

Stellaris® ARM® Cortex™ M4 Training

Low Power

Section 3: Exercises

Stellaris® ARM® Cortex™ M4 Training Low Power

Low Power Mode & Hibernation Module

Section 3 – Exercises



1



Agenda

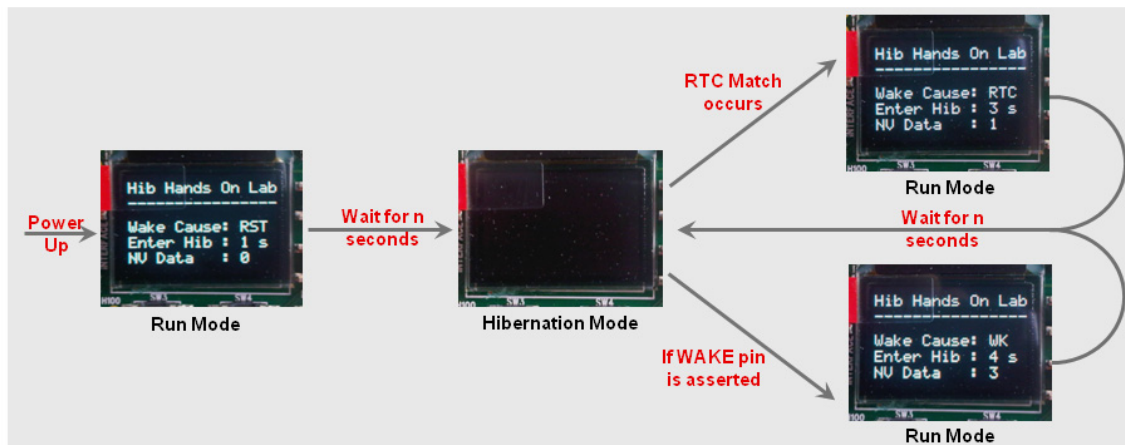
1. Exercise 1: Use RTC to wake up processor from hibernation mode	3
Objective	3
Theory	4
Code Modifications	5
Results	7
2. Exercise 2: Measure & compare current consumption in active & hibernation mode	8
Objective	8
Theory	9
Code Modifications	10
Results	12

Please note that exercise 2 is optional.

Exercise 1

Exercise 1: Objective

- Take an existing Hibernation project (from section 2). Modify it to configure Real Time Clock in order to wake-up processor from hibernation mode when RTC match occurs i.e. 5 seconds after processor goes in hibernation.
- Upon successful completion of this exercise, the application should function as follows:



Screen Shots: Exercise 1

3



Theory

In the project provided in section 2 of this workshop, the microcontroller is configured to enter in hibernation mode 'n' seconds after power up. When the microcontroller is in hibernation mode, the processor and all other peripherals are powered down to lower overall current consumption. Hibernation module remains powered by an external battery (coin cell), and wakes up the processor when WAKE pin is asserted.

In this exercise, we will modify the code from Section 2 in order to enable and configure the RTC. We will also configure hibernation module to wake the processor when RTC match occurs in addition to WAKE pin assertion.

The RTC match value will be set to 5 seconds. Therefore, the processor will wake up from the hibernation mode after 5 seconds are elapsed (while WAKE pin is not asserted) from the point it enters hibernation mode.

More Information

For more information on the Hibernation mode and RTC, please refer to the device datasheet.

Code Modifications

Please refer to file `hib_lab.c` in the CCS project explorer window. Double click on this file to open it in the project window for editing.

Sections A & B have been marked in the code for the purpose of pointing users to the relevant locations in the code to perform modifications.

1. To enable RTC, please uncomment the following code. RTC counter will be set to 0, and the match value will be set to 5 seconds.

