable, intelligent, and practical solution for modern utility management. The results were observed in several phases, including component-level testing, system integration, real-time monitoring, wireless communication, cloud interfacing, and automated control functionalities. Below is a detailed breakdown of the key findings across each functional domain of the system.

Snap shot of working Smart meter shows the integrated system effectively collected real-time data from three different sensors: ACS712 current sensor for electricity, YF-S201 water flow sensor for water measurement, and MQ-6 gas sensor for gas flow and leakage detection. Each sensor provided stable and accurate outputs under test conditions. The current sensor detected variations in energy consumption with sensitivity, and the readings could be converted into kilowatt-hours using simple mathematical logic within the microcontroller. The water flow sensor consistently measured flow rates and total water volume with minimal deviation, based on the pulse frequency generated. The gas sensor also accurately detected the presence of LPG, triggering safety alerts when concentration thresholds were exceeded.

Real-Time Data Processing and Display explains about the Arduino Uno and Node MCU microcontrollers successfully read sensor values and converted them in to user-friendly units such as kilowatt-hours(kWh), liters, and cubic meters. The processed data was displayed in real-time on an LCD screen, providing instant feedback to the user. The local display was effective even during offline operation, ensuring usability regardless of internet connectivity. The LCD successfully cycled through each utility reading with proper formatting and clear labeling. Relay modules connected to the micro controller were tested for automated control based on preset conditions. The system was programmed to cut off the utility supply when usage exceeded specified thresholds or when prepaid credit was exhausted.

Hardware Prototype:

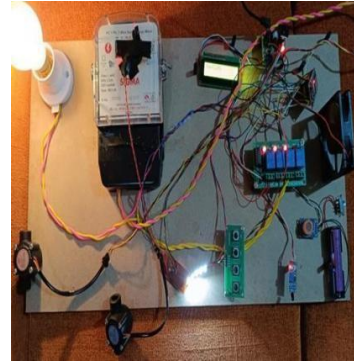


Fig 6.1:Hardware Prototype Of The Project

The figure 6.1 shows the Hardware prototype of the project without power Supply and the connection of Hardware components and the required program code is dumped into the ESP32.

TELEGRAM MESSAGE ALERT:

