Freescale MQX RTOS Example Guide

Low power example

This document explains the Low power example, what to expect from the example and a brief introduction to the API used.

The example

The example shows switching between several predefined low power operation-modes. The core and serial driver are being notified and their settings are updated based on new operation mode transparently for the application. The switch is initiated by the on-board button press. In some low power operation modes, the application should be woken up by a hardware event like timer wakeup interrupt or a Serial line RX interrupt. The immediate settings are being displayed on terminal and they are signaled by different LED patterns. The example also demonstrates low power feature of core sleep in idle task.

Running the example

The Low power application belongs to the set of examples of MQX low power support. The MQX_ENABLE_LOW_POWER macro must be set to non-zero in the user_config.h file prior to compilation of MQX kernel libraries and the example itself.

To run the example the corresponding IDE, compiler, debugger and a terminal program are needed.

Explaining the example

The application example creates two tasks:

- **main:** task sets up the BUTTON and LED1 pins and installs the timer wakeup interrupt. Any time the button is pressed this task toggles the LED1 and it changes the operation mode. In some modes, the timer wakeup interrupt or SCI RX interrupt needs to occur to wake up the core. The flow of main task is shown on the diagram below.

- **for_loop_led_task:** sets up the LED2 and toggles its state with a frequency given by current clock configuration. You will see a change of the clock configuration as a change of LED2 blink rate. When the LED2 is not blinking the core is in the low power stop mode and is not executing any task.

The LWGPIO driver is used for the LED and BUTTON pins handling. The button_led_init() function initializes the corresponding pins for input and output and also it installs the falling-edge interrupt for the button (with internal pull-up resistor enabled). The ISR handler toggles the LED2 and sets the event to trigger an action in the main task.

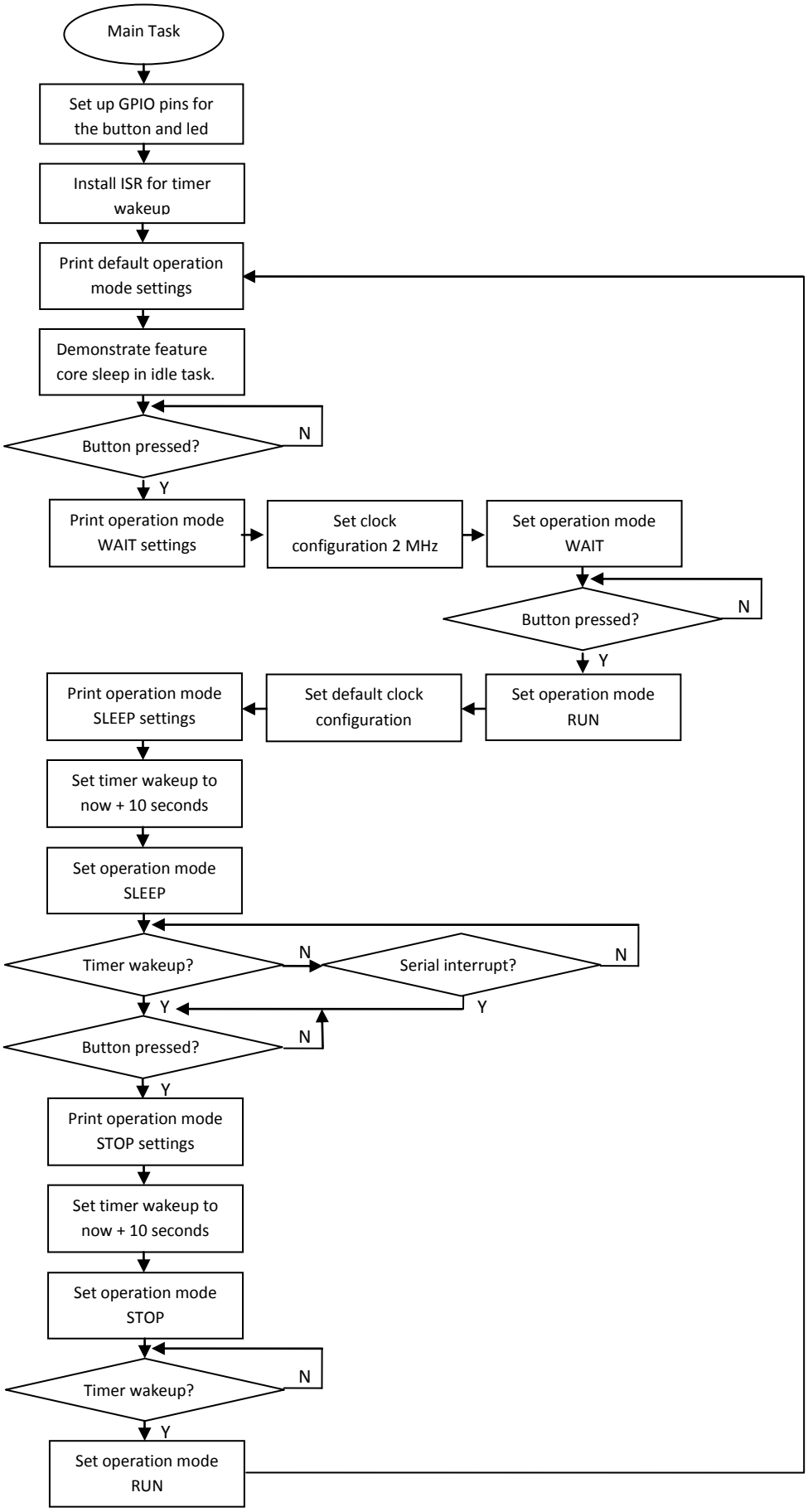
The operation modes are set using the Low-power Manager API function _lpm_set_operation_mode(). The parameter of this function is one of the LPM constants that correspond to predefined operation modes.

