Power Management on Zephyr

Vincent Wan
Linaro Assignee - Texas Instruments

Linaro TECH DAYS

Zephyr and Power Management

- “Resource-constrained” includes constraint on power consumption
- Zephyr has an existing power management subsystem

Power management features (as of Zephyr 2.20)

System Power Management

- Sleep and deep sleep states
- PM policy
  - Residency-based default
- Power state locking
- Power state forcing

Device Power Management

- Centralized device PM
  - Active, low power, suspend, off
- Busy status indication
- Device idle power management

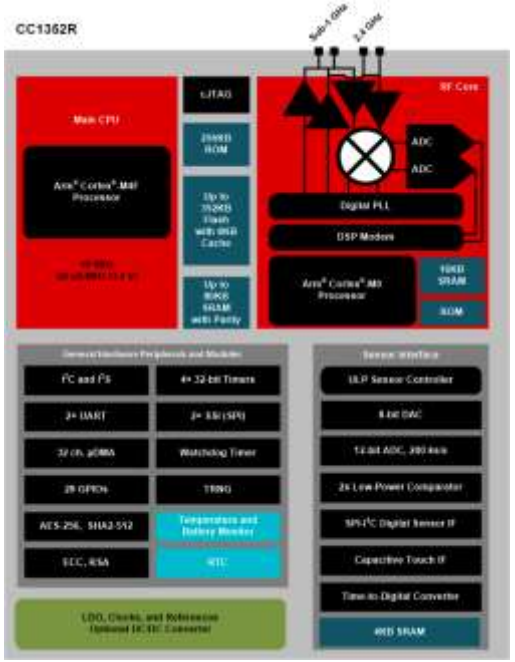
Bonus: Tickless Kernel

More info available in Zephyr documentation:
https://docs.zephyrproject.org/latest/reference/power_management/index.html

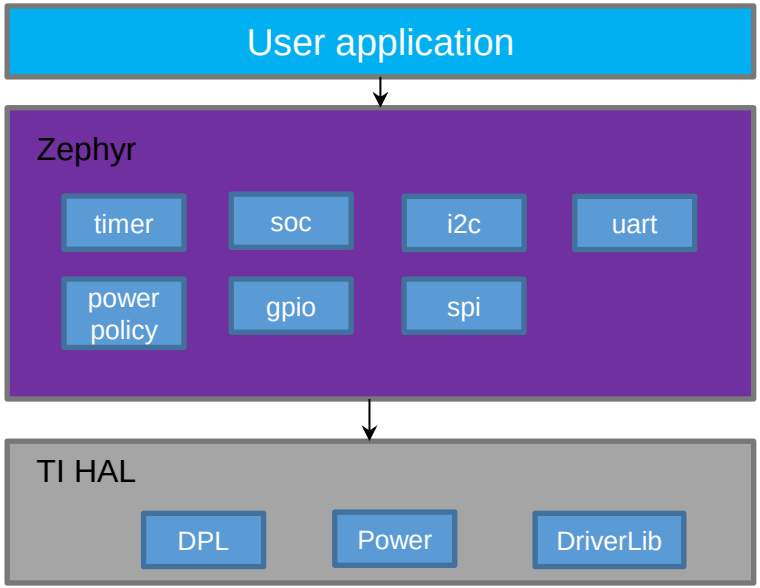
Power management support on CC1352R1

CC1352R1 LaunchPad

- ARM Cortex M4F
- Active mode: MCU @48 MHz, 2.9 mA
- Idle: 590 uA
- Standby: 0.85 µA (RTC on, 80KB RAM and CPU retention, peripherals off)
- Shutdown: 150 nA (wakeup on external events)



Implementation on CC1352R1



- Lean on Power Manager supplied by TI HAL:
  - Power_sleep()
    - For standby mode
  - Power dependencies
    - Auto on/off of peripherals and power domains depending on resources needed (reference-counting)
  - Power constraints
    - Anticipated need in TI's RF driver
  - Power notifications
    - Ensure device is re-configured on wakeup when only system power management is used
- Use support functions as needed from DPL and DriverLib

The journey (1 of 4)

- Add RTC-based timer driver
  - Wake-up from standby mode
  - Tickless kernel
- Map sleep states to SoC's supported power modes and implement sys_set_power_state()
  - Sleep 1: IDLE mode
  - Sleep 2: STANDBY mode
  - Deep Sleep 1: SHUTDOWN mode
- Initialize TI Power Manager during system initialization (SYS_INIT)

The journey (2 of 4)

- Create customized policy function
  - Residency-based
  - Only enters sleep states
  - Check any power constraints set by TI drivers
  - Schedule a k_timer to wake up slightly ahead of deadline to account for any latency

The journey (3 of 4)

- Implement power management in drivers (GPIO, UART, I2C, SPI)
  - Define list of “core devices” in subsys/power/device.c (timer and UART 0)
  - Enable all pins as wakeup source in GPIO driver
  - Implement <driver>_pm_control()
    - Handle low-power, suspend and off states in the same way
  - Lock sleep 2 (STANDBY mode) where appropriate
    - UART driver API is not very power-friendly. Difficult to lock out standby mode in driver when using uart_poll_in (Issue #23798)
  - System PM on, Device PM off: use TI Power Manager for notifications

The journey (4 of 4)

- Skipped Device Busy Indication feature
  - Used to indicate a device is busy
  - System power state locking is sufficient, and has better granularity
  - Busy status not being checked in Zephyr
- Tried Device Idle Power Management
  - Add ref count to each device, adjust count based on need, turn off when count is 0.
  - Does not work very well with centralized PM method (Issue #22391)
  - No concept of power domains

Power Measurements

Scenario	Zephyr (mA)	TI-RTOS (mA)
Sleep forever[1] (STANDBY)	0.001	0.001
Printing hello world once a second[1]	0.012	0.007
SPI loopback test repeated every second	0.137	n/a
Bme280 sample using boostxl_sensors	0.083	n/a
Deep sleep forever[1] (SHUTDOWN)	0.00019	0.00018

[1]With external flash powered off

- Measured with stune utility from TI EnergyTrace™ in CCS 10.0
  - ±0.5 uA nominal
  - Range: 1 uA to 100 mA
  - 256 ksps
  - Averaged over 10 sec
- Exception: deep sleep measured using a multimeter

Zephyr taken from
Commit id: dcde0d5ed5283b4eeba6e5651d7ee6570078f3bd
On https://github.com/vanti/zephyr/tree/demo

EnergyTrace™ Demo

CC1352R1 LaunchPad connected to Windows PC

- *Stune*: Command line utility from TI Code Composer Studio IDE

- *EnergyPy*: Python wrappers to invoke stune.exe and xdsdfu.exe

Summary

- Zephyr provides a basic power management framework
- With help from TI Power Manager, I was able to get decent results in static scenarios
- Basic system PM and device PM support on CC13x2 merged
- Shutdown mode support pending review
- PM subsystem would benefit from more features and better integration

Thank you

Accelerating deployment in the Arm Ecosystem

Linaro

TECH DAYS