International Journal of Science and Technology (IJST) | ISSN(O):2049-7318 | 2(3):76-78, 2026 | DOI: 10.5281/zenodo.21422827

IoT-Enabled Machine-to-Machine Communication for Smart Cities

M. Karunakar Reddy, S. Muni Mounika, G. Afreen Taj, G. Shaguftha, B. Soujanya

Department of Electronics & Communication Engineering Aditya
College of Engineering, Madanapalle, India

Abstract**—**This project presents an IoT-enabled machine-to-machine communication prototype designed for smart city energy monitoring and control applications. The system integrates multiple sensors and actuators to measure electrical units, temperature, current, and frequency while enabling automated control of electrical loads. An LDR sensor is used for unit measurement, while a PZEM-004T energy meter monitors current and frequency parameters. The Arduino UNO acts as the main controller, processing sensor data and managing devices such as relays, buzzer, LCD display, and a cooling fan. A NodeMCU module uploads the monitored data to the ThingSpeak cloud server for visualization and analysis. Alert mechanisms and load control enhance system reliability and safety. The prototype demonstrates efficient monitoring, data sharing, and basic automation suitable for smart city environments.

Index Terms**—**Machine-to-Machine (M2M), PZEM-004T, ThingSpeak, Energy monitoring, DHT11 Sensor, IoT, Arduino UNO, NodeMCU.

I. INTRODUCTION

The rapid growth of urban populations across the world has created an urgent need for smarter and more efficient management of city resources. Modern cities face numerous challenges, including rising energy demand, traffic congestion, environmental pollution, inefficient infrastructure management, and rising operational costs. To address these challenges, the concept of smart cities has emerged as a transformative approach that leverages advanced digital technologies to improve the quality of urban life while ensuring sustainable development.

Among the key enabling technologies behind smart city development is the Internet of Things (IoT), which allows physical devices to communicate, collect data, and operate intelligently through interconnected networks. A fundamental concept that enhances IoT efficiency is Machine-to-Machine (M2M) communication — the direct exchange of data between devices without requiring continuous human intervention.

In this project, an IoT-enabled M2M communication system is developed for smart city applications using commonly available sensors, microcontrollers, and communication modules. The system aims to demonstrate how interconnected devices can autonomously exchange data, monitor environmental and electrical parameters, and respond intelligently without continuous human intervention.

A. Internet of Things (IoT) Overview

The Internet of Things (IoT) refers to a network of physical objects connected to the internet that can collect, exchange, and analyze data automatically. These objects incorporate sensors, software, and communication hardware to interact with other devices and systems without requiring constant human intervention. The field has evolved due to the convergence of multiple technologies including ubiquitous computing, sensors, embedded systems, and machine learning. Traditional fields of embedded systems, wireless sensor networks, control systems, and automation independently and collectively enable the Internet of Things.

B. IoT Architecture

The IoT architecture defines how sensors, networks, and cloud systems interact to collect, process, and exchange data efficiently. The four-layer architecture comprises: (1) Sensing Layer — collects raw environmental data via sensors and actuators; (2) Network Layer — transmits data using Wi-Fi, Bluetooth, Zigbee, LoRaWAN, or cellular networks; (3) Data Processing Layer — analyzes and interprets sensor data for pattern detection, anomaly identification, and storage; and (4) Application Layer — interacts with end users and business systems via dashboards, mobile applications, and visualization tools.

C. Machine-to-Machine Communication

Machine-to-Machine (M2M) communication refers to automated, direct exchange of data between devices, machines, or sensors without human intervention. It is a foundational technology for IoT, often focusing on point-to-point communication to monitor, control, or manage assets remotely. At its core, M2M integrates sensors, communication modules, processing units, and cloud-based platforms. Core M2M principles include autonomous operation, periodic or event-driven data transmission, wired or wireless networking, and a data endpoint-communication network-integration point architecture. M2M systems employ specialized efficient protocols including MQTT (Message Queuing Telemetry Transport), CoAP, cellular networks (4G/5G/LTE-M/NB-IoT), LPWAN, and Wi-Fi [3].

