IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Ultra Low-Power Embedded System Design Using Stm32 Cortex-M Microcontroller.

[1]Vithalani Paresh Keshar, [2]S. G. Prakash, [3]Savita Naik

[1]Senior Grade Lecturer, [2]Senior Grade Lecturer, [3]Lecturer

[1, 2, 3] Department of Electronics & Communication Engineering

[1,2,3] Govt polytechnic, Raichur, Karnataka, India

Abstract:

The rapid growth of battery-operated and portable electronic systems has created a strong demand for ultra low-power embedded system designs that can operate reliably for extended durations without frequent battery replacement. In this paper, an energy-efficient embedded system based on the STM32 Cortex-M microcontroller is presented, focusing on both hardware- and software-level power optimization techniques. The proposed system exploits the low-power operating modes of the STM32 microcontroller, including Sleep, Stop, and Standby modes, along with dynamic clock scaling and selective peripheral activation to minimize overall power consumption.

A structured power measurement methodology is adopted to analyze current consumption under different operating conditions such as active processing, idle state, and low-power modes. The hardware architecture incorporates an efficient voltage regulation unit, optimized clock circuitry, and peripheral interfaces designed to reduce leakage and switching losses. On the software side, interrupt-driven task execution and optimized firmware scheduling are employed to ensure minimal active time of the processor.

Experimental results demonstrate a significant reduction in power consumption compared to conventional always-on embedded designs, while maintaining acceptable system performance. The findings indicate that the proposed STM32-based architecture is well suited for low-power applications such as wireless sensor nodes, portable monitoring devices, and Internet of Things (IoT) systems. The presented design methodology serves as a practical reference for developing energy-efficient embedded systems using ARM Cortex-M microcontrollers.

Index Terms - Ultra Low-Power Design, Embedded Systems, STM32 Microcontroller, ARM Cortex-M, Power Optimization, Low-Power Modes, Energy-Efficient Computing, Battery-Operated Devices, Power Management Techniques.

I. INTRODUCTION

The increasing deployment of embedded systems in portable, battery-powered, and remotely deployed applications has intensified the need for energy-efficient computing platforms. Conventional embedded designs, which primarily focus on performance and functionality, often result in excessive power consumption and limited operational lifetime. As a consequence, power optimization has emerged as a critical design constraint alongside cost, reliability, and computational efficiency. Microcontrollers based on ARM Cortex-M architecture have gained significant attention due to their balance between processing capability and low-power operation, making them suitable for energy-constrained environments.

Advancements in semiconductor fabrication and power management techniques have enabled modern microcontrollers to operate at significantly reduced power levels without compromising real-time performance. Among these, the STM32 family of microcontrollers offers a wide range of low-power features such as multiple sleep states, flexible clock management, and peripheral-level power gating. These features allow designers to implement intelligent power-aware strategies that dynamically adapt system behavior based on operational requirements. However, achieving optimal power efficiency requires careful coordination between hardware design and embedded firmware, which remains a challenge in practical system development.

This work focuses on the design and evaluation of an ultra low-power embedded system using an STM32 Cortex-M microcontroller. By combining efficient hardware architecture with optimized software control, the proposed system demonstrates how substantial power savings can be achieved in real-world applications. The study emphasizes practical implementation aspects and experimental validation, making it relevant for engineers and researchers working in the field of low-power embedded system design.

1.1 Motivation for Ultra Low-Power Embedded Systems

The motivation for ultra low-power embedded system design originates from the growing reliance on battery-powered and energy-harvesting devices in modern applications. Systems such as wireless sensor nodes, portable medical instruments, and remote monitoring units are often deployed in environments where frequent battery replacement or recharging is impractical. In such scenarios, even marginal reductions in power consumption can significantly extend device lifetime and reduce maintenance costs.

Moreover, many embedded systems spend a large portion of their operational time in idle or low-activity states, waiting for external events or periodic data acquisition tasks. Traditional always-on designs fail to exploit these idle periods effectively, leading to unnecessary energy dissipation. Ultra low-power design techniques aim to minimize energy usage during both active and inactive periods by leveraging low-power modes, dynamic clock control, and event-driven processing. This approach ensures that the system consumes energy only when meaningful computation or communication is required.

