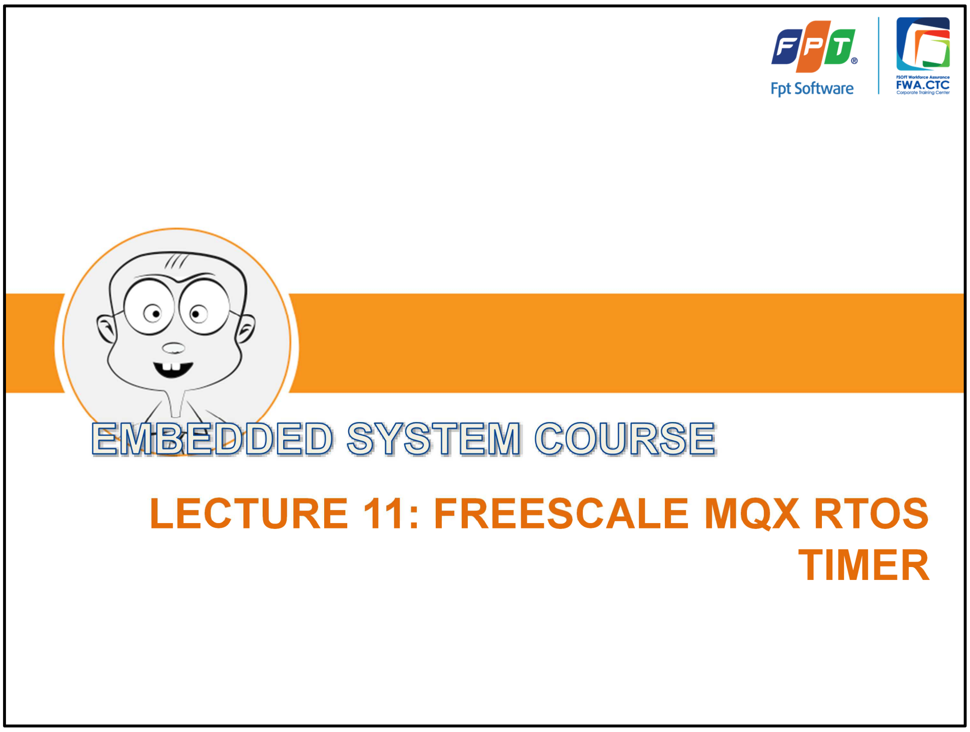
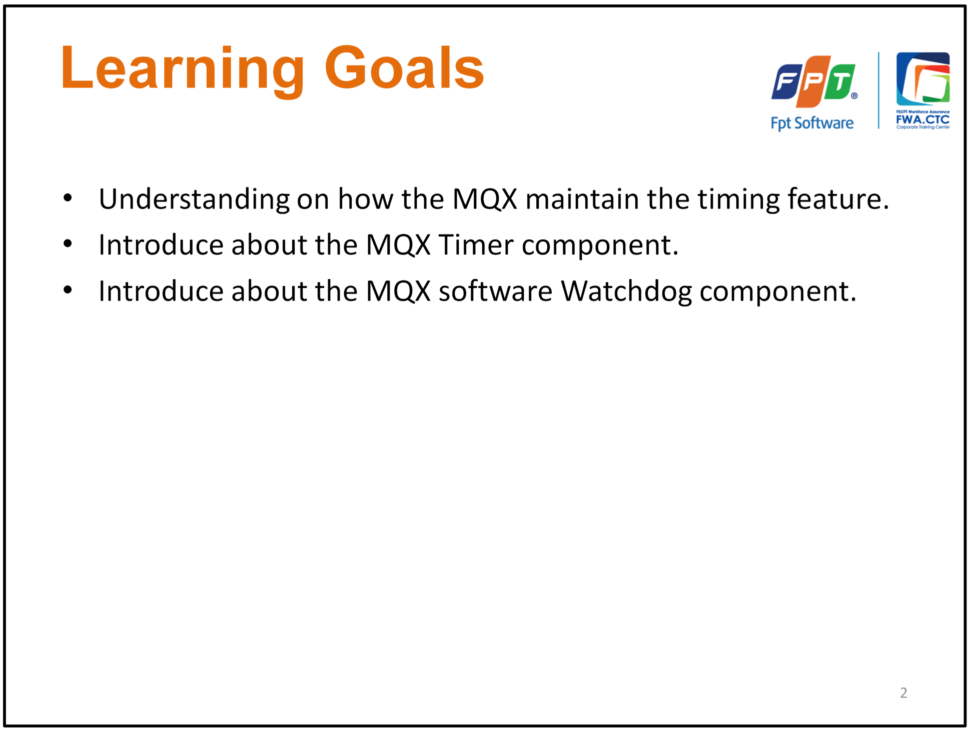


EMBEDDED SYSTEM COURSE

LECTURE 11: FREESCALE MQX RTOS TIMER



Learning Goals

- Understanding on how the MQX maintain the timing feature.
- Introduce about the MQX Timer component.
- Introduce about the MQX software Watchdog component.

