MANTECH Publications

Journal of Embedded Systems & Its Applications

Volume 6 Issue 2, July-December 2021

Low-Power Embedded System Design for IoT Devices

Ridhima Setty, Subodh Jain

Associate Professor Assistant Professor

Department of Computer Engineering

Valley View College of Engineering

Ridhimasett30y@yahoo.com, jains6subodh@gmail.com

Abstract

Internet of Things (IoT) devices are increasingly used in smart homes, healthcare, agriculture, and industrial monitoring. These devices often operate on battery power for long periods, sometimes in remote or hard-to-reach locations. Therefore, designing low-power embedded systems has become a critical requirement in IoT development. Low power consumption not only increases battery life but also reduces maintenance cost and improves sustainability. This paper reviews various techniques, components, and architectural strategies used for low-power embedded system design in IoT devices. It discusses microcontroller selection, sleep modes, communication protocols, energy harvesting, software optimization, and power management strategies. The paper also provides tables and diagrams for better understanding of design approaches. The aim is to provide an overview of how engineers can design energy-efficient IoT nodes without compromising performance.

Keywords: *Low power design, Embedded systems, IoT devices, Power management, Energy efficiency, Sleep modes, Microcontrollers, Energy harvesting*

1. Introduction

The rapid growth of IoT devices has changed the way systems interact with the environment. These devices are embedded with sensors, microcontrollers, and communication modules. Most IoT devices work in conditions where frequent battery replacement is not practical. Hence, low-power embedded design is not optional but necessary.

Traditional embedded systems focused on performance, but modern IoT systems must balance performance with energy efficiency. A small sensor node may be required to run for months or years using a coin cell battery. This creates many challenges in hardware and software design.

Low-power design includes careful selection of hardware components, use of efficient communication methods, optimized firmware, and intelligent power management.

2. Overview of Embedded Systems in IoT (Elaborated)

An embedded system in an IoT device is a small, purpose-built computing unit designed to sense, process, and communicate data with minimal human intervention. Unlike general-purpose computers, these systems are designed for a specific task and operate under strict constraints such as limited power, memory, and processing capability. The architecture of an IoT embedded node is usually simple in structure but very carefully optimized for energy efficiency and reliability.

A typical IoT embedded device consists of the following main building blocks:

2.1 Microcontroller or Microprocessor

The microcontroller (MCU) acts as the brain of the IoT device. It controls all operations such as reading sensor data, processing information, managing communication, and controlling power modes.

In low-power IoT systems, microcontrollers are preferred over microprocessors because they consume less power and integrate memory and peripherals on a single chip. Commonly used MCUs include ARM Cortex-M series, MSP430, AVR, and ESP series.

Important features of MCUs for IoT include:

- Low active and sleep current
- Multiple sleep modes
- Built-in ADC, timers, UART, SPI, I2C interfaces
- Fast wake-up time from sleep
- Small memory footprint

The choice of MCU directly affects the energy efficiency and response time of the device.

2.2 Sensors and Actuators

Sensors are the input components of an IoT device. They collect environmental data such as temperature, humidity, pressure, motion, light intensity, gas concentration, or sound.

Actuators are output devices that perform physical actions like switching a relay, opening a valve, or moving a motor.

Sensors in IoT devices must be:

- Low power consuming
- Accurate and reliable
- Able to operate in sleep or standby modes
- Compatible with MCU interfaces

Sometimes, sensors are powered only when readings are required to save energy. Actuators, although used less frequently, may consume significant power and must be carefully controlled.

2.3 Communication Module

Communication modules allow IoT devices to transmit collected data to gateways, servers, or cloud platforms. Wireless communication is preferred in most IoT applications due to flexibility and ease of deployment.

Common communication technologies used:

- **Wi-Fi** for high data rate applications
- **BLE (Bluetooth Low Energy)** for short range and wearable devices
- **Zigbee** for home automation and mesh networking
- **LoRa/LoRaWAN** for long range, low data rate applications like agriculture

Since data transmission consumes high energy, the communication module must be selected based on range, data rate, and power budget.

2.4 Power Supply (Battery or Energy Harvesting)

Power supply is one of the most critical parts of an IoT embedded system. Most IoT devices operate on batteries such as lithium coin cells, AA batteries, or rechargeable Li-ion batteries. In remote applications, energy harvesting techniques like solar panels or vibration energy are used.