The increasing emphasis on sustainable and environmentally responsible electronics further reinforces the need for energy-efficient embedded solutions. Reducing power consumption not only enhances battery life but also contributes to lower overall energy usage and heat dissipation, improving system reliability and long-term operational stability.

1.2 Applications of Energy-Efficient Microcontrollers

Energy-efficient microcontrollers play a vital role in a wide range of applications where power availability is limited or constrained. One of the primary application domains is wireless sensor networks, where sensor nodes are required to operate autonomously for extended periods while performing sensing, processing, and wireless communication tasks. Low-power microcontrollers enable such nodes to achieve long operational lifetimes without sacrificing responsiveness.

In the healthcare sector, portable and wearable medical devices rely heavily on low-power embedded platforms to ensure continuous monitoring and patient comfort. Devices such as portable diagnostic tools, health monitoring systems, and assistive technologies benefit from reduced power consumption, which allows compact battery designs and prolonged usage.

Industrial and consumer electronics applications also increasingly adopt energy-efficient microcontrollers for monitoring, control, and automation tasks. Smart meters, handheld instruments, and home automation systems require reliable operation with minimal energy consumption. The flexibility and low-power capabilities of STM32 Cortex-M microcontrollers make them suitable for these diverse application domains.

1.3 Contribution of the Proposed Work

The primary contribution of this work is the design and implementation of an ultra low-power embedded system based on the STM32 Cortex-M microcontroller, with a strong emphasis on practical power optimization strategies. The study presents a comprehensive approach that integrates hardware-level power management with software-driven control mechanisms to achieve significant energy savings.

Specifically, the proposed work evaluates the impact of various low-power operating modes on overall system consumption and demonstrates the effectiveness of dynamic clock scaling and selective peripheral activation. A systematic power measurement methodology is employed to analyze current consumption across different operational states, providing quantitative insights into energy usage patterns.

The experimental results validate the feasibility of achieving substantial power reduction without compromising system functionality or performance. By presenting both design methodology and empirical evaluation, this work contributes a practical reference framework for engineers and researchers seeking to develop energy-efficient embedded systems using STM32 and other ARM Cortex-M microcontrollers.

II. LITERATURE SURVEY

2.1 Low-Power Microcontroller Architectures

Low-power microcontroller architectures are designed to minimize energy consumption while maintaining sufficient computational capability for real-time embedded applications. Early low-power designs relied on reducing clock frequency and supply voltage; however, these approaches often resulted in degraded system performance. To overcome these limitations, modern microcontroller architectures integrate multiple power domains, flexible clock trees, and low-leakage fabrication technologies.

ARM Cortex-M–based microcontrollers have emerged as a dominant solution in low-power embedded applications due to their efficient instruction set, reduced pipeline complexity, and support for various low-power operational modes. Studies have shown that the simplified architecture of Cortex-M cores significantly reduces switching activity compared to traditional microcontroller architectures, leading to lower dynamic power consumption. Additionally, the introduction of deep sleep and standby modes enables substantial reductions in static power during idle periods.

Manufacturers such as STMicroelectronics, NXP, and Texas Instruments have incorporated advanced power management features into their microcontroller families. In particular, STM32 microcontrollers provide fine-grained control over clock sources and peripheral activation, allowing designers to tailor power consumption based on application requirements. However, existing literature indicates that these architectural advantages can only be fully realized when supported by appropriate firmware-level power management strategies.

2.2 Power Optimization Techniques in Embedded Systems

Power optimization in embedded systems is typically addressed through a combination of hardware- and software-based techniques. Hardware-level approaches include voltage regulation optimization, clock gating, and the selective disabling of unused peripherals. These techniques aim to reduce both dynamic and leakage power consumption by minimizing unnecessary switching and idle power draw.

On the software side, several studies emphasize the importance of power-aware firmware design. Event-driven programming models, interrupt-based execution, and efficient task scheduling have been shown to significantly reduce active processing time. Dynamic clock scaling and adaptive power mode switching further enhance energy efficiency by matching system performance to workload demands.

