Dialog SDK 5.x.x Training Materials – Sleep modes

2016 May

…personal
…portable
…connected

BLE Dialog Semiconductor Sleep modes

Sleep mode overview

EXTENDED Sleep mode

DEEP Sleep mode

Powering down individual retention memory cells

Conclusion

BLE Dialog Semiconductor Sleep modes

Sleep mode overview

BLE Dialog Semiconductor Sleep modes

The DA1458x has 2 sleep modes available:

- EXTENDED sleep mode (see bloc diagram next slide):
  - ➢ Only the System RAM 42 kB & Retention RAM remain switched on.

- DEEP sleep mode (see bloc diagram next slide):
  - ➢ Only the Retention RAM remains switched on.

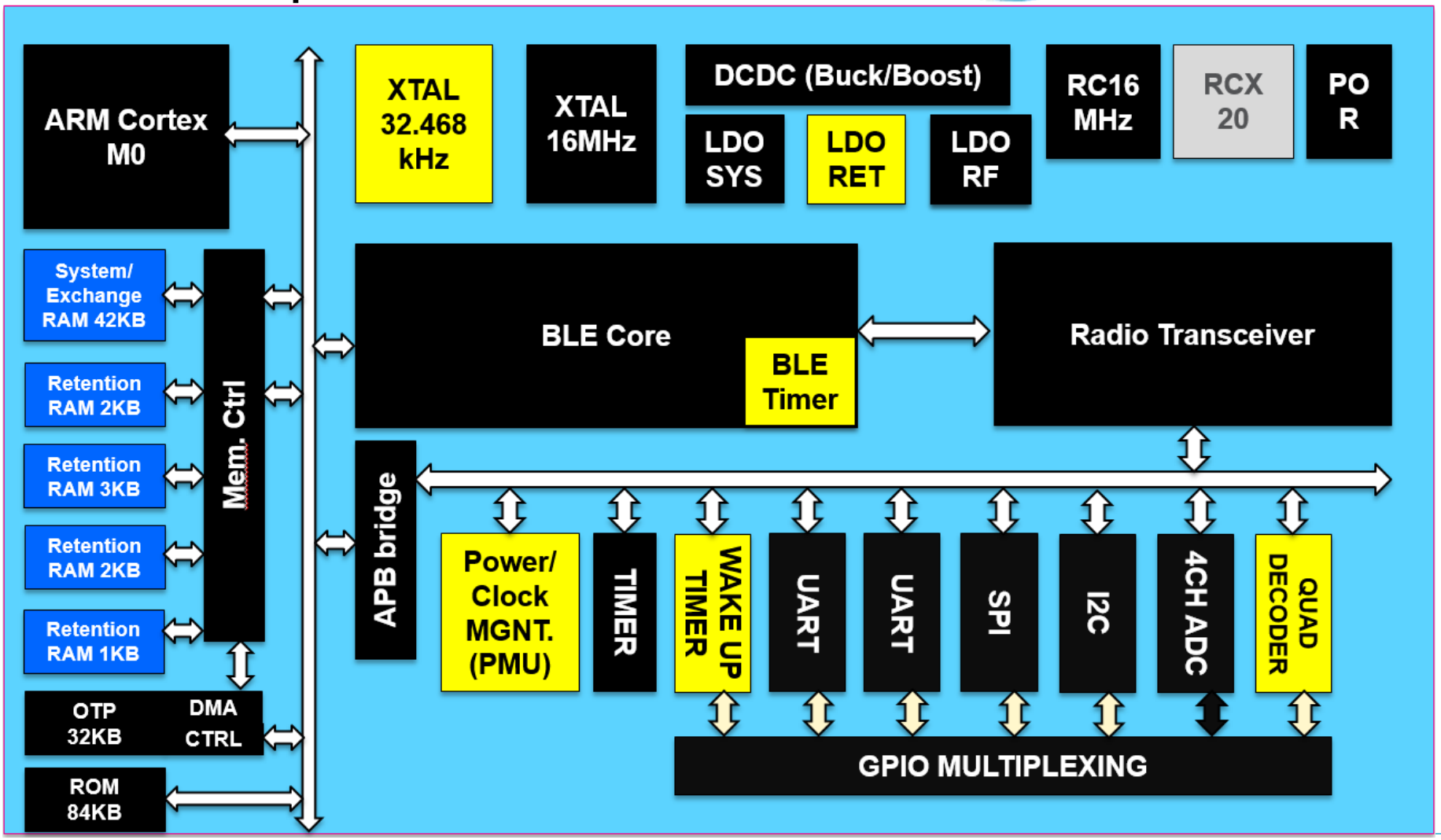
  Note: The OTP must be burnt to be able to measure the DEEP sleep current.