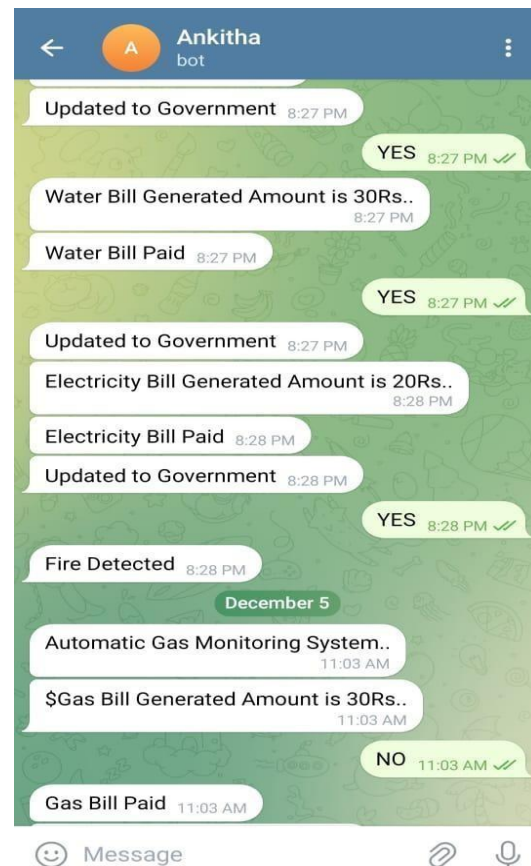


Fig 6.2: Telegram Message Alert

ARDUINO APPLICATION

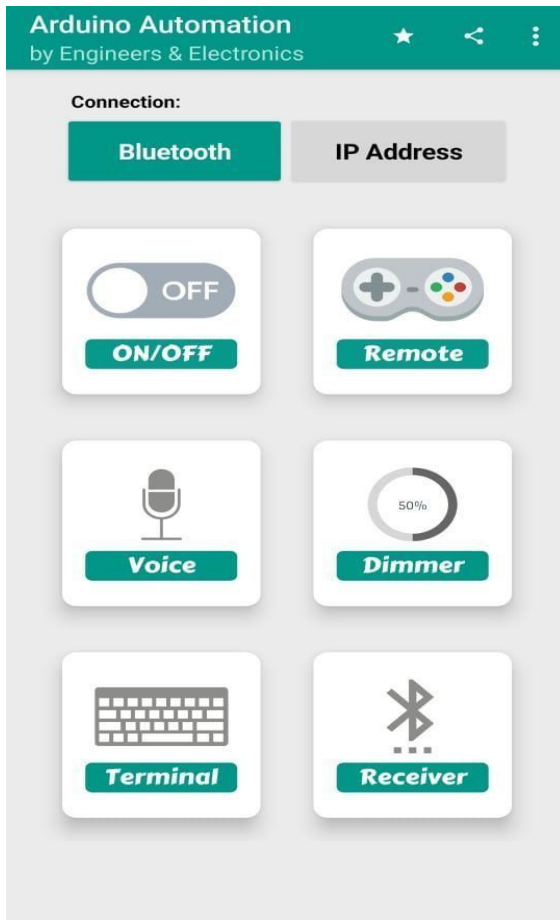


Fig 6.3:Arduino Application

LIMITATIONS:

- High Initial Cost
- Cybersecurity Risks
- Network Dependency
- Maintenance Cost
- Technical Complexity
- Power Dependency
- Privacy Concerns

VI. CONCLUSION

The Unified Smart Meter for Water, Gas, and Energy presents a practical and intelligent solution for modern utility management by integrating three essential services into a single system. Through the use of embedded systems, sensor technologies, wireless communication, and IoT platforms, the project successfully enables real-time monitoring, automated billing, remote control, and efficient resource management. It empowers both users and utility providers with greater transparency, control, and data-driven decision-making, supporting smarter and more sustainable utility usage.

APPLICATIONS:

- Real-time monitoring of water, gas, and electricity usage.
- Remote meter reading (no manual checking)
- Smart home and smart city integration
- Cloud monitoring using platforms like Amazon Web Services
- Renewable energy management (e.g., battery systems like Tesla Powerwall)
- Demand forecasting and data analytics
- Smart billing and automated reports
- Leakage and theft detection
- Energy and cost optimization

ADVANTAGES:

- Real-Time Monitoring
- Accurate Billing
- Remote Access
- Leakage Detection
- Energy Efficiency
- Reduced Operational Cost
- Theft Detection

VII. REFERENCES

- [1] Park, B S, Hyun, D.H, Cho, S.K. "Implementation of AMR system using power line communication". Conference and Exhibition 2002: Asia Pacific. IEEE/PES, Vol.1, Oct.2002, pp.18-21
- [2] Raja, G.T. "Electricity consumption and automatic billing through power line", Power Engineering Conference, 2007, PP. 1411 – 1415, 2007.
- [3] Bharath, P "Wireless automated digital energy meter", Sustainable energy technologies, International conference, 2008, PP. 564 – 567,2008.
- [4] Tasfin Mohaimeen Haq "Application of Power Line Carrier (PLC) in Automated Meter Reading (AMR) and Evaluating Non-Technical Loss (NTL)" International Journal of Research and Technology, Vol2, Issue8, August 2013.
- [5] Poonam Borle, Ankita Saswadkar, Deepali Hiwarkar, Rupali S.K ad "Automatic Metre Reading For Electricity Using Powerline Communication" International Journal of Advanced Research in Electrical and Instrumentation Engineering Vol2., Issue 3, March2013.
- [6] Mr. Vishal Kashinath Ovhal, Dr. U L. Bombale" Power Line Communication Based Automatic Electrical Billing Meter and Power Supply Control Using Frequency Shift Keying Modulation" International Journal of Modern Trends in Engineering and Research vol.2, Issue 4, April2015
- [7] G. Sujatha, D. Murali, R, Ram Kishore "Automated E B Billing and Supply Control using Power Line Communication" vol.45- NO 7, May 2015