Another widely explored technique involves the strategic use of low-power operating modes such as sleep, stop, and standby states. Research demonstrates that transitioning the microcontroller into low-power modes

during idle intervals can yield substantial energy savings, particularly in applications with low duty cycles. However, improper mode selection or frequent transitions may introduce latency and energy overheads, highlighting the need for careful design trade-offs.

Despite extensive research, many existing studies focus on theoretical models or simulation-based evaluations. Practical implementation challenges, such as peripheral wake-up latency and real-time constraints, are often underexplored. This gap motivates the need for experimental validation of power optimization techniques in real embedded hardware platforms.

2.3 Comparative Analysis of ARM Cortex-M Based Systems

Comparative studies of ARM Cortex-M–based microcontrollers have analyzed trade-offs among performance, power consumption, and system flexibility. Cortex-M0 and Cortex-M0+ cores are often preferred for ultra low-power applications due to their minimal instruction set and reduced power footprint, whereas Cortex-M3 and Cortex-M4 cores offer higher computational performance with slightly increased power consumption.

Research comparing STM32 microcontrollers with other Cortex-M–based platforms highlights the effectiveness of STM32's power management infrastructure, particularly its low-power modes and peripheral gating capabilities. The availability of multiple clock sources and scalable operating frequencies enables STM32 devices to operate efficiently across diverse workloads.

However, existing comparative analyses indicate that power efficiency is highly application-dependent and influenced by both hardware configuration and firmware design. While vendor datasheets provide nominal power consumption figures, real-world measurements often vary due to system-level factors. This observation underscores the importance of application-specific power profiling and experimental evaluation, as addressed in the proposed work.

III. PROBLEM STATEMENT & OBJECTIVES

3.1 Identified Power Consumption Challenges

Several power-related challenges have been identified in conventional embedded system implementations:

- **Inefficient Idle Power Management:**

Embedded systems often remain in active or semi-active states during idle periods, resulting in unnecessary energy consumption.

- **Static Clock and Voltage Configuration:**

Fixed clock frequencies and supply voltages lead to excess power usage during low computational workloads.

- **Uncontrolled Peripheral Power Usage:**

Peripherals such as timers, communication interfaces, and sensors frequently remain enabled even when not required, increasing overall system power draw.

- **Lack of Coordinated Hardware–Software Design:**

Inadequate synchronization between hardware power features and firmware control limits the effectiveness of available low-power modes.

- **Limited Experimental Power Profiling:**

Many designs lack systematic measurement and analysis of power consumption under different operating modes, making optimization difficult.