No matter which sleep mode is used, the DA1458x can be woken up in 2 ways:

- Synchronously, via the BLE timer which can be programmed to wake up the system,
- Asynchronously, via an external interrupt (input).

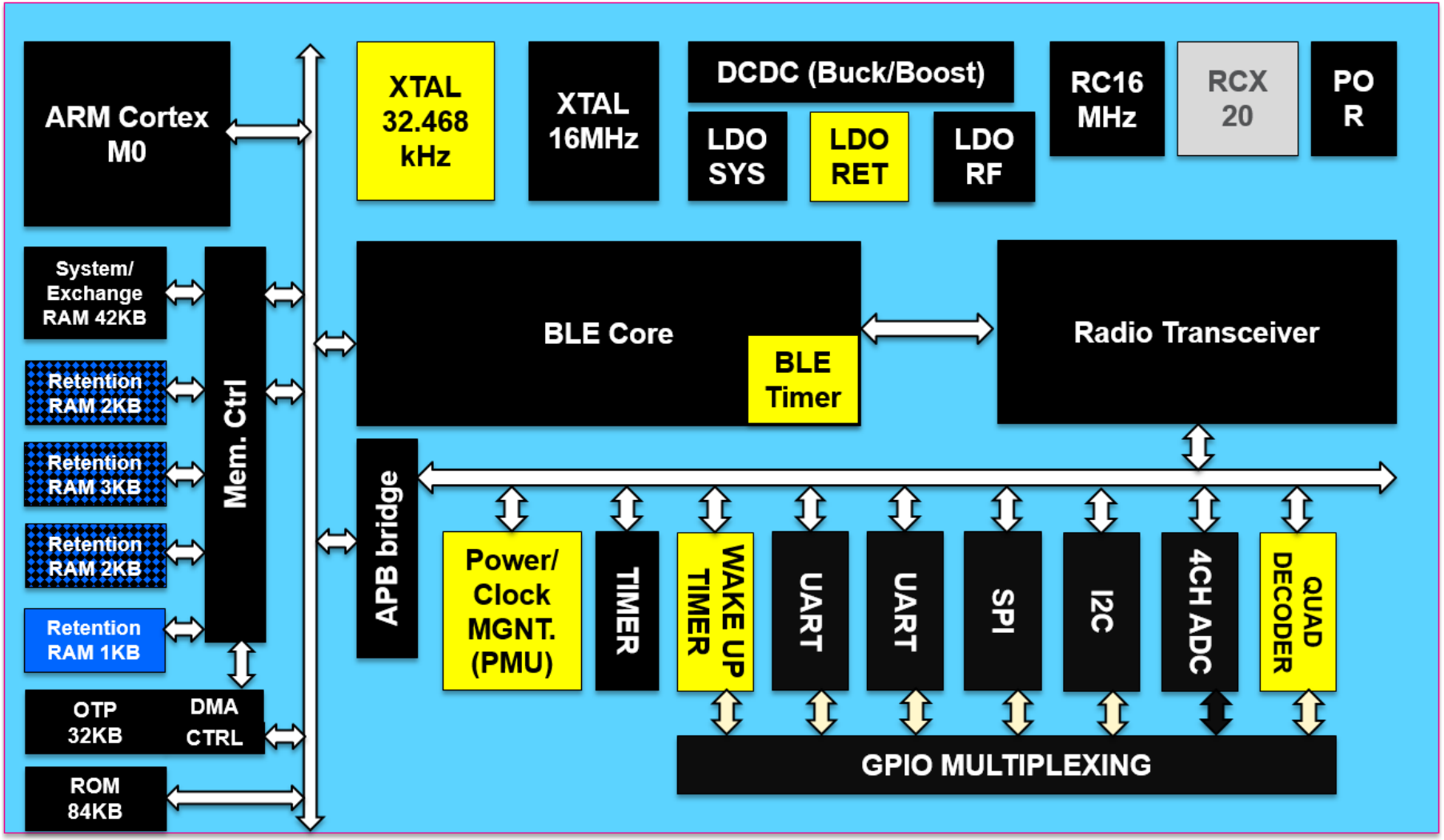
BLE Dialog Semiconductor Sleep modes

EXTENDED sleep mode:



BLE Dialog Semiconductor Sleep modes

DEEP sleep mode:



BLE Dialog Semiconductor Sleep modes

Sleep mode features:

1) External processor solution (via GTL interface):

A periodic wake-up period is used to poll the flow control of the GTL interface using the following #define:

```
#define CFG_MAX_SLEEP_DURATION_PERIODIC_WAKEUP_MS 500  // 500 msec
```

The default value is 500 msec which is a good comprise for pulling the UART interface.

The maximum value is 23.3 hours because a 27-bit timer is used. Max value = 2^27 * 0.625ms(BLE ticks duration)

The minimum value is 10 msec (The DA1458x needs 5.7ms to wake up) which is not an ideal option,
it is just being shown as a reference of the minimum value of a periodic wake-up.

BLE Dialog Semiconductor Sleep modes

Sleep mode features:

2) Internal processor solution:

A periodic wake-up period is used to wake up the DA1458x due to the following #define:

```
#define CFG_MAX_SLEEP_DURATION_EXTERNAL_WAKEUP_MS 10000  // 10s
```

The DA1458x will wake up in the period mentioned (in our case it is 10 sec) when no BLE & timer activities will be processed.