2

❖ MQX Timing features

❖ Deep look at how kernel handles time

❖ Absolute versus Relative Time

❖ Timer Component

❖ Watchdog

❖ Timer Exercise

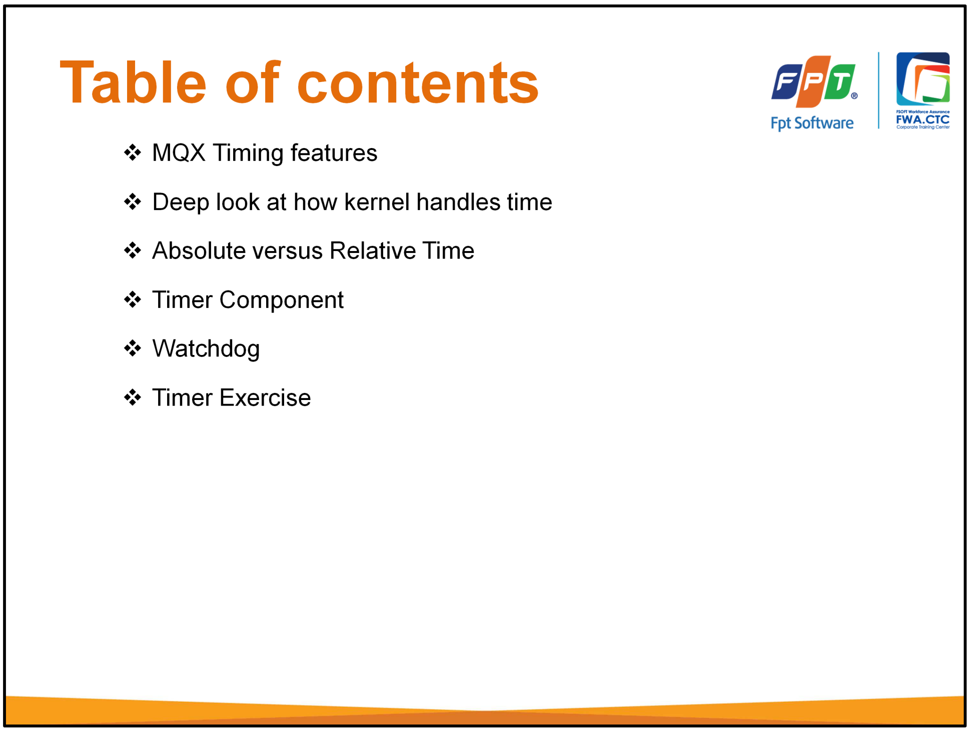


Table of contents

- MQX Timing features
- Deep look at how kernel handles time
- Absolute versus Relative Time
- Timer Component
- Watchdog
- Timer Exercise

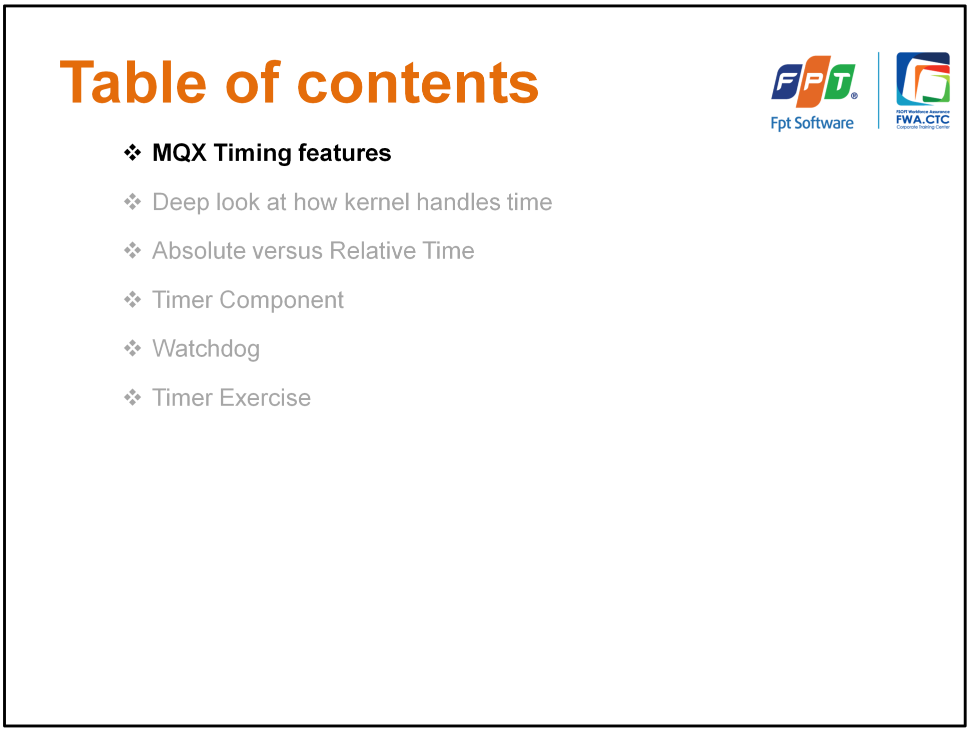
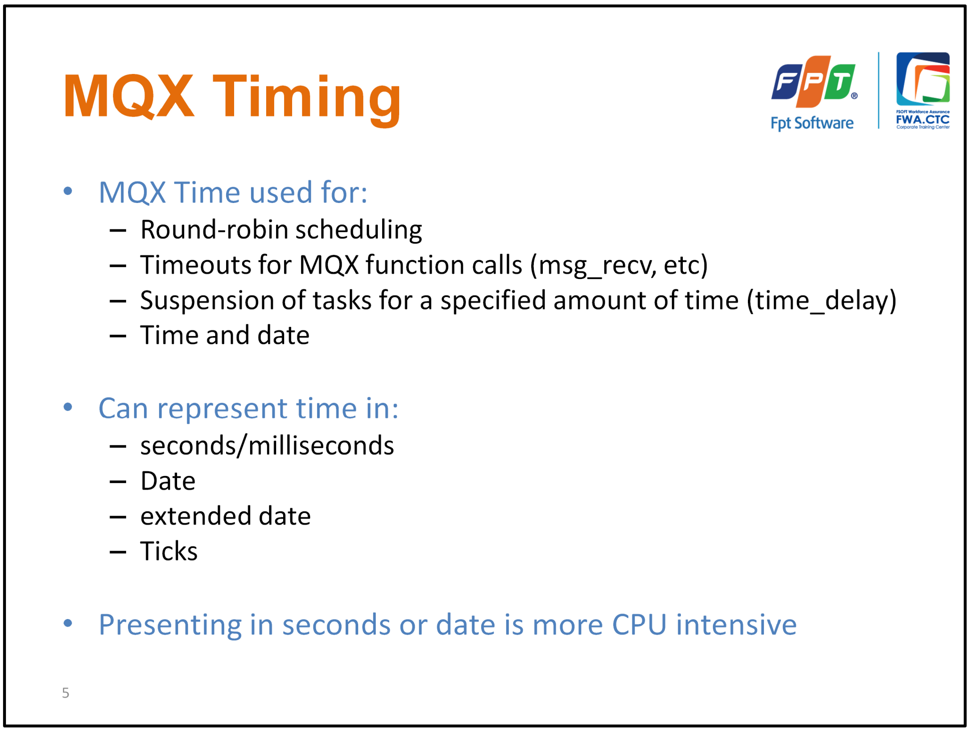