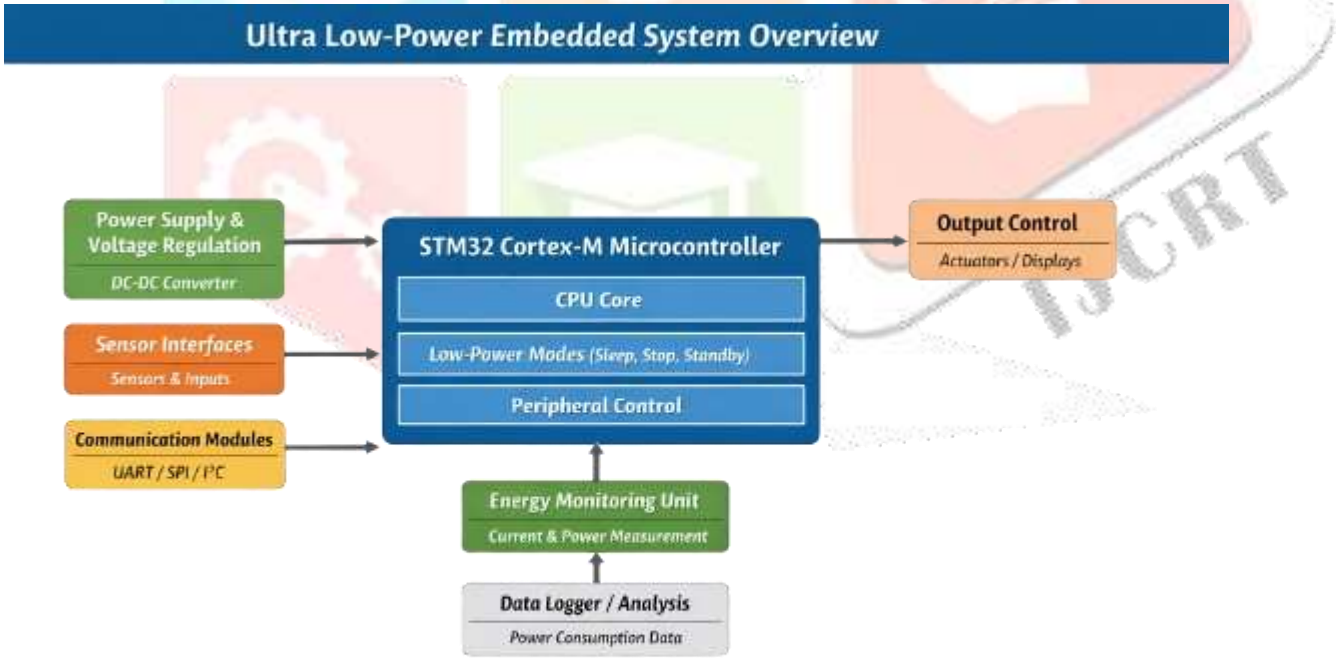
These challenges highlight the need for a structured design methodology that integrates power-aware hardware configuration with optimized firmware execution.

3.2 Research Objectives

The primary objective of this research is to design and evaluate an ultra low-power embedded system using an STM32 Cortex-M microcontroller through effective hardware and software optimization techniques. The specific objectives of the study are as follows:

1. To analyze the power consumption characteristics of the STM32 Cortex-M microcontroller under different operating modes, including active, idle, and low-power states.
2. To design a hardware architecture that supports efficient power regulation, clock management, and peripheral control.
3. To develop power-aware embedded firmware utilizing dynamic clock scaling, interrupt-driven execution, and selective peripheral activation.
4. To implement a systematic power measurement methodology for evaluating energy consumption in real-time embedded operation.
5. To experimentally validate the effectiveness of the proposed low-power design by comparing power usage across different operational scenarios.
6. To demonstrate the suitability of the proposed system for battery-operated and energy-sensitive applications.

IV. SYSTEM OVERVIEW



4.1 Overall System Architecture

The overall system architecture follows a centralized microcontroller-based design, where the STM32 Cortex-M device acts as the core processing unit. The system receives input signals from external sensors or peripheral devices and performs necessary data processing and control operations. Output signals are generated to drive external modules or communication interfaces as required.

The power supply unit provides a stable regulated voltage to the microcontroller and peripheral circuits, ensuring efficient operation with minimal power loss. Clock sources are configured to support dynamic frequency scaling, enabling the system to operate at lower frequencies during periods of reduced computational demand. Peripheral modules are interfaced through standard communication protocols such as GPIO, UART, SPI, or I²C, and are activated only when required.

An energy monitoring circuit is incorporated to measure current consumption under various operating conditions. This enables real-time evaluation of power usage during active processing, idle states, and low-power modes. The modular architecture facilitates systematic experimentation and performance analysis, making the system suitable for evaluating power optimization techniques in embedded environments.

4.2 STM32 Cortex-M Microcontroller Overview

The STM32 Cortex-M microcontroller family is based on the ARM Cortex-M architecture, which is optimized for low-power and real-time embedded applications. These microcontrollers offer a reduced instruction set, efficient interrupt handling, and deterministic execution behavior, making them suitable for energy-constrained systems.

STM32 devices integrate multiple low-power operating modes, including Sleep, Stop, and Standby modes, which significantly reduce power consumption during idle periods. The microcontroller supports flexible clock configurations using internal and external oscillators, allowing designers to balance performance and power usage. Additionally, on-chip peripherals such as timers, communication interfaces, and analog modules can be individually enabled or disabled to minimize unnecessary energy consumption.