The maximum value is 23.3 hours because a 27-bit timer is used. (2^27 * 0.625ms (BLE ticks))

The minimum value is 10 msec (The DA1458x needs 5.7ms to wake up) which is not an ideal option, it is just being shown as a reference of the minimum value of a periodic wake-up.

It can be disabled before going to sleep mode by calling the API: app_ble_ext_wakeup_on();
➔ **This will disable all BLE events and periodic events.**

When the 58x wakes up from hibernate mode, the following API must be called: app_ble_ext_wakeup_off();

Such procedure has been implemented in the Proximity Tag ref design SW from the link:

http://support.dialog-semiconductor.com/connectivity/reference-design/proximity-tag

BLE Dialog Semiconductor Sleep modes

EXTENDED Sleep mode

BLE Dialog Semiconductor Sleep modes

Setting the EXTENDED sleep mode

TODO 1 - open the proximity reporter project from:

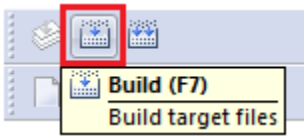
projects\target_apps\ble_examples\prox_reporter\Keil_5

TODO 2 – Open the file /* @file **user_config.h** */ which is under the user_config folder.

TODO 3 – Set the **app_default_sleep_mode** variable to **ARCH_EXT_SLEEP_ON** as shown below:

```
const static sleep_state_t app_default_sleep_mode = ARCH_EXT_SLEEP_ON;
```

TODO 4 – Build the project by pressing the BUILD button :