Note: even after a core wakeup event, the current operation mode stays unchanged until the next call to the _lpm_set_operation_mode() function.

The application uses the SCI and timer interrupts as events to wake-up the core from some low power operation modes. For the serial line, the interrupt is available only if an interrupt-driven driver is used "**i**ttyX:". The timer wakeup is always set to time "now + 10 seconds". The timer ISR itself wakes up the core and the _lpm_wakeup_core() function is used here to let the core to run also after the ISR finishes. Refer to Kinetis documentation for more information about specific CPU behavior in various low power modes.

Also note that the operation mode LPM_OPERATION_MODE_WAIT requires the clock to be switched to 2MHz first. This is done using the LPM API function _lpm_set_clock_configuration(). The original clock configuration is restored after the operation mode is changed to default again to avoid unsupported core configuration.

The low power feature of core sleep in idle task is demonstrated in LPM_OPERATION_MODE_RUN mode. The feature is enabled/disabled using function _lpm_idle_sleep_setup(). It can be turned on in all operation modes but it actually takes place only when it affects just the core behavior. The example shows that the number of loops in idle task gets minimized when this feature is enabled, because the idle task puts the core into sleep between interrupts (system tick).


Main Task
Set up GPIO pins for the button and led
Install ISR for timer wakeup
Print default operation mode settings
Demonstrate feature core sleep in idle task.
Button pressed?
N
Y
Print operation mode WAIT settings
Set clock configuration 2 MHz
Set operation mode WAIT
Button pressed?
N
Y
Set operation mode RUN
Set default clock configuration
Print operation mode SLEEP settings
Set timer wakeup to now + 10 seconds
Set operation mode SLEEP
Timer wakeup?
N
Serial interrupt?
N
Y
Y
Button pressed?
N
Y
Print operation mode STOP settings
Set timer wakeup to now + 10 seconds
Set operation mode STOP
Timer wakeup?
N
Y
Set operation mode RUN