The availability of low-power voltage regulators and fast wake-up mechanisms further enhances the suitability of STM32 microcontrollers for ultra low-power applications. These features enable rapid transitions between low-power and active states, ensuring energy efficiency without compromising system responsiveness.

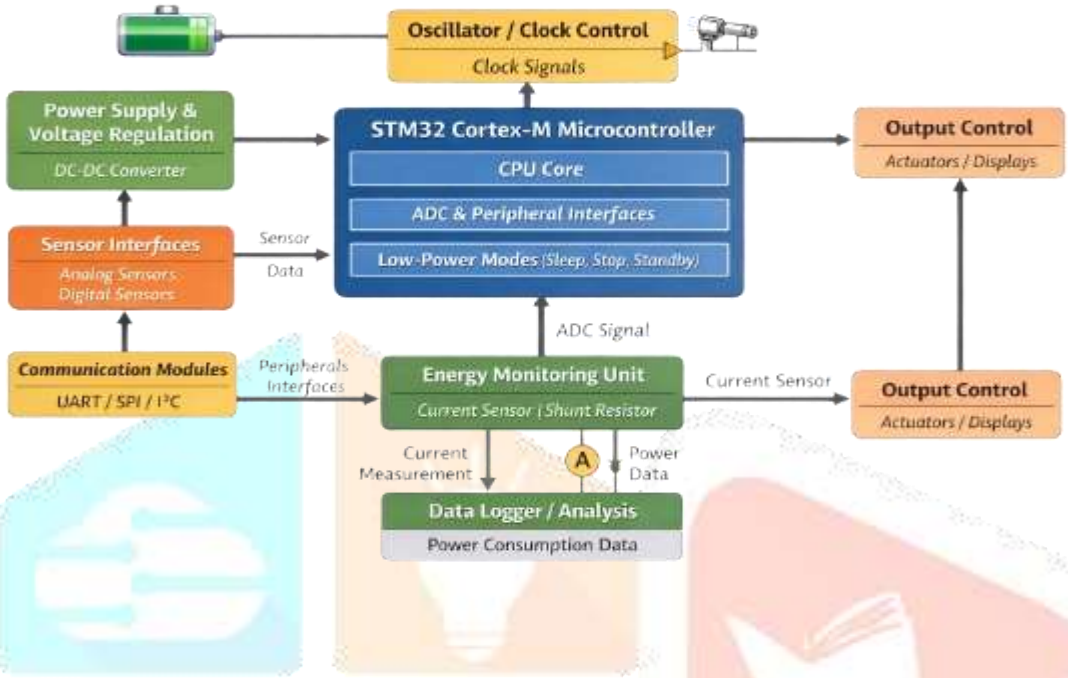
4.3 Power Management Strategy

The power management strategy adopted in the proposed system combines hardware-level control with software-based optimization techniques. At the hardware level, efficient voltage regulation and clock management are employed to reduce power dissipation. Unused peripherals are electrically disabled, and low-power clock sources are utilized during idle operation.

At the software level, the embedded firmware is designed using an interrupt-driven execution model, allowing the microcontroller to remain in low-power modes for extended durations. Dynamic clock scaling is implemented to adjust the processor frequency based on task requirements. During periods of inactivity, the system transitions into appropriate low-power modes, such as Sleep or Stop mode, to minimize energy consumption.

The power management strategy is validated through systematic measurement of current consumption across different operational states. This coordinated approach ensures optimal utilization of the STM32 microcontroller's low-power features and demonstrates the effectiveness of the proposed design in achieving ultra low-power operation.

