Freescale MQX RTOS Example Guide

Lowpower_vybrid example

This document explains the Lowpower_vybrid 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 stop mode the application is woken up by pressing another button. The immediate settings are being displayed on terminal and they are signaled by different LED patterns.

Notes:
1. Currently the example can only run with internal RAM version.
2. For M4 project a fused M4-primary CPU must be used.
3. For AUTOEVB board, connect P20[1..4] to P17[1..4] to make LEDs work.

In the following descriptions, INT_BUTTON and WAKEUP_BUTTON refers to different SW buttons on Tower Board and AutoEVB Board

	TOWER	AUTOEVB
INT_BUTTON	SW2	SW6
WAKEUP_BUTTON	SW1	SW9

Running the example

The Lowpower_vybrid 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 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 only one task called main_task which initializes the buttons for changing power operation mode and waking up the MCU from stop mode. Main_task also configures the LEDs to alert the user at the clock change via the blinking rate of LEDs. After initializing buttons and LEDs the main_task enters an endless loop of power mode switching.

- The default operation mode – RUN mode - and the CPU operation mode are displayed.
- The LED4, LED3, LED2 are blinking at high rate, LED1 is off.

```
MQX Power Modes Switch Demo
!!!CAUSION!!! This example only support RAM version
!!!CAUSION!!! Please select Int Ram version configuration to build and run

********************************************************************************
************** Operation mode : LPM_OPERATION_MODE_RUN ***********************
********************************************************************************
Core Low Power Settings:
  Mode:      LPM_CPU_POWER_MODE_RUN
Info: RUN operation mode is mapped on RUN power mode by default.
      The core runs at full clock speed.
      LEDs blinks quickly
Press SW2 to exit LED blink loop...OK
```

- Button SW2 must be pressed to switch from RUN mode to WAIT mode.
- Before entering WAIT mode the new clock configuration must be set by calling _lpm_set_clock_configuration() and then it calls function _lpm_set_operation_mode() to switch the CPU into LPR mode which corresponds to WAIT operation mode.
- The LED4, LED3, LED2 are blinking at rate slower than in RUN mode, LED1 is off.

```
********************************************************************************
************** Operation mode : LPM_OPERATION_MODE_WAIT ***********************
********************************************************************************
Core Low Power Settings:
  Mode:      LPM_CPU_POWER_MODE_LPR
Info: WAIT operation mode is mapped on VLPR power mode by default.
      PLL are disabled, core runs with FXOSC (32MHz).
      LED2 blinks slowly

Setting operation mode to LPM_OPERATION_MODE_WAIT ... OK
Press SW2 to exit LED blink loop...OK
```

- As SW2 is pressed the operation mode is switched to SLEEP mode while the CPU enters WAIT mode and only wakes up as SW1 is pressed. The CPU returns to LPR mode as it wakes up and LEDs continues to blink at slow rate.

```
********************************************************************************
************** Operation mode : LPM_OPERATION_MODE_SLEEP **********************
********************************************************************************
Core Low Power Settings:
  Mode:      LPM_CPU_POWER_MODE_WAIT
Info: SLEEP operation mode is mapped on WAIT power mode by default.
      The core is inactive in this mode, reacting only to wakeup interrupts.
      LEDs don't blink

Setting operation mode to LPM_OPERATION_MODE_SLEEP, Press SW1 to wakeup... OK
Press SW2 to exit LED blink loop...OK
```

- SW2 is pressed again which allows the CPU to enter STOP mode and it returns to LPR mode as SW1 is pressed. The LEDs continues blinking at the same rate as in SLEPP operation mode.