The power unit generally includes:

- Battery or energy source
- Voltage regulators or DC-DC converters
- Battery management system
- Charging circuit (if rechargeable)

Efficient power regulation ensures that minimal energy is wasted during voltage conversion.

2.5 Firmware for Control and Communication

Firmware is the software programmed into the MCU that controls the entire system. It decides when to read sensors, when to transmit data, and when to enter sleep mode. Efficient firmware design is essential for low-power operation. Good firmware practices include:

- Interrupt-driven operation instead of continuous polling
- Timed wake-up using RTC (Real Time Clock)
- Proper use of sleep and deep sleep modes
- Efficient communication scheduling
- Error handling and retry mechanisms

Firmware acts as the intelligence that balances performance and energy usage.

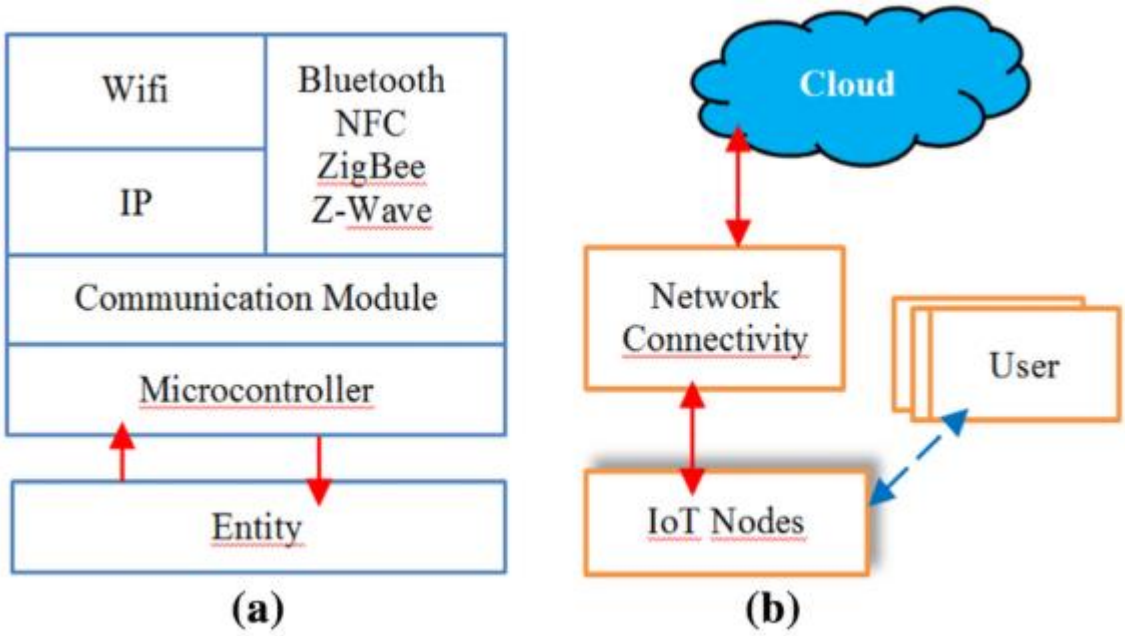

Figure 1: Basic IoT Embedded Node Architecture

Each block contributes to overall power consumption.

3. Sources of Power Consumption

Understanding where power is consumed inside an IoT embedded device is very important before trying to reduce it. In many designs, power is wasted not because of poor hardware, but because designers do not clearly identify which components are drawing current at

different stages of operation. An IoT node generally consumes power in sensing, processing, communication, and supporting circuitry.

The major sources of power consumption are discussed below.

3.1 Microcontroller (MCU) Activity

The microcontroller consumes power whenever it is in active mode executing instructions. The power usage mainly depends on:

- Clock frequency (higher frequency → higher current)
- Supply voltage
- Number of peripherals enabled
- Time spent in active mode

Even when no useful task is running, an MCU may still consume current if not placed into proper sleep mode. Many IoT devices waste energy because MCU stays active longer than required.

Key observation: Reducing active time and lowering clock speed significantly reduces energy consumption.

3.2 Wireless Communication Module

Wireless transmission is one of the highest power consuming operations in IoT devices. Sending data over Wi-Fi, BLE, Zigbee, or LoRa requires RF transmission which draws large current pulses.