V.HARDWARE DESIGN



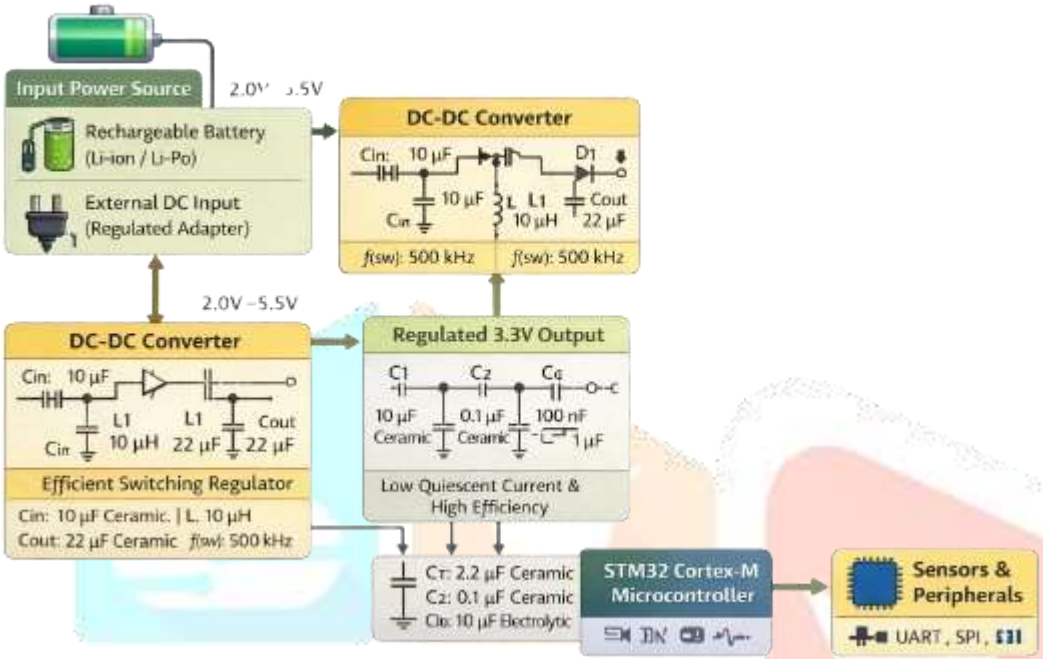
5.1 STM32 Cortex-M Microcontroller

The core of the proposed system is an STM32 Cortex-M–based microcontroller, selected for its low-power architecture and integrated power management features. The ARM Cortex-M core provides efficient instruction execution, fast interrupt handling, and deterministic real-time performance, making it suitable for energy-constrained embedded applications.

STM32 microcontrollers support multiple low-power operating modes, including Sleep, Stop, and Standby, which significantly reduce power consumption during idle periods. The device integrates on-chip Flash and SRAM memory, reducing the need for external memory components and associated power overhead. Additionally, a wide range of built-in peripherals such as timers, analog-to-digital converters (ADC), communication interfaces, and GPIO ports enable compact and energy-efficient system design.

The microcontroller operates at low supply voltages and supports flexible clock configurations, allowing designers to trade off processing performance against power consumption. These characteristics make the STM32 Cortex-M an ideal platform for ultra low-power embedded system development.

5.2 Power Supply and Voltage Regulation Unit



An efficient power supply and voltage regulation unit is essential for minimizing overall system power consumption. The proposed system employs a regulated DC power source derived from a battery or external adapter. A low-dropout (LDO) voltage regulator or DC–DC converter is used to provide a stable operating voltage to the STM32 microcontroller and peripheral circuits.

The voltage regulator is selected based on low quiescent current and high efficiency at low load conditions, which is critical for battery-powered applications. Proper decoupling capacitors are placed close to the microcontroller power pins to reduce noise and improve voltage stability. This configuration ensures reliable operation while minimizing energy losses due to inefficient regulation.

5.3 Clock Configuration and Peripheral Control

Clock configuration plays a vital role in determining the power consumption of an embedded system. In the proposed design, the STM32 microcontroller utilizes internal low-frequency and high-frequency oscillators to support dynamic clock scaling. During low computational workloads, the system operates at reduced clock frequencies to minimize dynamic power consumption.

Peripheral clocks are independently controlled through the microcontroller's clock gating mechanism. Unused peripherals are disabled to prevent unnecessary switching activity and leakage power. This selective peripheral control significantly reduces overall power consumption, especially in applications where only a subset of peripherals is required at a given time.The flexible clock architecture allows the system to achieve an optimal balance between performance and energy efficiency across different operating scenarios.