Table of contents

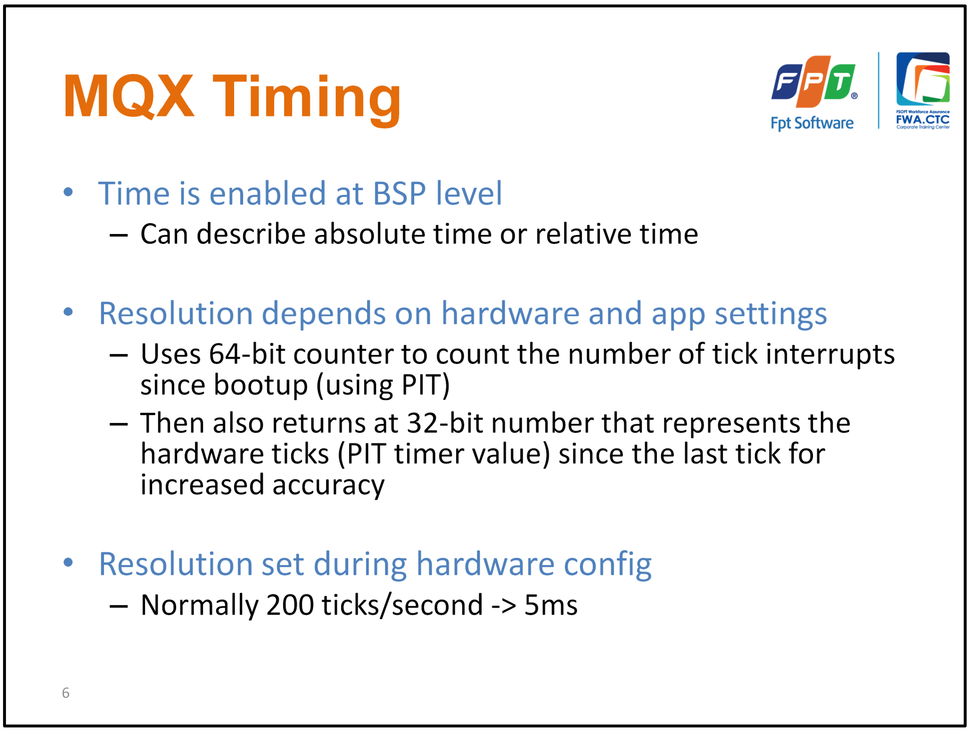
- ❖ **MQX Timing features**
- ❖ Deep look at how kernel handles time
- ❖ Absolute versus Relative Time
- ❖ Timer Component
- ❖ Watchdog
- ❖ Timer Exercise



MQX Timing

- MQX Time used for:
  - Round-robin scheduling
  - Timeouts for MQX function calls (msg_recv, etc)
  - Suspension of tasks for a specified amount of time (time_delay)
  - Time and date
- Can represent time in:
  - seconds/milliseconds
  - Date
  - extended date
  - Ticks
- Presenting in seconds or date is more CPU intensive

Time_delay not exact, just guarentee the minimum amount of delay. Ie if say delay 7ms, will delay for 10ms since tha’ts smaller than BSP resolution



MQX Timing

- Time is enabled at BSP level
  - Can describe absolute time or relative time
- Resolution depends on hardware and app settings
  - Uses 64-bit counter to count the number of tick interrupts since bootup (using PIT)
  - Then also returns at 32-bit number that represents the hardware ticks (PIT timer value) since the last tick for increased accuracy
- Resolution set during hardware config
  - Normally 200 ticks/second -> 5ms

Draw how this works on a board

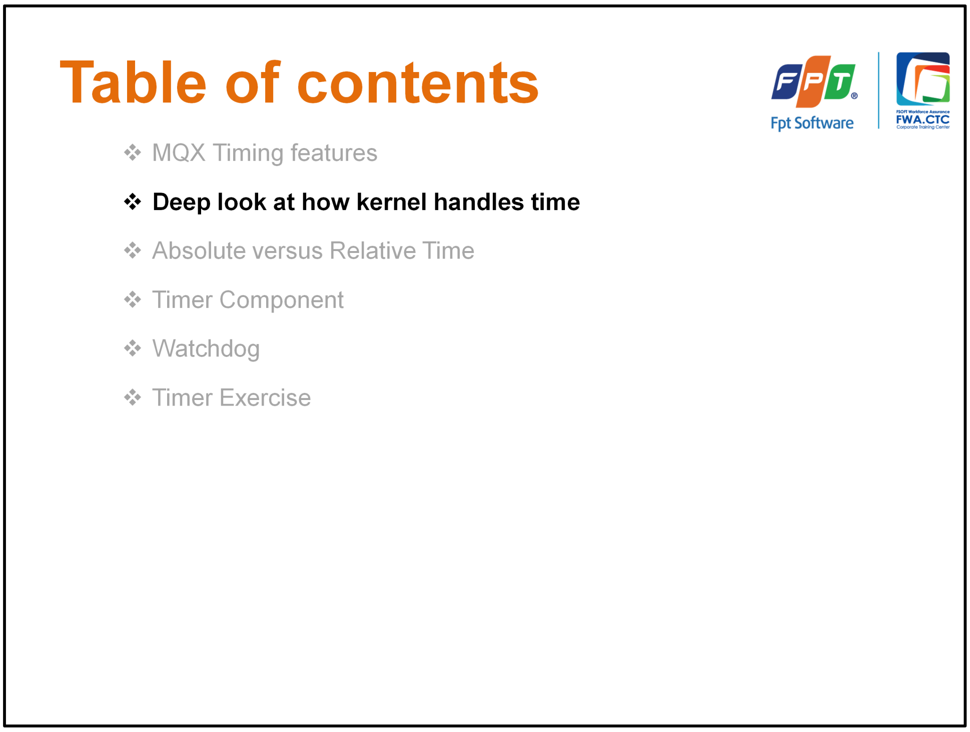
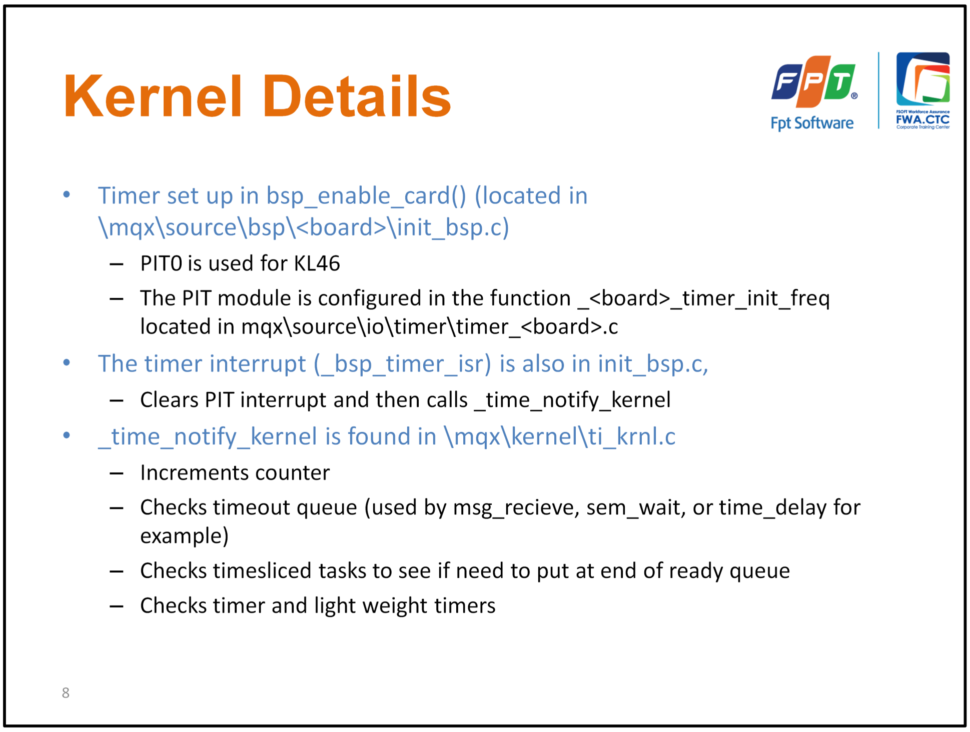