Please refer to SECTION: A in the code.

i.e. simply remove `/*` and `*/`

Change from:

```
// >>>>>>>>>>>> SECTION: A
//
// Set RTC counter = 0, enable RTC, set RTC match 0 register
// for 5 seconds.
//
/*
HibernateRTCSet(0);
HibernateRTCEnable();
HibernateRTCMatch0Set(5);
*/
```

Change to:

```
// >>>>>>>>>>>> SECTION: A
//
// Set RTC counter = 0, enable RTC, set RTC match 0 register
// for 5 seconds.
//
HibernateRTCSet(0);
HibernateRTCEnable();
HibernateRTCMatch0Set(5);
```

2. Configure hibernation module to wake the processor upon RTC match or WAKE pin assertion. Add `HIBERNATE_WAKE_RTC` in the function arguments for `HibernateWakeSet()` function.

Please refer to SECTION B: in the code.

Change from:

```
// >>>>>>>>>>>> SECTION: B
//
// Configure to wake on RTC match & nWAKE assertion.
```

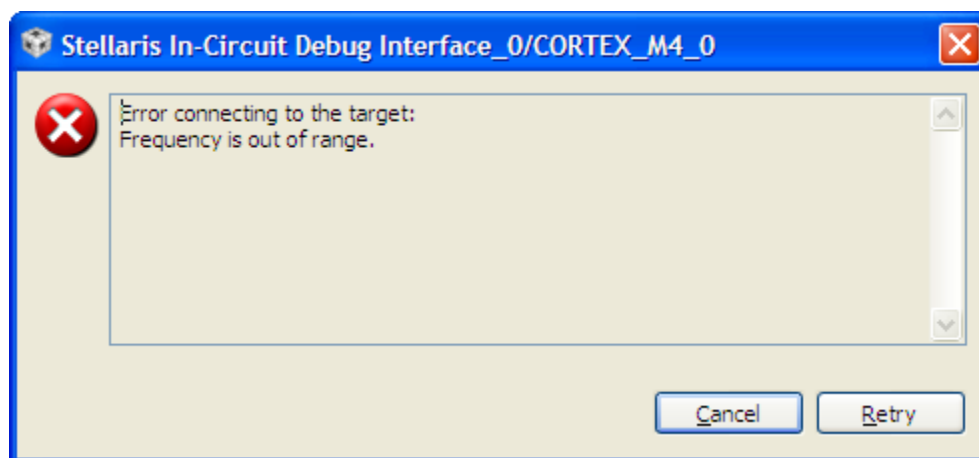
```
//  
HibernateWakeSet (HIBERNATE_WAKE_PIN);
```

Change to:

```
// >>>>>>>>>> SECTION: B  
//  
// Configure to wake on RTC match & nWAKE assertion.  
//  
HibernateWakeSet (HIBERNATE_WAKE_PIN (HIBERNATE_WAKE_RTC) ;
```

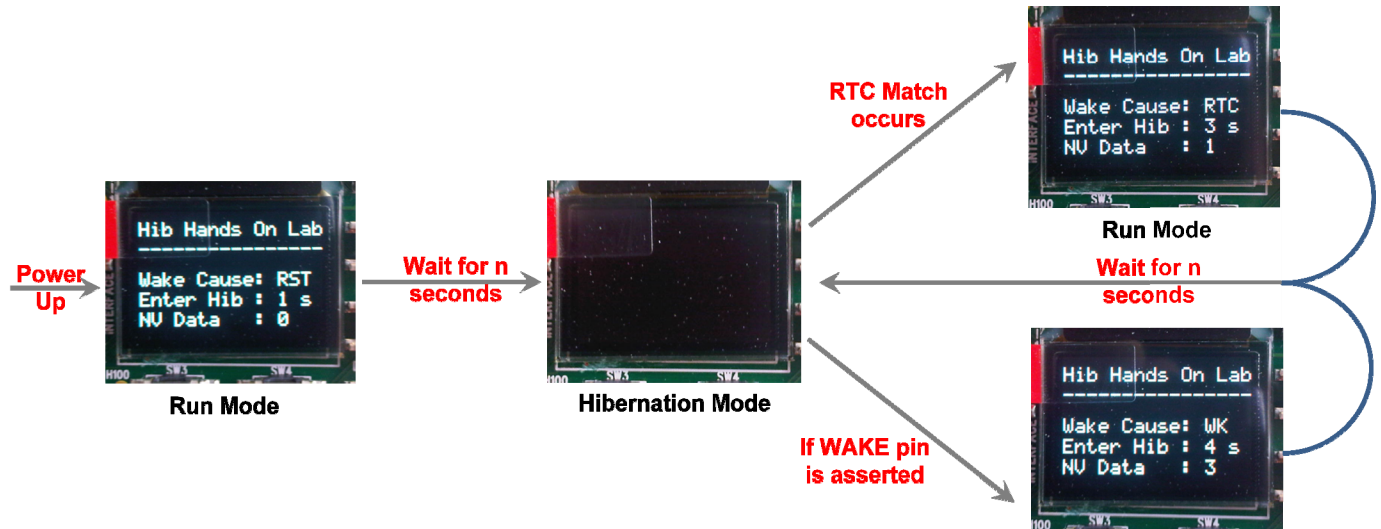
Rebuild the code, make sure that it is built successfully. Press the “debug” button to download the code the microcontroller.