5.4 Sensor and External Interface Modules

The proposed system supports interfacing with external sensors and modules for data acquisition and control. Sensors are connected to the STM32 microcontroller through analog or digital interfaces such as ADC channels, GPIO pins, I²C, SPI, or UART communication protocols.

To reduce power consumption, sensor modules are activated only when required for data sampling or event detection. External interfaces are configured to operate in low-power modes whenever possible, and

communication transactions are designed to be brief and event-driven. This approach minimizes active processing time and reduces energy usage associated with data transmission.

5.5 Energy Measurement Circuit

An energy measurement circuit is incorporated into the hardware design to monitor and analyze system power consumption. The circuit typically consists of a low-value shunt resistor placed in series with the power supply line, along with a current sensing amplifier or measurement module.

The voltage drop across the shunt resistor is measured to calculate the instantaneous current drawn by the system. These measurements are logged and analyzed under different operating conditions, such as active processing, idle state, and low-power modes. This enables accurate evaluation of the effectiveness of the implemented power optimization techniques.

VI. SOFTWARE DESIGN & IMPLEMENTATION

6.1 Firmware Development Environment (IDE & Tools)

The embedded firmware is developed using a standard integrated development environment (IDE) suitable for STM32 microcontrollers. Tools such as STM32CubeIDE or equivalent ARM-based development environments are utilized for code development, compilation, debugging, and flashing. The development environment supports C programming language, which provides low-level hardware access and efficient execution required for embedded applications.

Peripheral configuration and clock setup are performed using vendor-provided configuration utilities, which simplify the initialization process and ensure correct usage of low-power features. Debugging and testing are carried out using in-circuit debugging interfaces such as SWD or JTAG, allowing real-time observation of system behavior and power mode transitions.

The use of standardized development tools ensures portability, maintainability, and reproducibility of the proposed software design.

6.2 Low-Power Operating Modes

The STM32 Cortex-M microcontroller provides multiple low-power operating modes that enable significant reduction in energy consumption during idle periods. These modes are effectively utilized in the proposed system based on operational requirements.

- **Sleep Mode:**

In Sleep mode, the CPU clock is halted while peripheral clocks may remain active. This mode is used during short idle intervals where quick wake-up is required.

- **Stop Mode:**

Stop mode disables the main clocks and reduces power consumption further by retaining memory contents and register states. The system enters this mode during extended idle periods while maintaining fast recovery capability.

- **Standby Mode:**

Standby mode offers the lowest power consumption by shutting down most internal circuitry. It is used when the system remains inactive for long durations, with wake-up triggered by external interrupts or reset events.

The firmware dynamically selects the appropriate low-power mode based on system state and workload characteristics to maximize energy savings.

6.3 Peripheral Power Control Algorithm

Peripheral power consumption is managed through a software-based peripheral control algorithm. The algorithm ensures that peripherals are enabled only when required and disabled immediately after task completion. This minimizes unnecessary power usage caused by idle peripherals.

The algorithm follows a sequence of initialization, active operation, and deactivation for each peripheral. Clock gating mechanisms are employed to disable peripheral clocks when not in use. Communication interfaces and sensors are activated on demand and placed in low-power or shutdown states during idle periods.

This structured peripheral control strategy significantly reduces both dynamic and static power consumption, contributing to the overall ultra low-power performance of the system.

6.4 Task Scheduling and Interrupt Handling

The proposed system adopts an interrupt-driven task execution model to minimize continuous processor activity. Instead of using polling-based mechanisms, tasks are triggered by hardware interrupts generated by timers, sensors, or communication events.

A lightweight scheduling approach is implemented, where the processor remains in a low-power state until an interrupt occurs. Upon interrupt detection, the required task is executed, and the system returns to an appropriate low-power mode after task completion. This approach ensures efficient utilization of processor resources and reduces active processing time.

Interrupt priorities are carefully configured to handle time-critical events while maintaining predictable system behavior. The combination of interrupt-driven execution and optimized scheduling enhances energy efficiency without compromising system responsiveness.