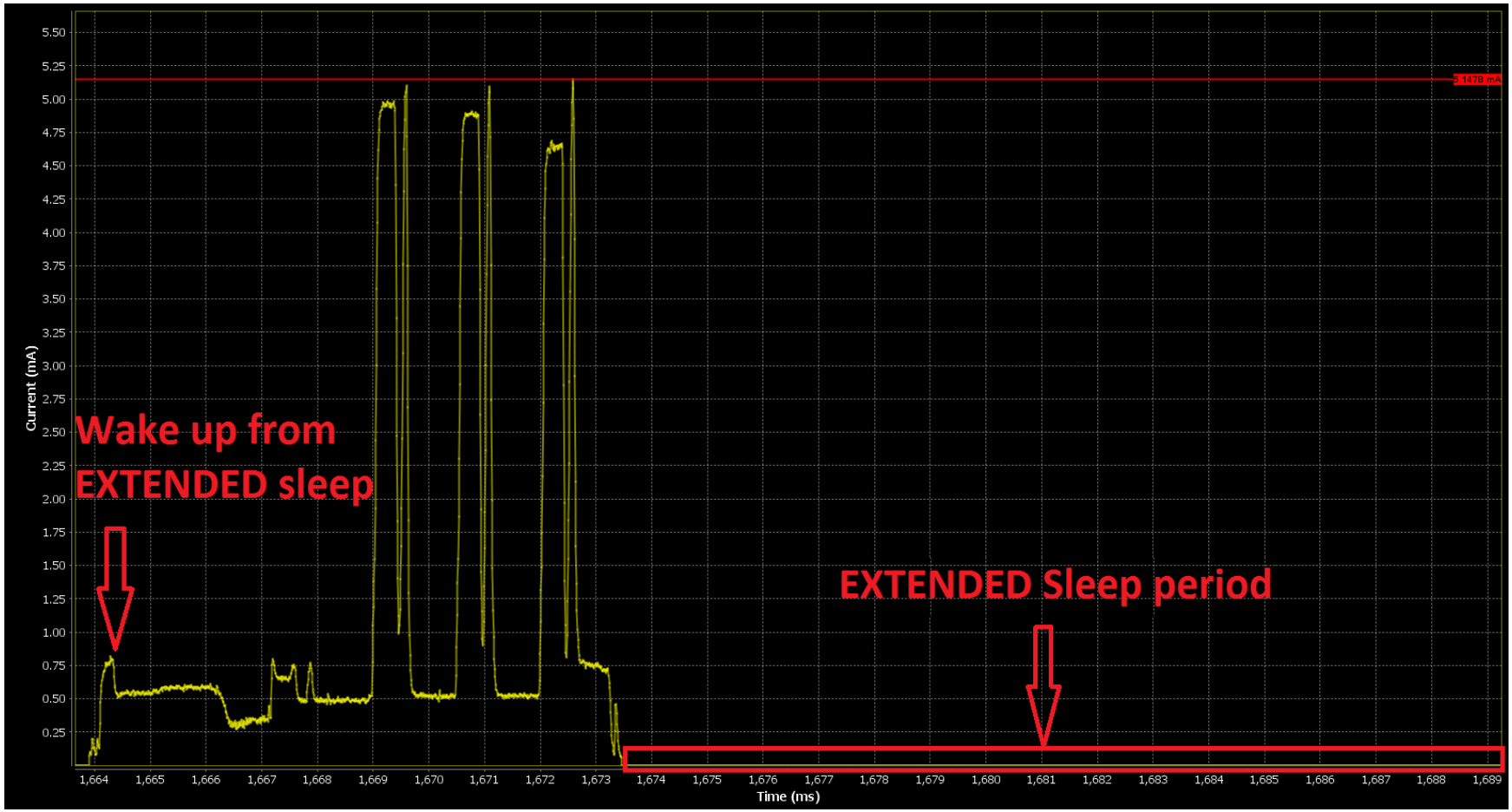
TODO 5 - Connect a PRO or BASIC board to the PC.

TODO 6 - Press the Start DEBUG session button and press again on the same button.
This it will stop the debug session and make the DA1458x running.

BLE Dialog Semiconductor Sleep modes

TODO 6 - open our SmartSnippets tool (available from our portal: http://support.dialog-semiconductor.com/

You should see:



BLE Dialog Semiconductor Sleep modes

Measuring the EXTENDED sleep mode

TODO 1 – Open the file /* @file **user_config.h** */ which is under the user_config folder.

