Power Management Service

Reaching low and high power states in TF-M

Frank Audun Kvamtrø

September 2025

Introduction

Rationale for this service

- Power management needs to be handled in the **Root of Trust**
- A well-defined service is much better than using e.g. **IOCTL** for a fully vendor-customized solution

Intent

- Add support standardized low-power modes
  - **Suspend**, **Hibernate** and **System Off** - *all optional to implement*
- Allow for **device-specific** power modes
  - Extendible on vendor level
- Provide integration points to ease development and adoption
  - Matched with relevant documentation, possibly including reference-kit implementation
- Provide a service that is certifiable

Service API – Platform level

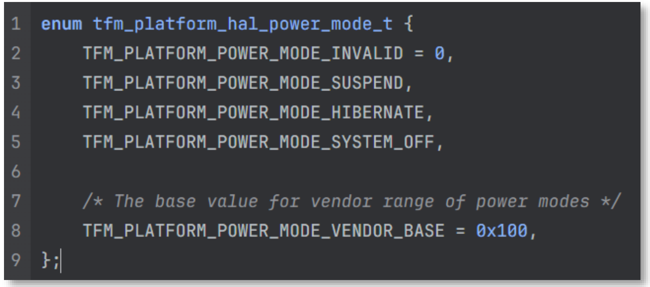
- Enum-based signals to set power modes
- Vendor range for extensions
  - Base + index

Reusing error-codes from platform layer

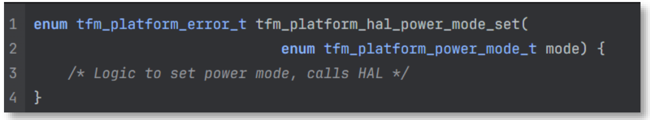
- TFM_PLATFORM_ERR_SUCCESS
  - Mode change successful - or no signal if the power mode stalls the device
- TFM_PLATFORM_ERR_SYSTEM_ERROR
  - Internal hardware error
- TFM_PLATFORM_ERR_NOT_SUPPORTED
  - Mode not enabled/supported

Option: Adding new error-code:

- TFM_PLATFORM_ERR_BUSY – Mode change deferred
  - It is the assumption that the easiest way to track whether a power mode change can happen is by e.g. storing a generic "busy state". This limits the need to give the Power Management Service knowledge about the rest of the system.



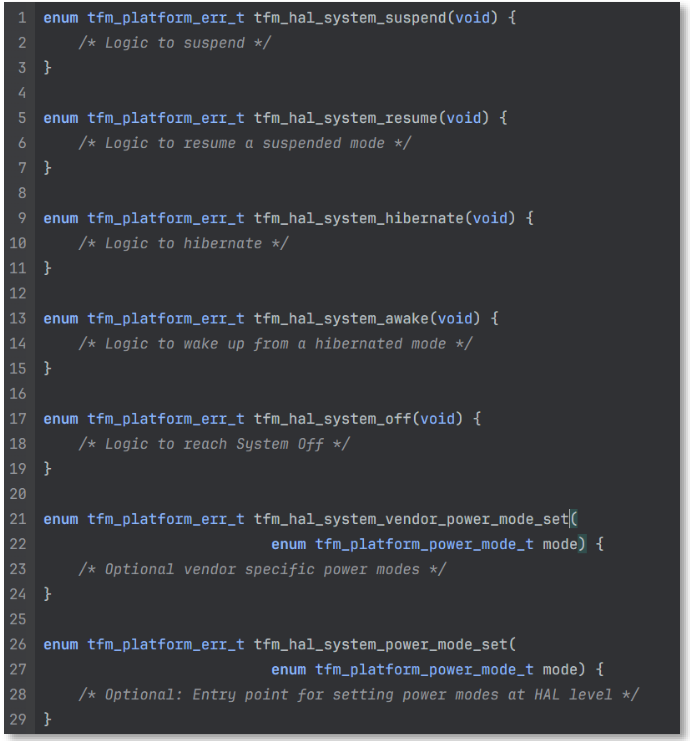
```
enum tfm_platform_hal_power_mode_t {
    TFM_PLATFORM_POWER_MODE_INVALID = 0,
    TFM_PLATFORM_POWER_MODE_SUSPEND,
    TFM_PLATFORM_POWER_MODE_HIBERNATE,
    TFM_PLATFORM_POWER_MODE_SYSTEM_OFF,

    /* The base value for vendor range of power modes */
    TFM_PLATFORM_POWER_MODE_VENDOR_BASE = 0x100,
};
```



```
enum tfm_platform_error_t tfm_platform_hal_power_mode_set(
                              enum tfm_platform_power_mode_t mode) {
    /* Logic to set power mode, calls HAL */
}
```

HAL Level

- Differentiated APIs for **Suspend**, **Hibernate**, and **System Off** power modes
  - Suspend + Resume
  - Hibernate + Awake
  - System off has matching wakeup API, use regular reset..
- Optional: Custom function for vendor-specific power modes
  - It is assumed that the vendor handles going to a “higher” power level e.g. with the same service API
  - No assumption on change to overall service of the device when any vendor specific power mode is reached
- Optional: API to handle all power modes at HAL level
  - Could be handled on service level instead…



```c
enum tfm_platform_err_t tfm_hal_system_suspend(void) {
    /* Logic to suspend */
}

enum tfm_platform_err_t tfm_hal_system_resume(void) {
    /* Logic to resume a suspended mode */
}

enum tfm_platform_err_t tfm_hal_system_hibernate(void) {
    /* Logic to hibernate */
}

enum tfm_platform_err_t tfm_hal_system_awake(void) {
    /* Logic to wake up from a hibernated mode */
}

enum tfm_platform_err_t tfm_hal_system_off(void) {
    /* Logic to reach System Off */
}

enum tfm_platform_err_t tfm_hal_system_vendor_power_mode_set(
                                  enum tfm_platform_power_mode_t mode) {
    /* Optional vendor specific power modes */
}

enum tfm_platform_err_t tfm_hal_system_power_mode_set(
                                  enum tfm_platform_power_mode_t mode) {
    /* Optional: Entry point for setting power modes at HAL level */
}
```

Controlling scope – At least in the beginning...

- Manage power mode with simple information exchange
  - Ensuring service API can be stable and generic
- HAL level entry-points hides implementation details
  - No special handling of reserved memory, peripheral usage and/or understanding of chips/cores in service
  - Vendor can extend with additional power modes according to their needs
- Separation of concern between NSPE and SPE by not transferring data
  - Current design: No wakeup signal emitted on service level

Open question:

- For simplification, the platform and HAL abstraction layer is somewhat "mixed"
  - Would this be acceptable or do we require **service to HAL** conversions for error-codes, modes, and configurations?

Next steps

- PR raised with System Off example on nRF device
- Follow up either in PR or in mailing list
- Discussing acceptance and timeline for official support…