For example:

- Wi-Fi transmission may draw 150–300 mA
- BLE transmission around 10–20 mA
- LoRa transmission around 30–40 mA

Even though transmission happens for a short time, the current is very high, which impacts battery life.

Important point: Reducing frequency of data transmission saves more power than reducing MCU computation.

3.3 Sensors

Sensors may look small, but some sensors draw continuous current if powered all the time. Gas sensors, GPS modules, and analog sensors can consume significant power.

Power depends on:

- Sampling rate
- Warm-up time
- Operating voltage
- Continuous vs intermittent operation

Sensors left ON continuously is a common mistake in low-power design.

3.4 LEDs, Displays, and Indicators

Visual indicators like LEDs and LCD/OLED displays consume unnecessary power in battery-operated IoT nodes.

- An LED can draw 5–20 mA continuously
- OLED/LCD displays require backlight power

These components are often not required in final deployment and should be removed or minimized.

3.5 Memory and Peripherals

External memory chips, ADCs, DACs, and timers also consume power when enabled.
Though individually small, together they contribute noticeable current.
Unused peripherals left enabled in MCU registers can increase consumption.

3.6 Voltage Regulators and Power Conversion Losses

Linear voltage regulators waste energy as heat when stepping down voltage. This is a hidden but important source of power loss.
For example:

- Converting 5V to 3.3V using linear regulator wastes energy
- DC-DC buck converters are more efficient (80–95%)

Poor regulator choice can reduce battery life significantly.

Table 1: Power in IoT devices is mainly consumed by:

Component	Power Usage Level	Remarks
Microcontroller	Medium	Depends on clock frequency
Sensors	Low to Medium	Depends on sampling rate

Component	Power Usage Level	Remarks
Communication module	High	Transmission is costly
LEDs/Displays	Medium	Avoid in battery systems
Memory and peripherals	Low	Often ignored but important

Communication consumes more power than computation in many IoT devices.

4. Low-Power Microcontroller Selection (Elaborated)

Selecting a suitable microcontroller (MCU) is one of the most important decisions in designing a low-power IoT embedded system. The MCU controls sensing, processing, communication, and power states. If the MCU itself consumes high current, no amount of optimization in other parts can fully compensate. Therefore, careful selection based on power characteristics and application needs is required.

Unlike traditional embedded designs where performance was the main concern, IoT designs focus more on energy efficiency, sleep behavior, and peripheral integration.

4.1 Why Microcontroller Choice Matters

The MCU operates in two primary states:

- **Active mode** – executing instructions, reading sensors, processing data
- **Sleep/standby mode** – waiting for interrupt or timer event

In IoT devices, the MCU stays in sleep mode for most of the time. Hence, sleep current is often more important than active current.

A microcontroller with 1 µA sleep current is far better than one with 10 µA sleep current when the device sleeps for hours.

Table 1: Choosing the right MCU is the first step. Popular low-power MCUs include:

MCU Family	Key Feature	Typical Current
ARM Cortex-M0+	Ultra low sleep current	< 2 µA
MSP430	Very low standby power	< 1 µA
AVR ATmega	Simple architecture	~5 µA sleep
ESP32	Wi-Fi + BLE but higher power	~10 µA sleep

Important factors:

- Multiple sleep modes
- Low active current
- Fast wake-up time
- Integrated peripherals

5. Sleep Modes and Duty Cycling

Most IoT embedded devices are not required to run continuously. In many applications such as environmental monitoring, smart agriculture, health tracking, or industrial sensing, the device only needs to wake up at certain intervals, collect data, transmit it, and then return to a low-power state. This pattern of operation is called **duty cycling**, and it is one of the most effective methods for reducing overall power consumption.

The basic idea is very simple: **keep the device in sleep mode for as long as possible and minimize the time spent in active mode**.

5.1 Understanding Sleep Modes

Modern microcontrollers provide several sleep modes to reduce power usage when the device is idle. In sleep mode, most parts of the MCU are turned off while only essential circuits remain active.

Common sleep modes include:

Sleep Mode	Description	Power Consumption
Idle Mode	CPU stopped, peripherals running	Moderate
Sleep Mode	CPU and some peripherals off	Low
Deep Sleep	Only RTC or interrupt active	Very Low
Standby/Hibernate	Almost complete shutdown	Ultra Low

In deep sleep or standby modes, current consumption can go below 1 μA.