Note: If you see following error message, it means that your board is not connect to your computer. This can happen if there is a issue with ICDI drivers on your computer (which probably won't be the case), or that your board is in hibernation mode while you are attempting to launch the debugger.



Results

Upon successful completion of this exercise, the application should function as follows:



Congratulations, you have completed exercise 1 of this lab!

Exercise 2

Exercise 2: Objective

Modify an existing project (from exercise1) to measure & compare current consumption in:

- (1) Run mode (when PLL is active)
- (2) Hibernation mode
 - When RTC is disabled
 - When RTC is enabled

4



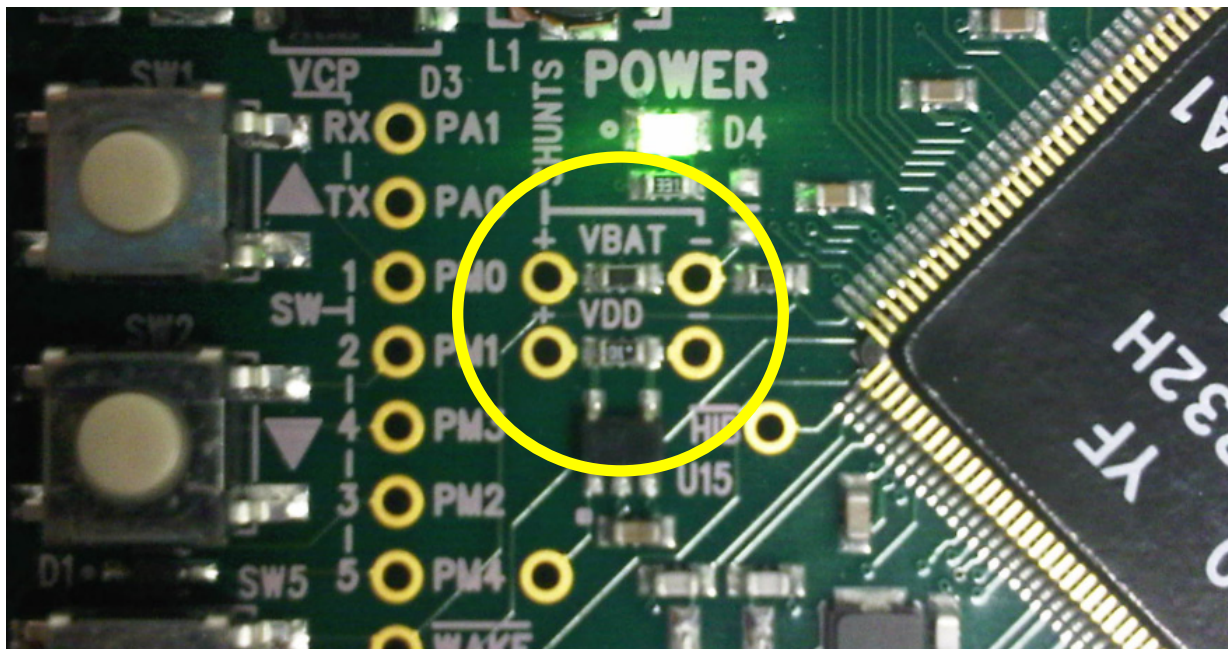
Theory

In this exercise, we will measure and compare microcontroller's current consumption in different modes of operation.

