TIMERS

There are multiple clock sources

1-Precision Internal Oscillator (PIOSC): by default after reset //pll

- It operates without external components.
- Provides a clock frequency of **16 MHz ±1%** at room temperature.
- Accuracy varies to **±3% across temperature ranges**.

2- Main Oscillator (MOSC) : can configured to 2 //pll

- **External Single-Ended Clock Source**: An external clock signal can be connected to the OSC0 input pin. The acceptable frequency range for this configuration is specified in the microcontroller's datasheet.
- **External Crystal Oscillator**: A crystal can be connected across the OSC0 (input) and OSC1 (output) pins. When using the Phase-Locked Loop (PLL), the crystal frequency must be between 5 MHz and 25 MHz (inclusive). Without the PLL, the crystal frequency can range from 4 MHz to 25 MHz. Supported crystal frequencies are listed in the XTAL bit field of the Run-Mode Clock Configuration (RCC) register.

3- Low-Frequency Internal Oscillator(LFIOSC)

- designed for use during Deep-Sleep power-saving modes. It operates at a typical frequency of 30 kHz

4- Hibernation Module Clock Source :

- An **external crystal oscillator** provide Precisely **32.768 kHz**
- **High accuracy** and stability, ideal for real-time clock (RTC) applications.

MODES

1. **Run Mode**: Full performance, highest power consumption.
2. **Sleep Mode**: CPU off, peripherals active, lower power.
3. **Deep-Sleep Mode**: CPU and most peripherals off, very low power.
4. **Hibernate Mode**: Almost everything off, only essential state kept, lowest power.

General-Purpose Timer Module (GPTM) provides versatile timer functionalities for counting or timing external events. Here's a summary of its key features:

Key Features:

1. **Timer Blocks**:
    - **Six 16/32-bit GPTM blocks** and **six 32/64-bit Wide GPTM blocks**.
    - Each 16/32-bit block provides two timers (Timer A and Timer B), which can be used independently or concatenated as a 32-bit timer.
2. **Operating Modes**:
    - **16/32-bit Modes**:
        - One-shot timer
        - Periodic timer
        - Real-Time Clock (RTC) using a 32.768 kHz external clock
        - Event counters or edge-counting modes
        - PWM output with software-programmable inversion
    - **32/64-bit Modes**: wide blocks
        - One-shot timer (32-bit or 64-bit)
        - Periodic timer (32-bit or 64-bit)
        - RTC using external 32.768 kHz clock
        - Edge counting or time capture with a prescaler
        - PWM with programmable output inversion
3. **Timer Modes**:
    - **Count Up/Down**: Timers can count up or down.
4. **ADC Triggering**:
    - Timers can trigger **ADC conversions** (in periodic and one-shot modes).
    - ADC trigger signals from all timers are **ORed together**, meaning only one timer should be used to trigger ADC events.

Steps to configure:

1. Disable the Timer and configure clock

- Ensure the timer is **disabled** by clearing the **TnEN** bit in the **GPTMCTL** register.
- RCGCTIMER or RCGCWTIMER // one wide 6 pins each for the whole clk

2. Configure the GPTM

- Write **0x0000.0000** to the **GPTMCFG** register to configure the timer block.

3. Set the Timer Mode

- Write the desired value to the **GPTMTnMR** register:
    - **0x1** for **One-Shot mode**.
    - **0x2** for **Periodic mode**.

5. Load the Start Value

- Write the **start value** into the **GPTM Timer n Interval Load Register (GPTMTnILR)**.

6. Enable Interrupts (Optional)

- If **interrupts** are required, enable them by setting the corresponding bits in the **GPTM Interrupt Mask Register (GPTMIMR)**.

7. Enable the Timer

- Set the **TnEN** bit in the **GPTMCTL** register to **enable** the timer and start counting.

8. Wait for Time-out or Interrupt

- Poll the **GPTMRIS** register or wait for the interrupt to be triggered (if enabled).
- Clear the status flags by writing **1** to the corresponding bit in the **GPTM Interrupt Clear Register (GPTMICR)**.
- In **One-Shot mode**, the timer stops counting after the time-out event, and you need to restart it.
- In **Periodic mode**, the timer reloads automatically and continues counting after the time-out event.

GPTMCTL control→ en/dis (A or B)

GPTMCFG configuration→ 16/32 or 32/64

GPTMTnMR mode register→ sets the mode (periodic , …) or (count up or down)

GPTMICR clear register→ clear after interrupt

GPTMRIS raw → to see if interrupt happened regardless en/dis

GPTMTnILR interval load→ load the value

GPTMIMR interrupt mask → en/dis interrupt

GPTMMIS masked interrupt→ see only statues of masked

GPTMTnAPR prescalar register→ set prescalar