5.2 What is Duty Cycling

Duty cycling is the technique where the device alternates between active and sleep states based on time or events.

For example, consider a temperature sensor node:

- Wake up every 10 minutes
- Read temperature
- Transmit data
- Go back to sleep

If the active time is 2 seconds and sleep time is 598 seconds, the duty cycle is very low, which results in huge power saving.

Duty Cycle Formula:

Duty Cycle=Active TimeTotal TimeDuty\ Cycle = \frac{Active\ Time}{Total\ Time}Duty Cycle=Total TimeActive Time

Lower duty cycle means lower average power consumption.

5.3 Role of RTC (Real Time Clock)

RTC is very important in duty-cycled systems. It runs even when MCU is in deep sleep and wakes it up after a predefined interval.

Without RTC, accurate timed wake-up is difficult and may consume more power.

5.4 Interrupt-Based Wake-up

Instead of waking up at fixed intervals, some IoT nodes wake up only when an event occurs.

Examples:

- Motion detected by PIR sensor
- Door opened using magnetic sensor
- Threshold temperature reached

This method reduces unnecessary wake-ups and saves more power.

5.5 Impact on Average Power Consumption

Consider an MCU drawing:

- 15 mA in active mode for 2 seconds
- 2 μA in sleep mode for 10 minutes

The average current becomes very small due to long sleep duration. This allows device to run for years on small battery.

5.6 Firmware Support for Duty Cycling

Proper firmware design is necessary to implement duty cycling:

- Configure sleep modes correctly
- Use timers and interrupts
- Turn off unused peripherals before sleep
- Save system state before entering sleep

If firmware is not properly written, MCU may not enter lowest power state.