Table of contents

- MQX Timing features
- **Deep look at how kernel handles time**
- Absolute versus Relative Time
- Timer Component
- Watchdog
- Timer Exercise



Kernel Details

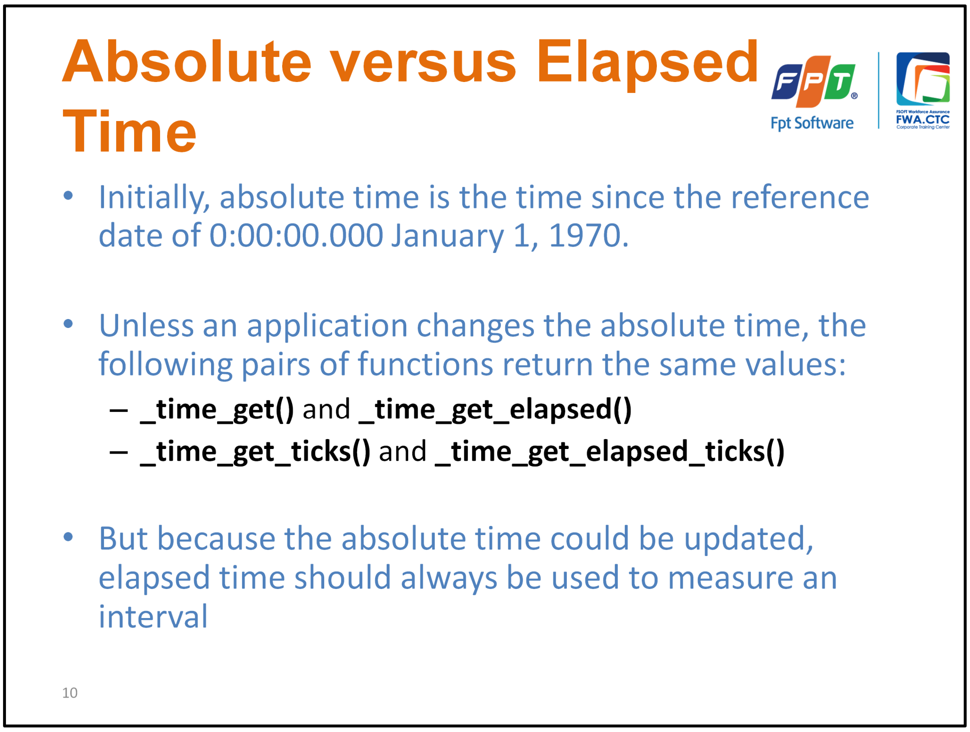
- Timer set up in bsp_enable_card() (located in \mqx\source\bsp\<board>\init_bsp.c)
  - PIT0 is used for KL46
  - The PIT module is configured in the function _<board>_timer_init_freq located in mqx\source\io\timer\timer_<board>.c
- The timer interrupt (_bsp_timer_isr) is also in init_bsp.c,
  - Clears PIT interrupt and then calls _time_notify_kernel
- _time_notify_kernel is found in \mqx\kernel\ti_krnl.c
  - Increments counter
  - Checks timeout queue (used by msg_recieve, sem_wait, or time_delay for example)
  - Checks timesliced tasks to see if need to put at end of ready queue
  - Checks timer and light weight timers

8

And remember because an interrupt happened, it will re-look at the scheduling options, so if a higher priority task is found to be in the ready queue now, then it will run instead

Table of contents

- MQX Timing features
- Deep look at how kernel handles time
- **Absolute versus Relative Time**
- Timer Component
- Watchdog
- Timer Exercise



Absolute versus Elapsed Time

Fpt Software

FWA.CTC

- Initially, absolute time is the time since the reference date of 0:00:00.000 January 1, 1970.
- Unless an application changes the absolute time, the following pairs of functions return the same values:
  - **_time_get()** and **_time_get_elapsed()**
  - **_time_get_ticks()** and **_time_get_elapsed_ticks()**
- But because the absolute time could be updated, elapsed time should always be used to measure an interval

Many ways to see the time and many time functions. See MQXUG or MQXRM for the details