VII. POWER OPTIMIZATION TECHNIQUES

7.1 Dynamic Clock Scaling

Dynamic clock scaling is an effective technique for reducing dynamic power consumption in embedded systems. Since processor power consumption is directly proportional to clock frequency, lowering the operating frequency during low computational demand significantly reduces energy usage.

In the proposed system, the STM32 microcontroller dynamically adjusts its clock frequency based on the current workload. High-frequency operation is used only during computation-intensive tasks such as data processing or communication, while lower frequencies are selected during idle or monitoring phases. Internal oscillators and clock prescalers are configured to support seamless transitions between different frequency levels.

This adaptive clock management approach ensures that the processor operates at the minimum required frequency for a given task, thereby reducing unnecessary switching activity and overall power dissipation.

9.2 Peripheral Gating Techniques

Peripheral gating techniques play a crucial role in minimizing power consumption by disabling unused hardware modules. Many embedded systems suffer from excessive power usage due to peripherals remaining active even when not required.

The STM32 microcontroller supports independent clock gating for on-chip peripherals, allowing each module to be enabled or disabled through software control. In the proposed design, peripherals such as timers,

communication interfaces, and analog modules are activated only during task execution and immediately disabled upon task completion.

By selectively gating peripheral clocks, both dynamic and leakage power consumption are significantly reduced. This approach is particularly effective in systems where peripheral usage is sporadic or event-driven, resulting in substantial energy savings during idle periods.

9.3 Software-Based Power Reduction Methods

Software-based power reduction methods complement hardware-level optimization techniques by minimizing processor active time and improving execution efficiency. The proposed firmware is designed using an interrupt-driven architecture, eliminating continuous polling and reducing unnecessary CPU activity.

Task execution is optimized by grouping operations and minimizing context switching overhead. Delay-based loops are avoided, and timer interrupts are used instead to schedule tasks efficiently. Additionally, the firmware ensures that the system enters appropriate low-power modes immediately after completing required operations.

Memory usage is optimized to reduce access time and energy consumption, and unnecessary debugging features are disabled in the final deployment. These software-level optimizations collectively contribute to reducing overall power consumption and extending system operational lifetime.

REFERENCES

[1] ARM Ltd., *ARM Cortex-M3 Processor Technical Reference Manual*, ARM DDI 0337, 2011.

[2] ARM Ltd., *Low-Power Design Techniques for ARM Cortex-M Processors*, ARM White Paper, 2012.

[3] STMicroelectronics, *STM32F1 Series Reference Manual*, RM0008, Rev. 10, 2013.

[4] STMicroelectronics, *Getting Started with Ultra-Low-Power STM32 Microcontrollers*, Application Note AN3364, 2014.

[5] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed., Prentice Hall, 2011.

[6] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design & Test of Computers*, vol. 18, no. 2, pp. 62–74, 2011.

[7] D. Pletcher, "Low-power embedded system design using ARM microcontrollers," *IEEE Embedded Systems Letters*, vol. 4, no. 3, pp. 69–72, 2012.

[8] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual*, Springer, 2012.

[9] N. Kim, T. Austin, D. Blaauw, et al., "Leakage current: Moore's law meets static power," *IEEE Computer*, vol. 36, no. 12, pp. 68–75, 2011.

[10] P. Dutta, D. Culler, and S. Shenker, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," *ACM SenSys*, pp. 71–84, 2012.

[11] S. Mahlknecht and M. Bösch, "Performance evaluation of low-power microcontrollers for embedded systems," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 6, pp. 2573–2582, 2013.

[12] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," *USENIX Annual Technical Conference*, pp. 271–284, 2014.

[13] K. Lee and D. Lee, "Energy-efficient task scheduling for embedded real-time systems," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 148–155, 2013.

[14] STMicroelectronics, *STM32 Low-Power Modes and Power Consumption Measurement*, Application Note AN2629, 2015.

[15] R. Zurawski, *Embedded Systems Handbook*, CRC Press, 2012.