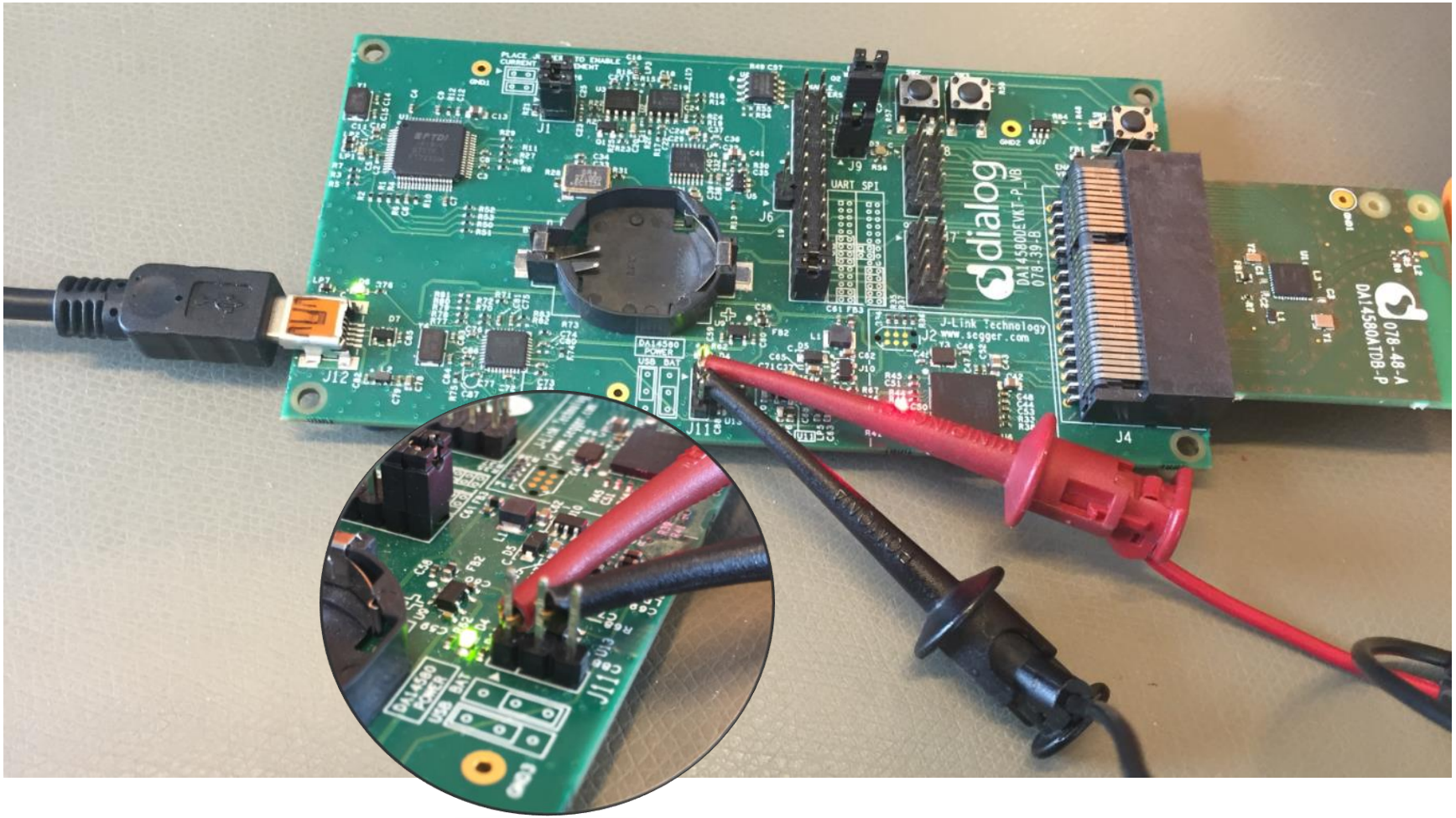
TODO 2 – Change the **.intv** variable (as shown below) to 10000 (=6.2 sec) of the **user_undirected_advertise_conf** structure in order to have a bigger advertising interval. This will leave us some time to measure the EXTENDED sleep current.

```
static const struct advertise_configuration user_undirected_advertise_conf = {
    /// Advertise operation type.
    .advertise_operation = ADV_UNDIRECT,
    /// Own BD address source of the device:
    .address_src = GAPM_PUBLIC_ADDR,
    /// Advertise interval
          .intv = 10000, // EXTENDED SLEEP CURRENT = 10000*0.625 = 6.2 sec
    ///Advertising channel map
    .channel_map = 0x7,
};
```

TODO 3 – Repeat TODO 4 up to TODO 6 of the previous slide.