TABLE I. DIFFERENCES BETWEEN IOT AND M2M COMMUNICATION

Feature	IoT	M2M
Definition	Network of connected devices communicating via the internet	Direct communication between machines
Communication Type	Device-to-device, device-to-cloud, device-to-user	Machine-to-machine only
Network Usage	Primarily internet-based	Cellular, wired, or private networks
Architecture	Open architecture with cloud integration	Often closed or proprietary
Scalability	Highly scalable (millions of devices)	Limited scalability
Data Processing	Cloud computing and big data analytics	Usually processed locally
Human Interaction	Via mobile apps or dashboards	Mostly automatic
Protocol Usage	HTTP, MQTT, CoAP	Telecom or proprietary protocols

II. SYSTEM ARCHITECTURE

The proposed system integrates sensors (LDR, DHT11, PZEM-004T), microcontrollers (Arduino UNO), a Wi-Fi module (NodeMCU ESP8266), relay modules, LCD display, buzzer, and a CPU cooling fan. The Arduino UNO acts as the central processing unit, gathering sensor data and executing control logic. The NodeMCU module provides internet connectivity for cloud data transmission to the ThingSpeak platform.

The system block diagram (Fig. 1) illustrates the architecture. Sensors feed real-time data to the Arduino UNO, which processes inputs and drives output actuators. Processed data is relayed to the NodeMCU over serial communication, which uploads it to the ThingSpeak IoT cloud for remote monitoring.

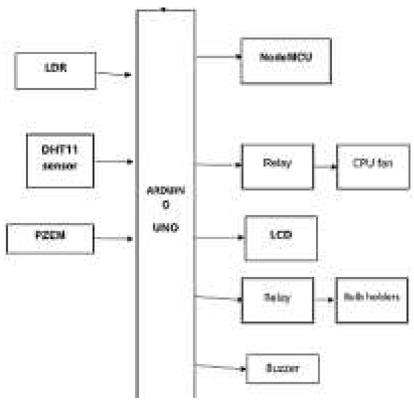

Fig. 1. System Block Diagram

III. HARDWARE COMPONENTS

A. Arduino UNO

The Arduino UNO is the central microcontroller based on the ATmega328P (8-bit AVR). It features 14 digital I/O pins, 6 analog input pins, 32 KB flash memory, and operates at 16 MHz. It is programmed using the Arduino IDE with C/C++ and interfaces with all sensors and actuators in the system.

B. NodeMCU (ESP8266)

The NodeMCU module provides built-in Wi-Fi connectivity via the ESP8266 SoC. It receives processed data from the Arduino via serial communication and uploads it to the ThingSpeak cloud platform for real-time visualization and remote monitoring.

C. DHT11 Temperature & Humidity Sensor

The DHT11 is a low-cost digital sensor measuring temperature (0–50°C, ±2°C accuracy) and humidity (20–80%, ±5% accuracy) at 1 Hz sampling rate. It interfaces with the Arduino via a single digital pin and triggers the cooling fan automatically when the temperature threshold is exceeded.

D. PZEM-004T Energy Meter Module

The PZEM-004T is a multi-function AC power monitor capable of measuring voltage (80–260 VAC), current (0–100 A), active power, energy (kWh), and frequency (45–65 Hz) via TTL serial communication. It provides accurate electrical parameter data for energy monitoring.

E. LDR Sensor, Relay, Buzzer & LCD

The LDR (Light Dependent Resistor) measures ambient light intensity for automated lighting control. A 5V relay module switches AC loads (bulbs and fan) based on sensor thresholds. The buzzer provides audible alerts on abnormal conditions. The 16x2 LCD display shows real-time sensor values locally.

III(B) SOFTWARE COMPONENTS

A. Arduino IDE

The Arduino Integrated Development Environment (IDE) is an open-source software platform used to write, compile, and upload programs to microcontroller boards. It supports C and C++ programming languages and comprises an editor for code writing and a compiler for uploading to Arduino modules. Key features include syntax highlighting, error detection, a serial monitor for real-time debugging, a board manager, and a library manager for sensor and module integration. The IDE is cross-platform, supporting Windows, Linux, and macOS.

In this project, the Arduino IDE is used for writing and editing microcontroller firmware, compiling code and detecting errors, uploading firmware to IoT devices, monitoring real-time data through serial communication, and testing sensor functionality and device interactions.