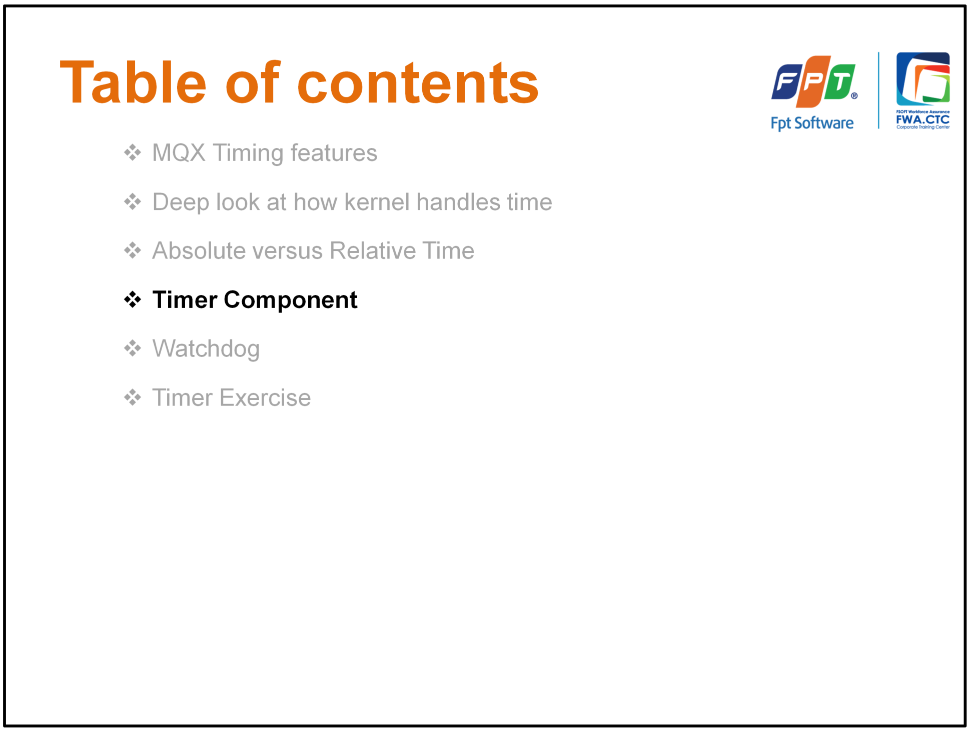
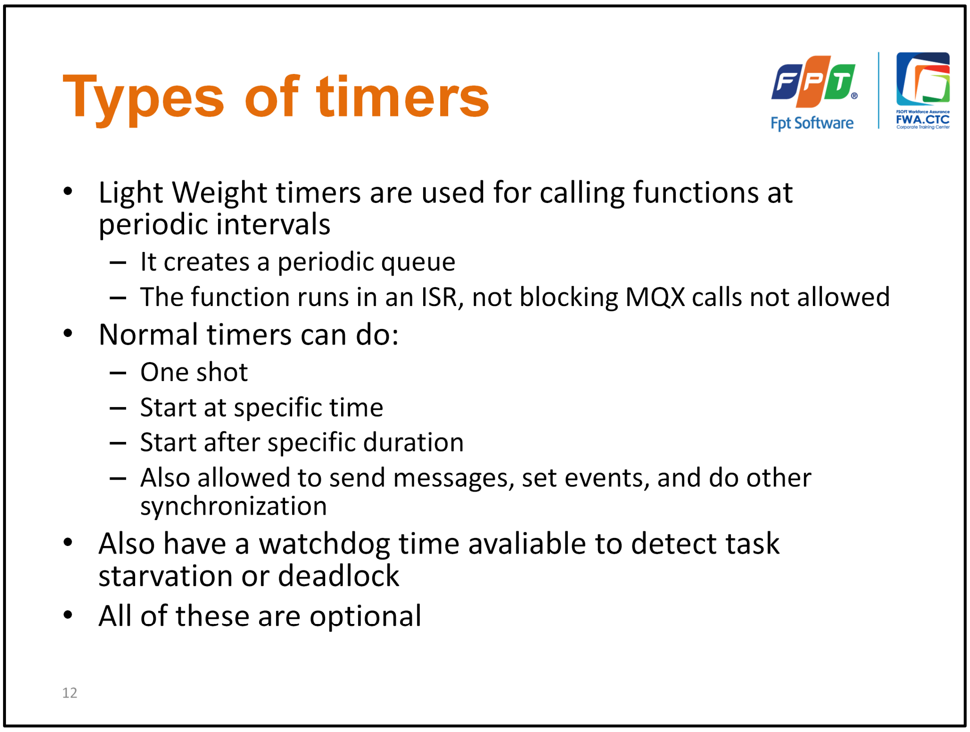


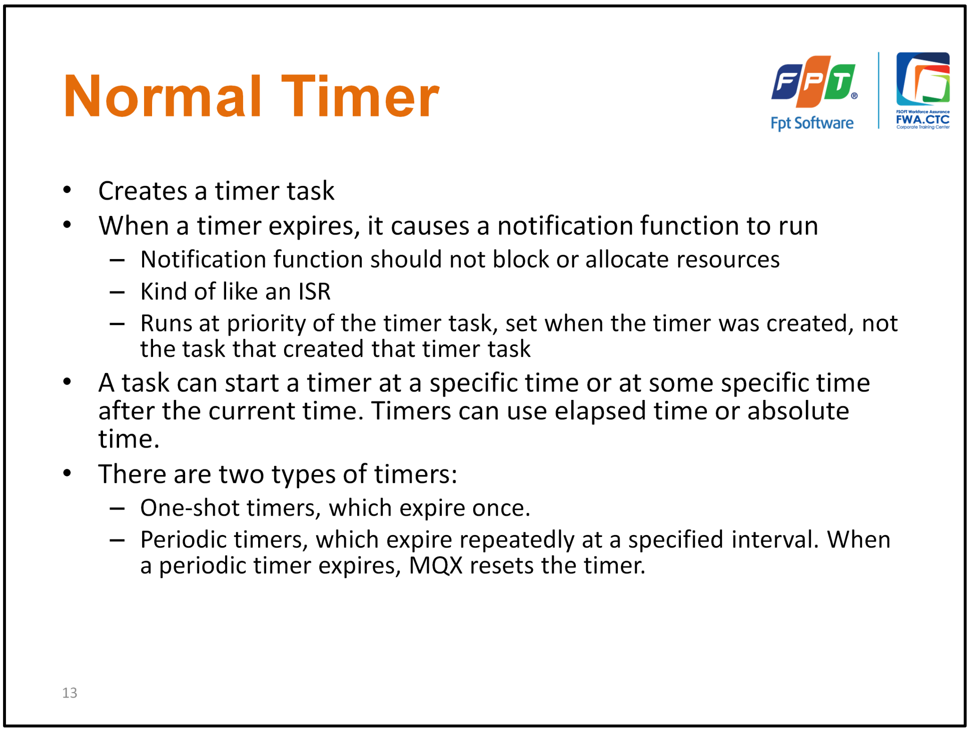
Table of contents

- ❖ MQX Timing features
- ❖ Deep look at how kernel handles time
- ❖ Absolute versus Relative Time
- ❖ **Timer Component**
- ❖ Watchdog
- ❖ Timer Exercise