BLE Dialog Semiconductor Sleep modes

TODO 4 – Connect an Amperemeter as follow to measure the EXTENDED sleep current.



BLE Dialog Semiconductor Sleep modes

TODO 5 – Measure the **EXTENDED sleep** current.

It should be around **1.4 µA**. In our case, we measure 1.35 µA.

It is **NOT RECOMMENDED** to use our SmartSnippets tool to measure currents lower than 100 µA.

BLE Dialog Semiconductor Sleep modes

DEEP Sleep mode

BLE Dialog Semiconductor Sleep modes

Setting the DEEP sleep mode

TODO 1 - Open the proximity reporter project from:

projects\target_apps\ble_examples\prox_reporter\Keil_5

TODO 2 – Open the file /* @file **user_config.h** */ which is under the user_config folder.

TODO 3 – Set the **app_default_sleep_mode** variable to **ARCH_DEEP_SLEEP_ON** as shown below:

```
const static sleep_state_t app_default_sleep_mode = ARCH_DEEP_SLEEP_ON;
```

TODO 4 – Open the file /* @file **da1458x_config_advanced.h** */ which is under the user_config folder.

TODO 5 – Define the CFG_BOOT_FROM_OTP

TODO 6 – Open the file /* @file **da1458x_config_basic.h** */ which is under the user_config folder.

TODO 7 – Undefine the CFG_MEM_MAP_EXT_SLEEP parameter

- Undefine the CFG_DEVELOPMENT_DEBUG parameter
- Define the CFG_MEM_MAP_DEEP_SLEEP parameter

BLE Dialog Semiconductor Sleep modes

Setting the DEEP sleep mode

TODO 8 – Change the **.intv** variable (as shown below) to 10000 (=6.2 sec) of the **user_undirected_advertise_conf** structure in order to have a bigger advertising interval. This will leave us some time to measure the EXTENDED sleep current.

```
static const struct advertise_configuration user_undirected_advertise_conf = {
    /// Advertise operation type.
    .advertise_operation = ADV_UNDIRECT,
    /// Own BD address source of the device:
    .address_src = GAPM_PUBLIC_ADDR,
    /// Advertise interval
            .intv = 10000, // EXTENDED SLEEP CURRENT = 10000*0.625 = 6.2 sec
    ///Advertising channel map
    .channel_map = 0x7,
};
```

TODO 9 – Connect a PRO or BASIC board to the PC.

TODO 10 – Burn the OTP using the SmartSnippets tool.