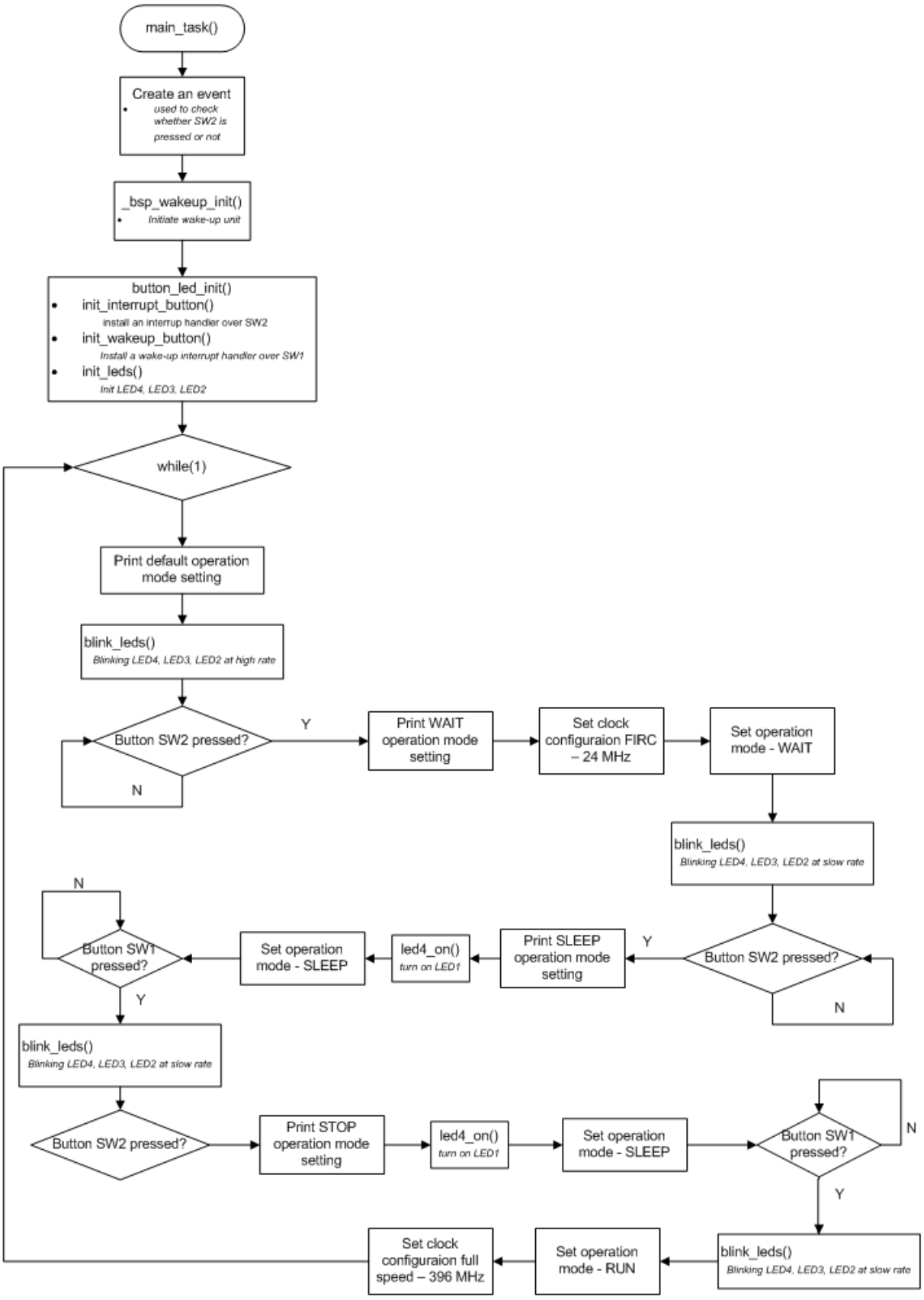
```
*********************************************************************************
********** Operation mode : LPM_OPERATION_MODE_STOP **********
*********************************************************************************
Core Low Power Settings:
  Mode:       LPM_CPU_POWER_MODE_STOP
Info: STOP operation mode is mapped on STOP power mode by default.
      The core is inactive in this mode, reacting only to wakeup interrupts.
      LEDs don't blink

Setting operation mode to LPM_OPERATION_MODE_STOP, Press SW1 to wakeup... OK
Press SW2 to exit LED blink loop...OK

Setting operation mode to LPM_OPERATION_MODE_RUN ... OK
```

If SW2 is pressed the default clock configuration is set and CPU enters RUN mode. The MCU continues the loop of switching power mode.

The logic flow of main_task is shown in following figure.


main_task()
Create an event
used to check whether SW2 is pressed or not
_bsp_wakeup_init()
Initiate wake-up unit
button_led_init()
init_interrupt_button()
install an interrup handler over SW2
init_wakeup_button()
Install a wake-up interrupt handler over SW1
init_leds()
Init LED4, LED3, LED2
while(1)
Print default operation mode setting
blink_leds()
Blinking LED4, LED3, LED2 at high rate
Button SW2 pressed?
Y
N
Print WAIT operation mode setting
Set clock configuraion FIRC – 24 MHz
Set operation mode - WAIT
blink_leds()
Blinking LED4, LED3, LED2 at slow rate
Button SW2 pressed?
Y
N
Print SLEEP operation mode setting
led4_on()
turn on LED1
Set operation mode - SLEEP
Button SW1 pressed?
N
Y
blink_leds()
Blinking LED4, LED3, LED2 at slow rate
Button SW2 pressed?
Print STOP operation mode setting
led4_on()
turn on LED1
Set operation mode - SLEEP
Button SW1 pressed?
N
Y
blink_leds()
Blinking LED4, LED3, LED2 at slow rate
Set operation mode - RUN
Set clock configuraion full speed – 396 MHz