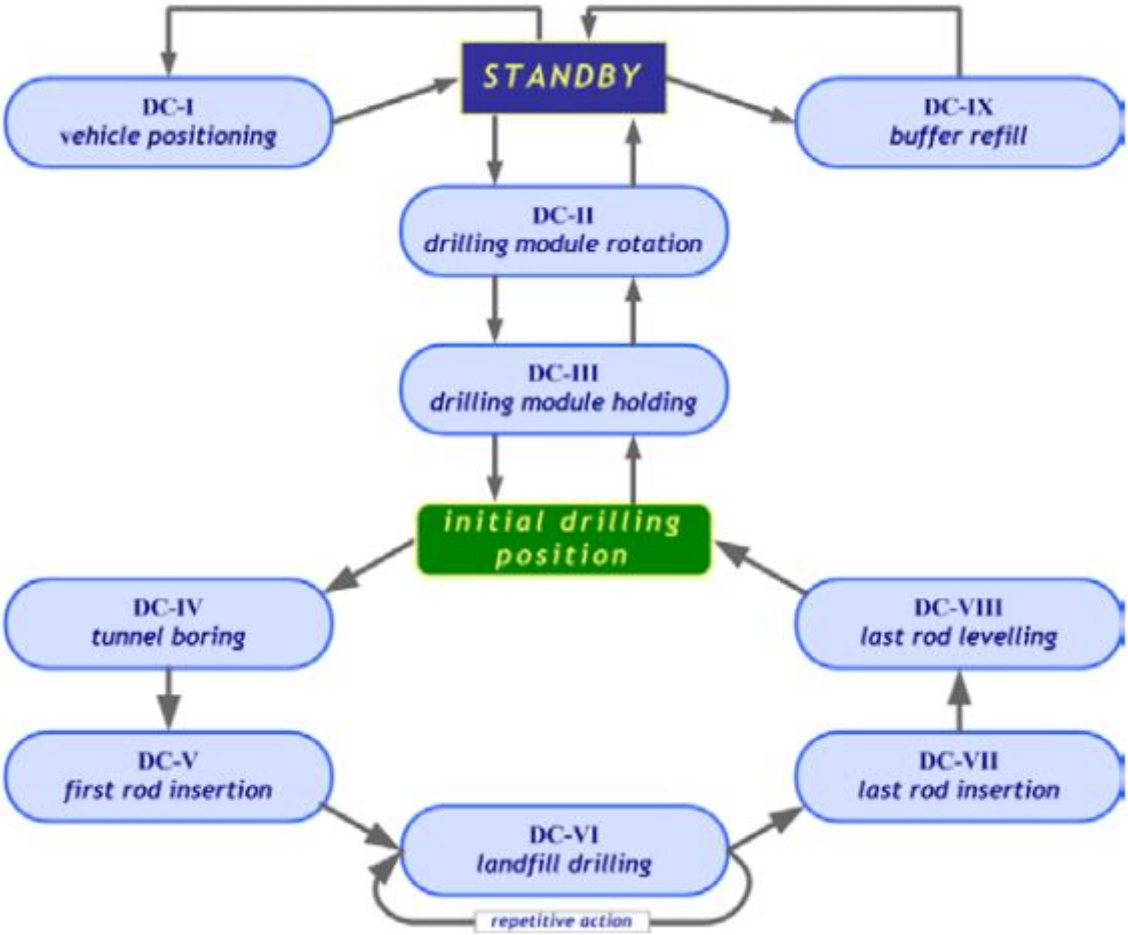

Figure 2: Duty Cycle Operation

Using deep sleep can reduce average power drastically.

6. Communication Protocols and Power

Wireless communication is power hungry. Choosing protocol wisely is important.

Protocol	Range	Power Use	Best Use Case
BLE	Short	Very Low	Wearables
Zigbee	Medium	Low	Home sensors
LoRa	Long	Very Low	Agriculture
Wi-Fi	Medium	High	High data rate

LoRa and BLE are more suitable for low-power IoT.

7. Sensor Optimization Techniques

Sensors can also consume power if kept active.

Techniques:

- Power sensors only when needed
- Reduce sampling rate
- Use interrupt based sensing
- Use analog sensors when possible

8. Power Management Techniques

Power management ICs (PMICs) and regulators help in efficiency.

- Use buck converters instead of linear regulators
- Dynamic voltage scaling
- Battery monitoring circuits
- Efficient charging circuits

9. Energy Harvesting for IoT

Some IoT devices use renewable sources.

Source	Use Case
Solar	Outdoor sensors
Vibration	Industrial machines
Thermal	Wearables
RF energy	Low power tags

Energy harvesting can extend life almost infinitely for small devices.

10. Firmware and Software Optimization

Software design affects power a lot.

- Use interrupts instead of polling
- Avoid unnecessary loops
- Reduce clock speed when possible
- Efficient coding practices
- Optimize memory usage

Poor coding can waste battery quickly.

11. Case Study: Low Power Environmental Monitoring Node

A typical node using:

- ARM Cortex M0+
- LoRa module
- Temperature & humidity sensor
- Solar panel + battery

The node wakes every 15 minutes, senses data, transmits, and sleeps.

Average current reduces to less than 50 µA.

12. Challenges in Low Power Design

- Balancing performance and power
- Cost constraints
- Environmental conditions
- Security requirements add processing
- Battery degradation over time

13. Future Trends

- Ultra low power AI chips
- TinyML for edge processing
- Advanced energy harvesting
- Low power wide area networks (LPWAN)
- Smart power management algorithms

14. Conclusion

Low-power embedded system design is a fundamental requirement for successful IoT deployments. Engineers must consider power at every stage: component selection, communication method, firmware design, and energy source. Using sleep modes, duty cycling, efficient protocols like LoRa and BLE, and energy harvesting methods can significantly improve battery life. Future developments in low-power processors and intelligent software will make IoT devices more sustainable and reliable. Proper design decisions can make IoT nodes operate for years without battery replacement, which is the ultimate goal for many applications.

References

1. P. Harrop, “Energy Efficient IoT Devices,” IDTechEx Report, 2022.
2. A. Dunkels, “Designing Low Power Wireless Sensor Networks,” IEEE Communications, 2021.
3. Texas Instruments, *MSP430 Low Power Design Guide*, 2020.
4. ARM Ltd., *Cortex-M Series Power Optimization Manual*, 2021.
5. J. Hui and D. Culler, “IP for Smart Objects,” ACM IoT Journal, 2020.
6. Microchip Technology, *AVR Low Power Tips*, Application Note, 2019.
7. Semtech, *LoRaWAN Low Power Design*, Technical Paper, 2022.
8. S. Raza et al., “Battery Lifetime Analysis of IoT Nodes,” IEEE Sensors Journal, 2021.
9. K. Ashton, “That ‘Internet of Things’ Thing,” RFID Journal, 2019.
10. N. S. Kim, “Energy Harvesting for Embedded Systems,” Elsevier Microelectronics, 2022.