Types of timers

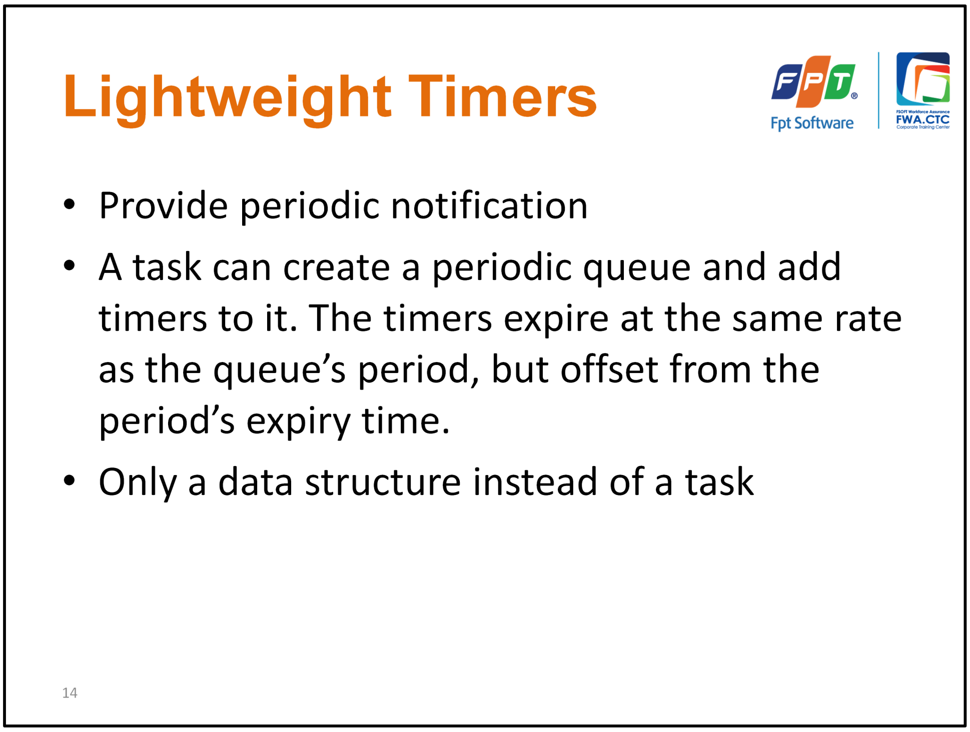
- Light Weight timers are used for calling functions at periodic intervals
  - It creates a periodic queue
  - The function runs in an ISR, not blocking MQX calls not allowed
- Normal timers can do:
  - One shot
  - Start at specific time
  - Start after specific duration
  - Also allowed to send messages, set events, and do other synchronization
- Also have a watchdog time avaliable to detect task starvation or deadlock
- All of these are optional



Normal Timer

- Creates a timer task
- When a timer expires, it causes a notification function to run
  - Notification function should not block or allocate resources
  - Kind of like an ISR
  - Runs at priority of the timer task, set when the timer was created, not the task that created that timer task
- A task can start a timer at a specific time or at some specific time after the current time. Timers can use elapsed time or absolute time.
- There are two types of timers:
  - One-shot timers, which expire once.
  - Periodic timers, which expire repeatedly at a specified interval. When a periodic timer expires, MQX resets the timer.

See MQXUG and MQXRM for API details



Lightweight Timers

- Provide periodic notification
- A task can create a periodic queue and add timers to it. The timers expire at the same rate as the queue's period, but offset from the period's expiry time.
- Only a data structure instead of a task

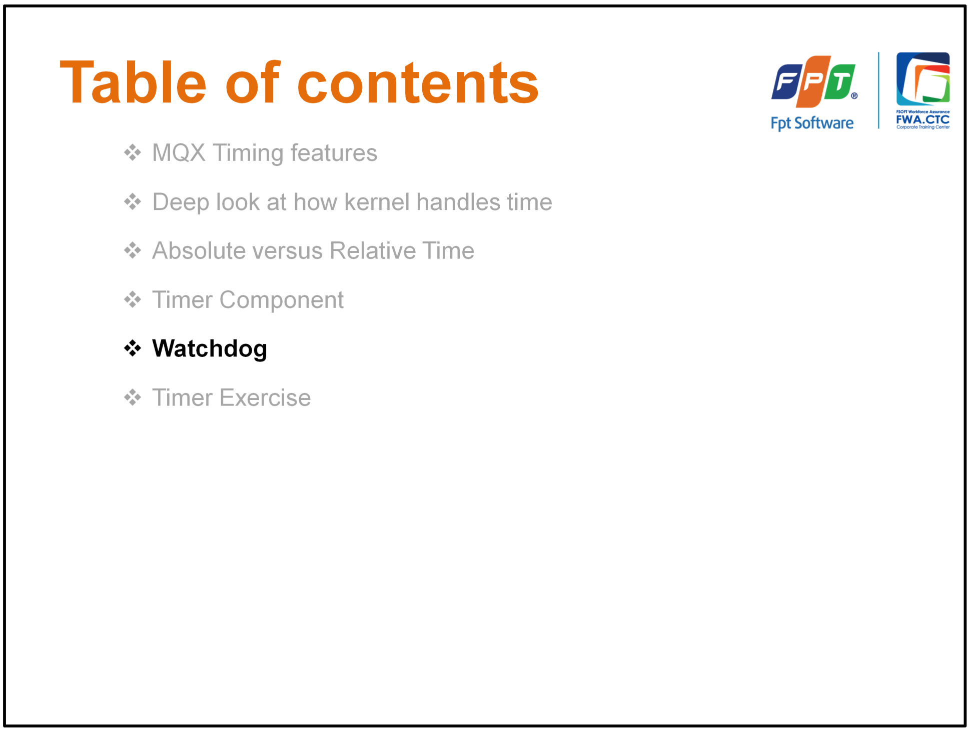
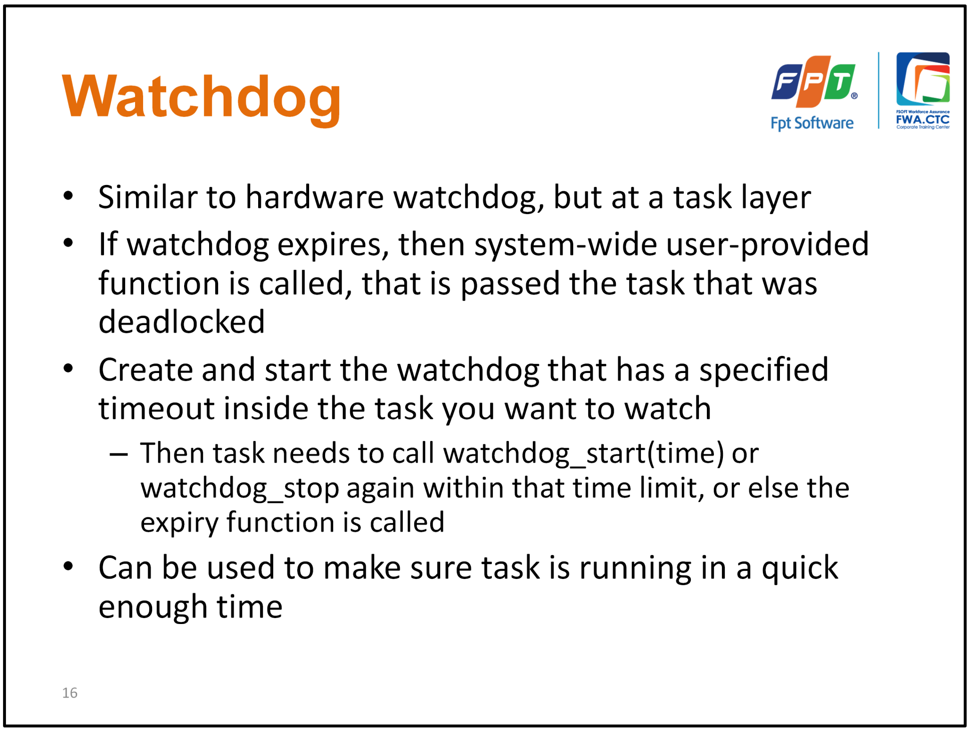