Notably, in run/active mode the current consumption depends on the system clock frequency, system clock source, number of active peripherals etc.

In hibernation mode, the processor and all the peripherals are powered down to lower overall current consumption. The current consumption in hibernation mode with RTC enabled will be slightly greater than when RTC is disabled.

We will use a digital multimeter to measure the current consumption by measuring the voltage drop across R_{VBAT} and R_{VDD} resistors on the evaluation board. Please see the figure below to locate current sense resistors and test/measurement points.



Current consumption can be calculated as follows:

$$I_{\text{HIBERNATE_MODE}} = V_{\text{BAT}} / R_{\text{VBAT}}$$

$$I_{\text{RUN_MODE}} = V_{\text{DD}} / R_{\text{VDD}}$$

Note: $R_{\text{VBAT}} = 1\text{k}\Omega$

$$R_{\text{VDD}} = 0.1\Omega$$

Code Modifications

If you have completed exercise 1 successfully, you do not need to modify your code to accomplish 2/3rd of exercise 2. With application from exercise 1, your evaluation board is cycling between run mode and hibernation mode. Use a digital multimeter to measure the voltages as explained in the previous section of this document and record the values in the table on the next page.

Test	Mode ↓
1	Run Mode
2	Hibernation Mode (RTC enabled)
3	Hibernation Mode (RTC disabled)

Important: To perform tests 1 and 2, you do not need to modify the code/ reprogram the board.

In order to measure the current consumption in hibernation mode with RTC disabled, you will have to “undo” the changes you did to the hibernation project that we started this workshop with.

Test 3 (optional)

1. To disable RTC, please comment the following code. And add HibernateRTCDisable().

Please refer to SECTION: A in the code again. Since you have already modified section A in the previous exercise.

Change from:

```
// >>>>>>>>>> SECTION: A
//
// Set RTC counter = 0, enable RTC, set RTC match 0 register
// for 5 seconds.
//

HibernateRTCSet(0);
HibernateRTCEnable();
HibernateRTCMatch0Set(5);
```

Change to:

```
// >>>>>>>>>> SECTION: A
//
// Set RTC counter = 0, enable RTC, set RTC match 0 register
// for 5 seconds.
//
```

```
/*  
HibernateRTCSet(0);  
HibernateRTCEnable();  
HibernateRTCMatch0Set(5);  
*/  
HibernateRTCDisable();
```

2. Configure hibernation module to wake the processor only on WAKE pin assertion.

Please refer to SECTION B: in the code.

Change from:

```
// >>>>>>>>> SECTION: B  
//  
// Configure to wake on RTC match & nWAKE assertion.  
//  
HibernateWakeSet(HIBERNATE_WAKE_PIN, HIBERNATE_WAKE_RTC);
```

Change to:

```
// >>>>>>>>> SECTION: B  
//  
// Configure to wake on RTC match & nWAKE assertion.  
//  
HibernateWakeSet(HIBERNATE_WAKE_PIN);
```

Rebuild the code, and download it to the microcontroller. Measure the current consumption

Results

Measure voltage across current sense resistors, and calculate the current consumption in different modes in the table below:

Test	Parameter→ Mode ↓	Voltage across sense resistor (V)	Sense Resistor (R)	Current ($I = V/R$)
1.	Run Mode	mV	0.1 Ω	mA
2.	Hibernation Mode (RTC enabled)	mV	1k Ω	μ A
3.	Hibernation Mode (RTC disabled)	mV	1k Ω	μ A

Congratulations, you have completed exercise 2 of this lab!