**Module: hal\_core\_pm (Power Management)**

**Overview**

The hal\_core\_pm module is responsible for **power management (PM)** within the **Hardware Abstraction Layer (HAL)**. It provides interfaces and mechanisms for **efficient power utilization**, **dynamic power scaling**, **low-power states**, and **energy-aware scheduling**. The module ensures that power consumption is optimized without compromising **system performance, reliability, and responsiveness**.

**Key Responsibilities of hal\_core\_pm**

1. **Dynamic Power Scaling**
   * Adjusts **CPU, GPU, NPU, and memory power states** based on system demand.
   * Uses **Dynamic Voltage and Frequency Scaling (DVFS)** for power-efficient performance.
   * Supports **AI-powered workload prediction** for proactive power scaling.
2. **Sleep and Low-Power States**
   * Provides various **power-saving states** such as:
     + **Idle Mode** – Low-power state when no tasks are running.
     + **Sleep Mode** – Reduces clock speed and voltage when inactive.
     + **Deep Sleep Mode** – Powers down non-essential components while keeping essential tasks active.
     + **Hibernation** – Saves system state and shuts down almost all components.
   * Manages **power wake-up sources** such as timers, interrupts, and external events.
3. **Energy-Aware Scheduling**
   * Works with the scheduler to allocate tasks **based on energy efficiency**.
   * Ensures high-priority tasks run on **power-efficient cores** while background tasks are delayed or batched.
4. **Peripheral and Device Power Control**
   * Enables **dynamic power gating** to shut down unused hardware components.
   * Supports **runtime power management (RPM)** for on-demand device activation.
5. **Thermal Management & Protection**
   * Monitors **temperature sensors** to prevent overheating.
   * Throttles performance or **engages cooling mechanisms** when required.
6. **Battery & Energy Harvesting Support**
   * Manages **battery charging, power usage, and discharge cycles**.
   * Interfaces with **renewable energy sources** (solar, kinetic, etc.) for energy harvesting.
7. **Power Security & Isolation**
   * Ensures **secure power domains** for trusted execution environments.
   * Prevents **malware or rogue applications** from over-consuming power resources.

**Power Management Workflow**

**1. Power Request Detection**

* The system detects power management events (e.g., **idle system state, high CPU load, overheating, user-defined policies**).

**2. Power Adjustment Decision**

* The **Power Governor** determines the best power mode based on system workload.
* If **low activity is detected**, the system transitions to a lower power state.
* If **high performance is required**, the system scales up voltage and frequency.

**3. Power State Transition**

* The system applies **DVFS adjustments** to processors, memory, and peripherals.
* Components that are **not in use** are powered down.

**4. Wake-up Event Handling**

* Interrupts or system events (e.g., user interaction, scheduled tasks) trigger a wake-up.
* The system **restores previous states** and resumes normal operations.

**Key Components of hal\_core\_pm**

| **Component** | **Description** |
| --- | --- |
| **Power Governor** | Dynamically manages power scaling and optimization. |
| **DVFS Engine** | Adjusts CPU/GPU/NPU voltage and frequency based on workload. |
| **Sleep & Wake Controller** | Manages low-power transitions and wake-up mechanisms. |
| **Thermal Protection** | Monitors temperatures and prevents overheating. |
| **Battery & Energy Harvester** | Optimizes power from battery and external energy sources. |

**Example Power Management Code**

**Adjusting Power States**

#include "hal\_core\_pm.h"

// Set CPU to low-power mode

void enter\_low\_power\_mode() {

hal\_pm\_set\_power\_state(HAL\_PM\_SLEEP\_MODE);

}

// Wake up system when needed

void wake\_up\_system() {

hal\_pm\_wake\_up\_event(HAL\_WAKEUP\_TIMER);

}

## ****Integration with Other HAL Components****

| **HAL Component** | **Role in Power Management** |
| --- | --- |
| hal\_cpu | Adjusts CPU power levels based on workload. |
| hal\_mem\_manager | Controls memory power states and optimizations. |
| hal\_sensors | Monitors temperature and system activity. |
| hal\_connectivity | Manages power for network interfaces (WiFi, 5G). |
| hal\_trustzone | Ensures secure power management operations. |

## ****Future Enhancements****

* **AI-Powered Energy Optimization**
  + Predictive analysis for **workload-based power scaling**.
* **Autonomous Power Management**
  + Self-adjusting power profiles based on usage patterns.
* **Quantum & Neuromorphic Power Optimization**
  + Custom power management for **QPU/NPU workloads**.

## ****Summary****

The **hal\_core\_pm** module ensures **efficient power consumption**, **dynamic power scaling**, **low-power modes**, and **thermal protection**. It optimizes energy use across processors, memory, and peripherals while ensuring **performance and security**.