Table of contents

- ❖ MQX Timing features
- ❖ Deep look at how kernel handles time
- ❖ Absolute versus Relative Time
- ❖ Timer Component
- ❖ **Watchdog**
- ❖ Timer Exercise



Watchdog

- Similar to hardware watchdog, but at a task layer
- If watchdog expires, then system-wide user-provided function is called, that is passed the task that was deadlocked
- Create and start the watchdog that has a specified timeout inside the task you want to watch
  - Then task needs to call watchdog_start(time) or watchdog_stop again within that time limit, or else the expiry function is called
- Can be used to make sure task is running in a quick enough time

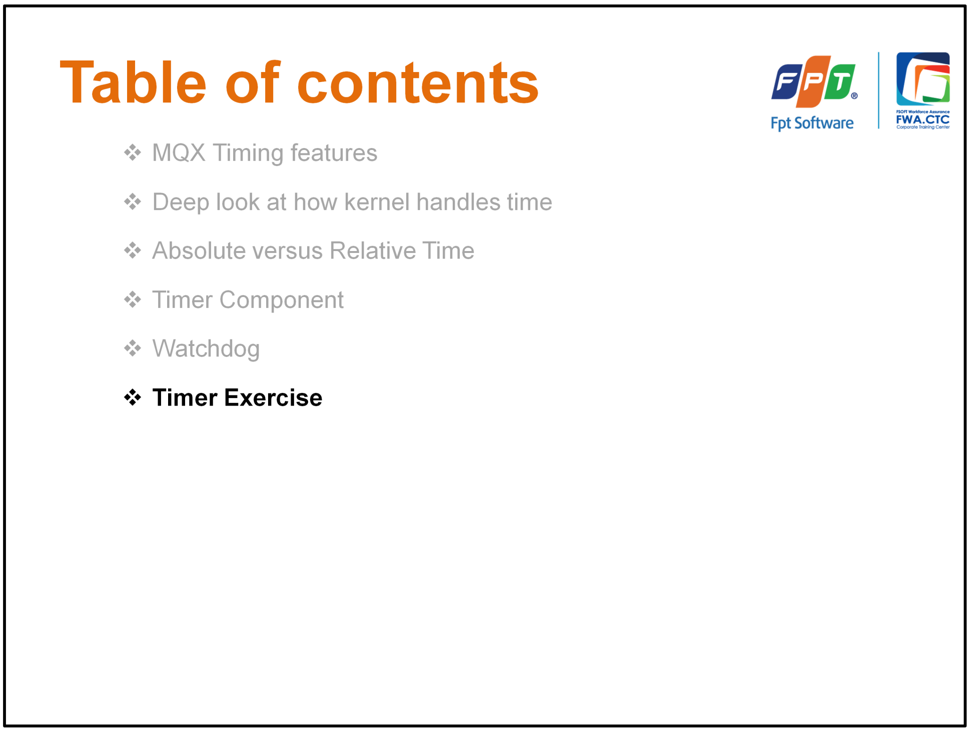
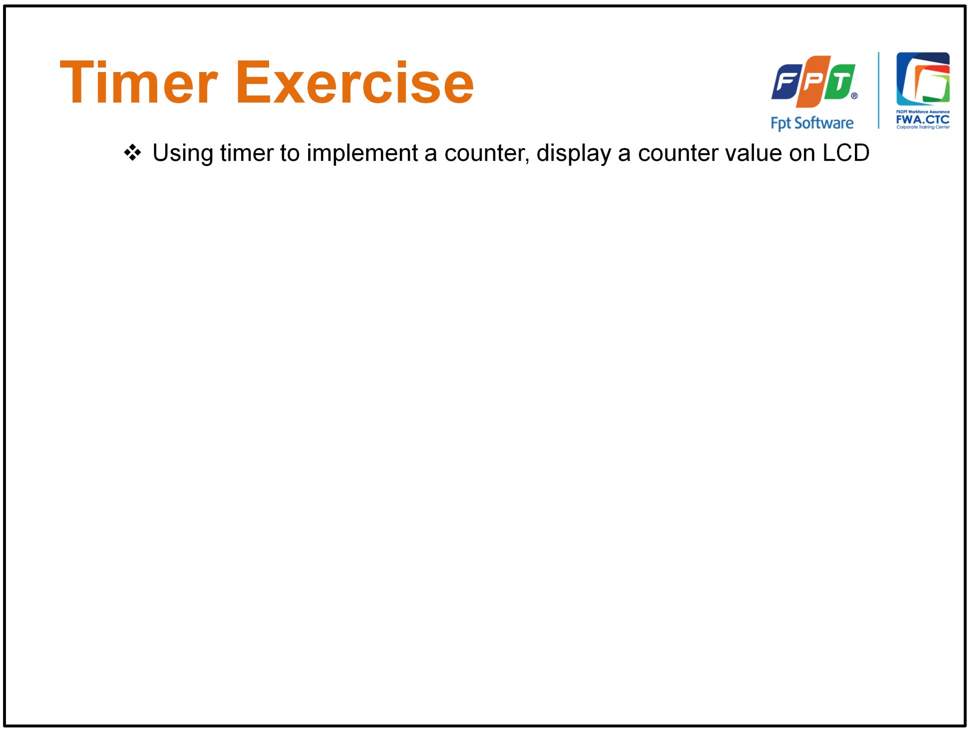