B. Embedded C

Embedded C is the programming language used to develop firmware for the microcontroller. It provides direct access to hardware resources while maintaining high execution speed. Key features include hardware-level programming for direct sensor and GPIO interaction, efficient memory management suitable for resource-constrained devices, real-time control capability, support for peripheral interfaces (UART, SPI, I2C), deterministic execution timing, compact code size, and wide industry adoption in automotive, healthcare, and smart city applications. In this system, Embedded C implements sensor data collection and processing, communication protocols between devices, automated device control logic, and cloud data transmission routines.

IV. IMPLEMENTATION

The system firmware is developed using Embedded C on the Arduino IDE. Sensor data from the DHT11, LDR, and PZEM-004T are continuously read by the Arduino UNO. The Arduino processes these values and executes automated control: activating the CPU fan via relay when temperature exceeds 30°C, controlling the lighting relay based on LDR readings, triggering the buzzer for overload or high-temperature conditions, and displaying all parameters on the LCD.

Processed data is serialized into a formatted string and sent to the NodeMCU module at 9600 baud via software serial. The NodeMCU parses this data and transmits it to the ThingSpeak platform over HTTP, enabling cloud-based real-time visualization of temperature, current, voltage, and energy consumption.

The power supply module steps down 230V AC mains to 5V DC using a transformer, bridge rectifier, filter capacitors, and a 7805 voltage regulator, providing stable power to all components.

V. RESULTS

The prototype was successfully implemented and tested. The system demonstrated reliable real-time data collection, automated device control, and seamless cloud connectivity. All 12 test cases — including system initialization, temperature-based fan control, LDR lighting automation, PZEM energy monitoring, IoT data transmission, buzzer alerts, LCD display, serial communication, power stability, scalability, network failure handling, and response time — passed with expected results.

M2M communication was verified through autonomous device interactions — temperature exceeding 30°C triggered the cooling fan automatically, and lighting responded to LDR thresholds without manual intervention. ThingSpeak cloud dashboards confirmed real-time data updates for temperature, humidity, voltage, and current with a transmission delay of approximately 2–3 seconds. System response time was consistently below 2 seconds, satisfying smart city real-time requirements.

CONCLUSION

This paper presented an IoT-enabled Machine-to-Machine communication prototype for smart city energy monitoring and control. The system successfully integrates Arduino UNO, NodeMCU, and multiple sensors to enable autonomous device interaction, real-time monitoring, and cloud-based data visualization. Automated load control and alert mechanisms demonstrated improved energy efficiency and safety. The modular and scalable design confirms the system's suitability for smart city deployments. Future work will incorporate AI-based analytics, advanced cybersecurity, multi-node networks, and broader sensor integration for comprehensive urban monitoring.

REFERENCES

1. N. O. Nwazor and V. Ugah, "Device-To-Device (D2D) Data Communications in 5G Networks," International Journal of Advances in Engineering and Pure Sciences, vol. 4, pp. 1151–1154, Jan. 2022.
2. Kadiyan, M. Singh, A. Kumar, and A. Saini, "Smart Home Automation System Based on IoT," IJSREM, vol. 7, pp. 1–8, May 2023.
3. K. L. Raju et al., "Home Automation and Security System with Node MCU using IoT," IEEE ViTECoN, Mar. 2019.
4. Vikram N. et al., "A Low-Cost Home Automation System Using Wi-Fi Based Wireless Sensor Network Incorporating IoT," IEEE IACC,pp. 174–178, 2017.
5. Kurniawan, "Arduino IoT Cloud," Beginning Arduino Nano 33 IoT, pp. 131–155, Jan. 2021.
6. M. A. Areqi, A. T. Zahary, and M. N. Ali, "State-of-the-Art Device-to-Device Communication Solutions," IEEE Access, vol. 11, pp. 46734–46764, 2023.
7. Hercog et al., "Design and Implementation of ESP32-Based IoT Devices," Sensors, vol. 23, no. 15, p. 6739, 2023.
8. H. Al-Obaidi and A. M. Al-Khafaji, "IoT Integration in Smart Cities," 2023.
9. P. A. Adepoju et al., "Recent Advances in Smart City Using IoT and 5G," 2025.
10. H. Rehan, "Information-Communication Technologies of IoT in Smart Cities," 2024.