Table of contents

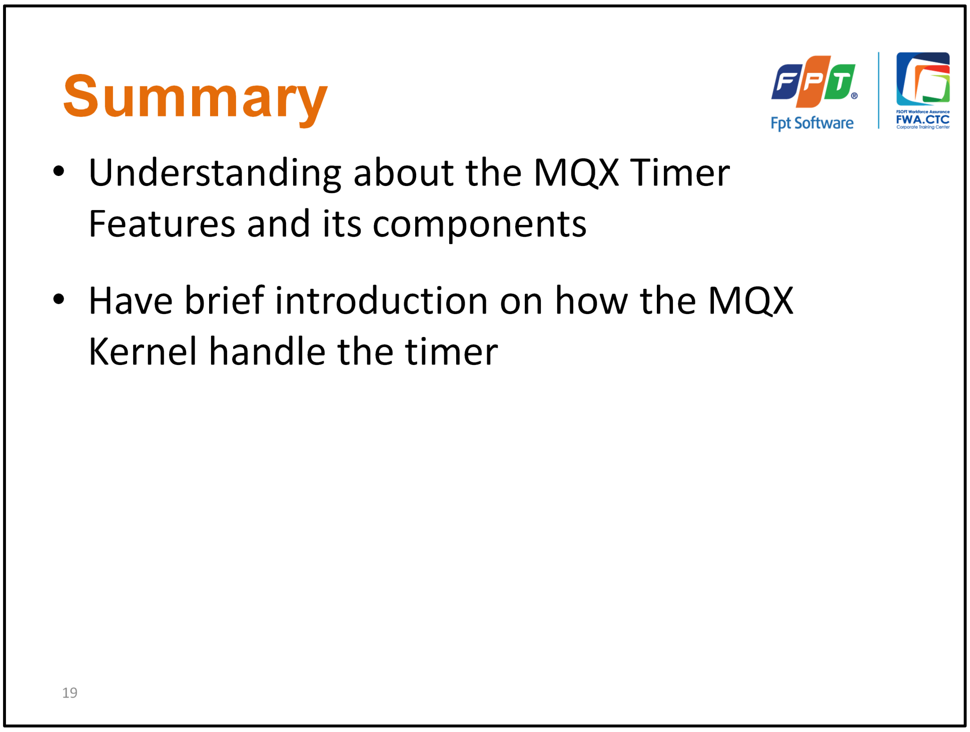
- MQX Timing features
- Deep look at how kernel handles time
- Absolute versus Relative Time
- Timer Component
- Watchdog
- **Timer Exercise**



Timer Exercise

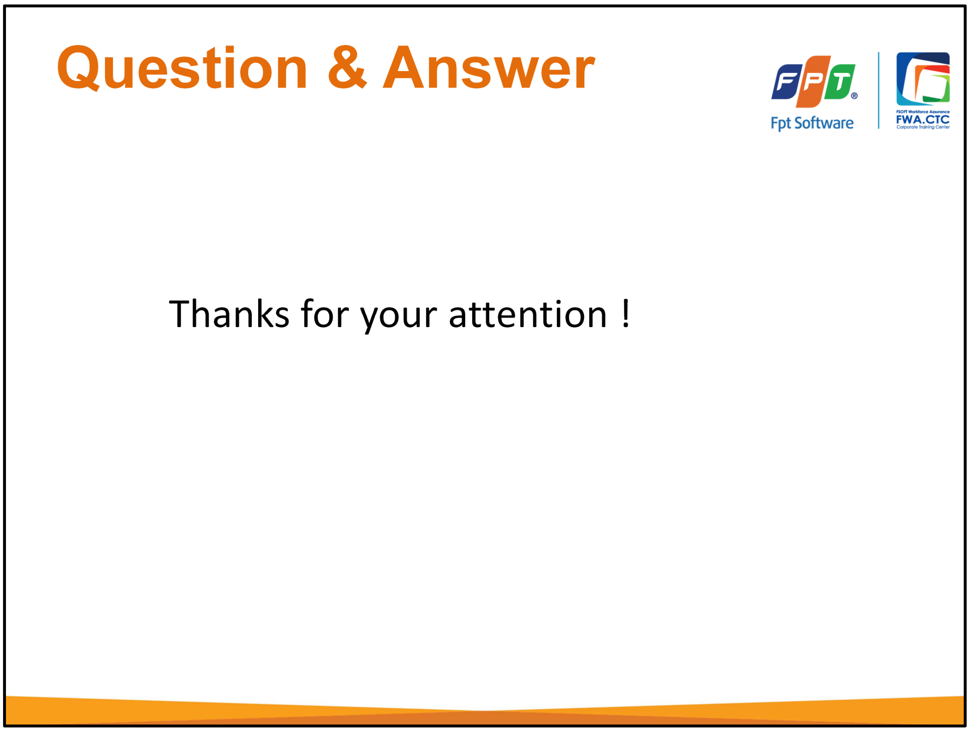
Fpt Software
FWA.CTC

- ❖ Using timer to implement a counter, display a counter value on LCD



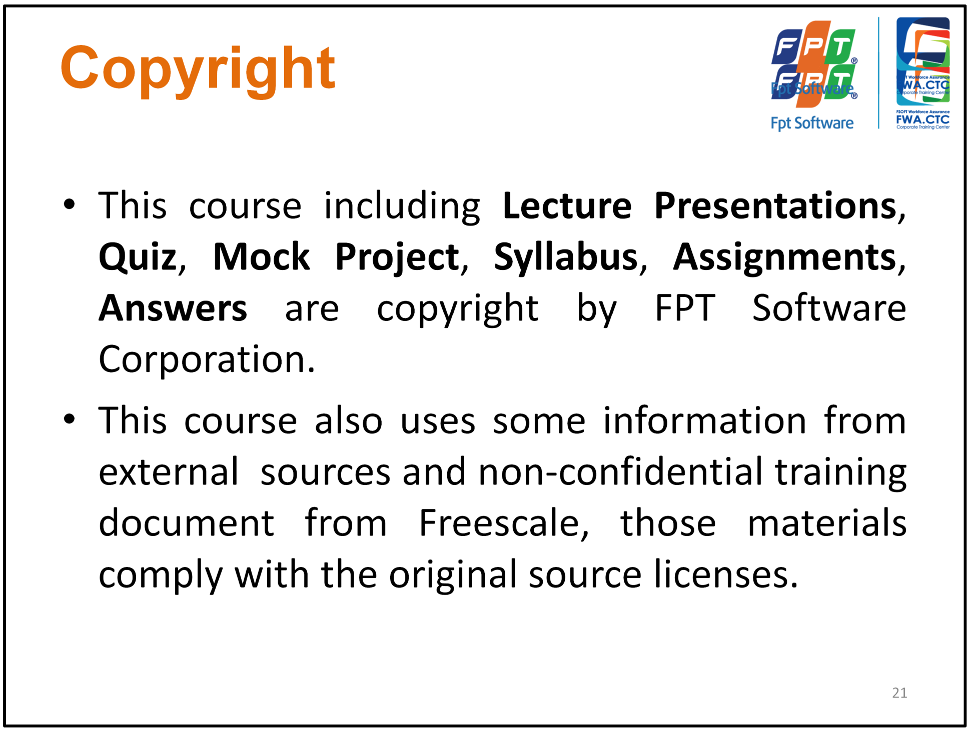
Summary

- Understanding about the MQX Timer Features and its components
- Have brief introduction on how the MQX Kernel handle the timer



Question & Answer

Thanks for your attention !



Copyright