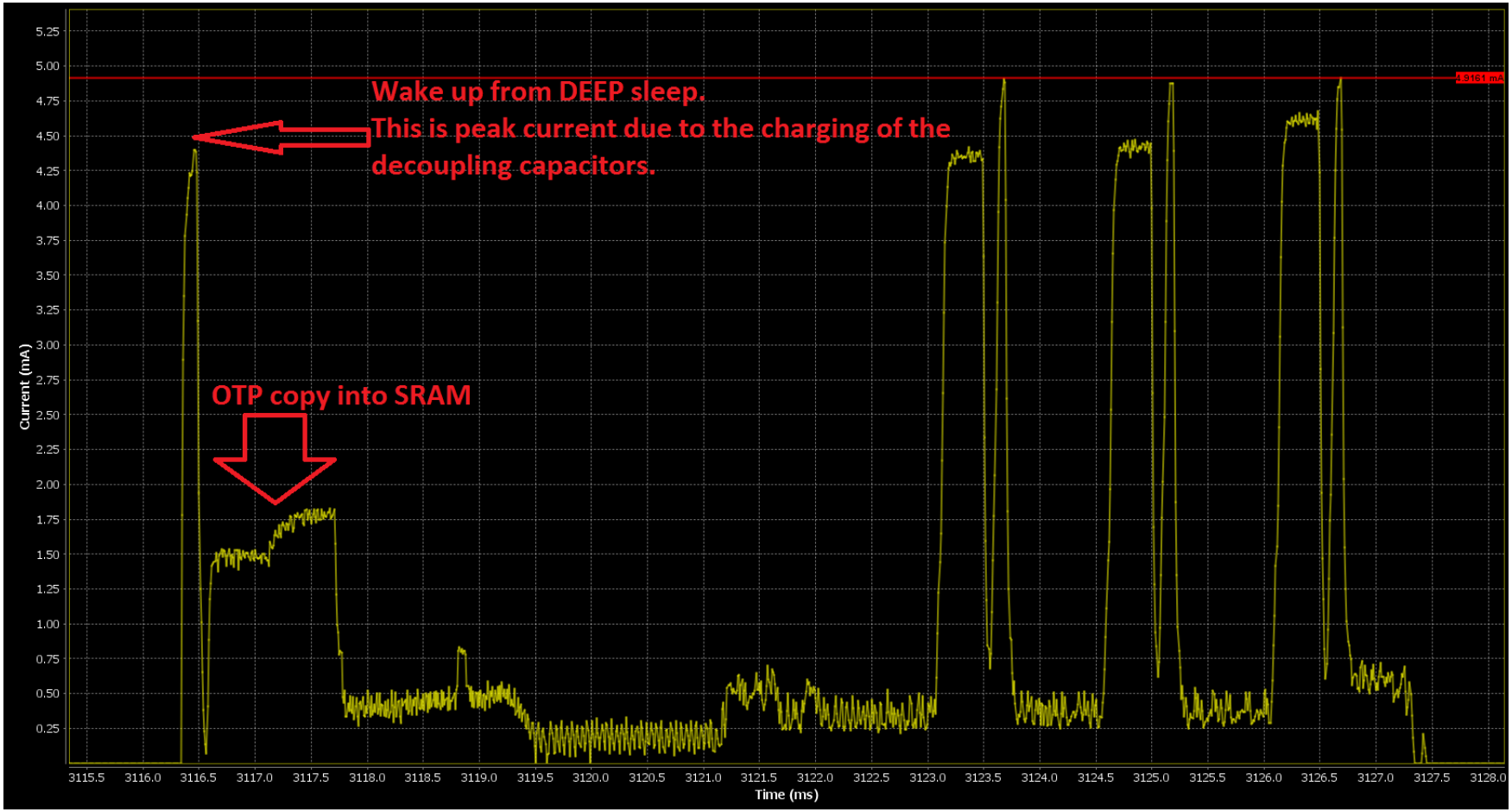
Description on how to burn the OTP is mentioned in the User guide from the Help tab.

BLE Dialog Semiconductor Sleep modes

Setting the DEEP sleep mode

TODO 6 - open our SmartSnippets tool (available from our portal: http://support.dialog-semiconductor.com/

You should see:

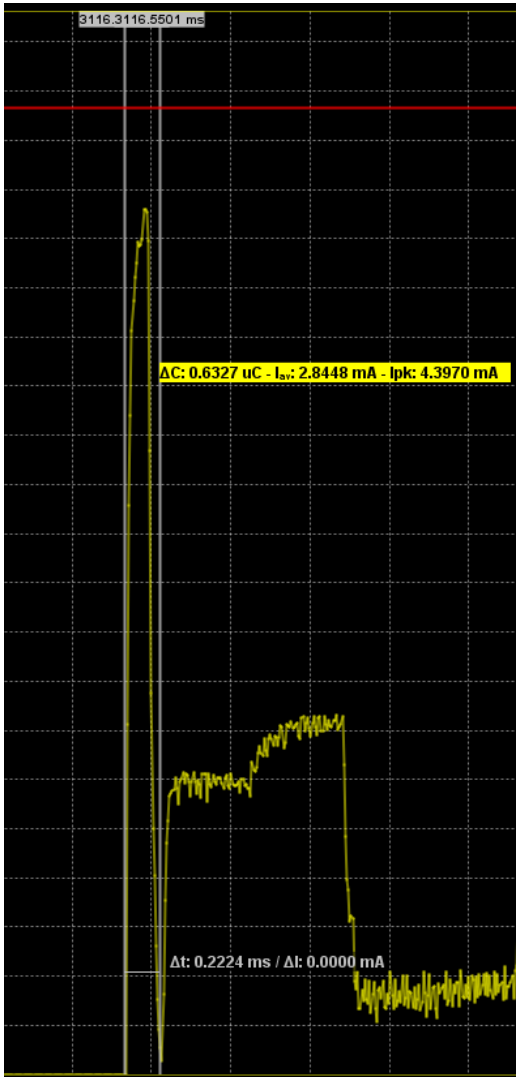


BLE Dialog Semiconductor Sleep modes

Setting the DEEP sleep mode

Measurements:

Current peak due to the charging caps needs ≈ 0.6 µC

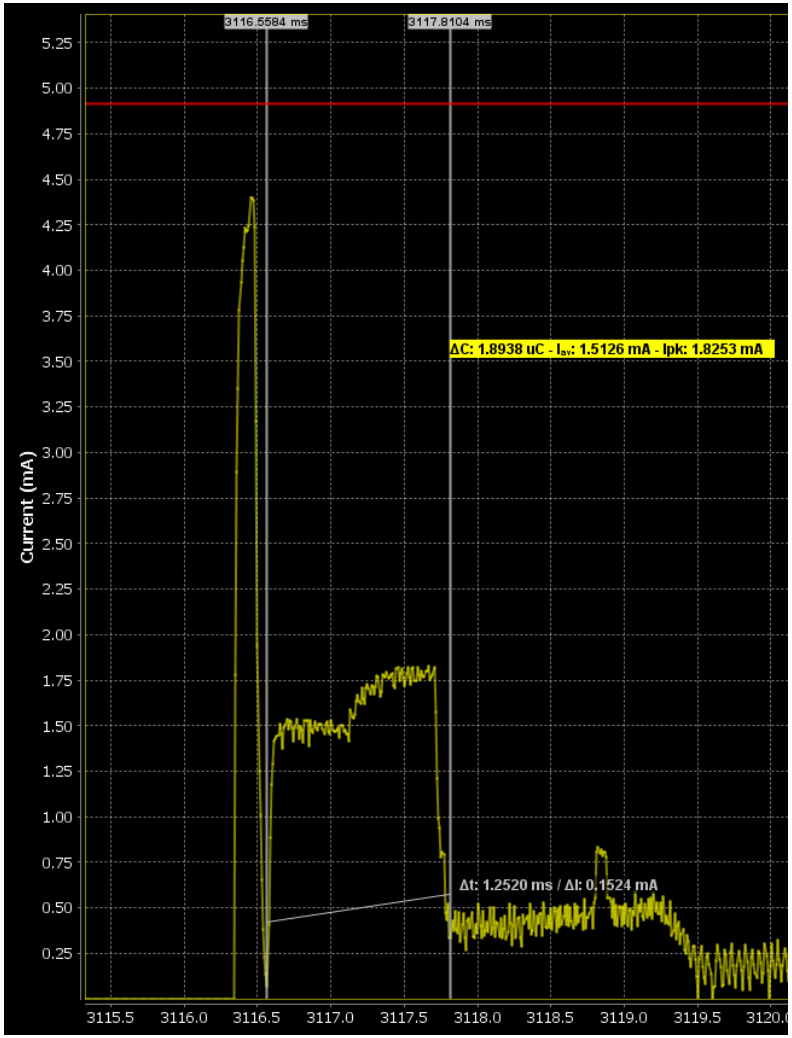


BLE Dialog Semiconductor Sleep modes

Setting the DEEP sleep mode

Measurements:

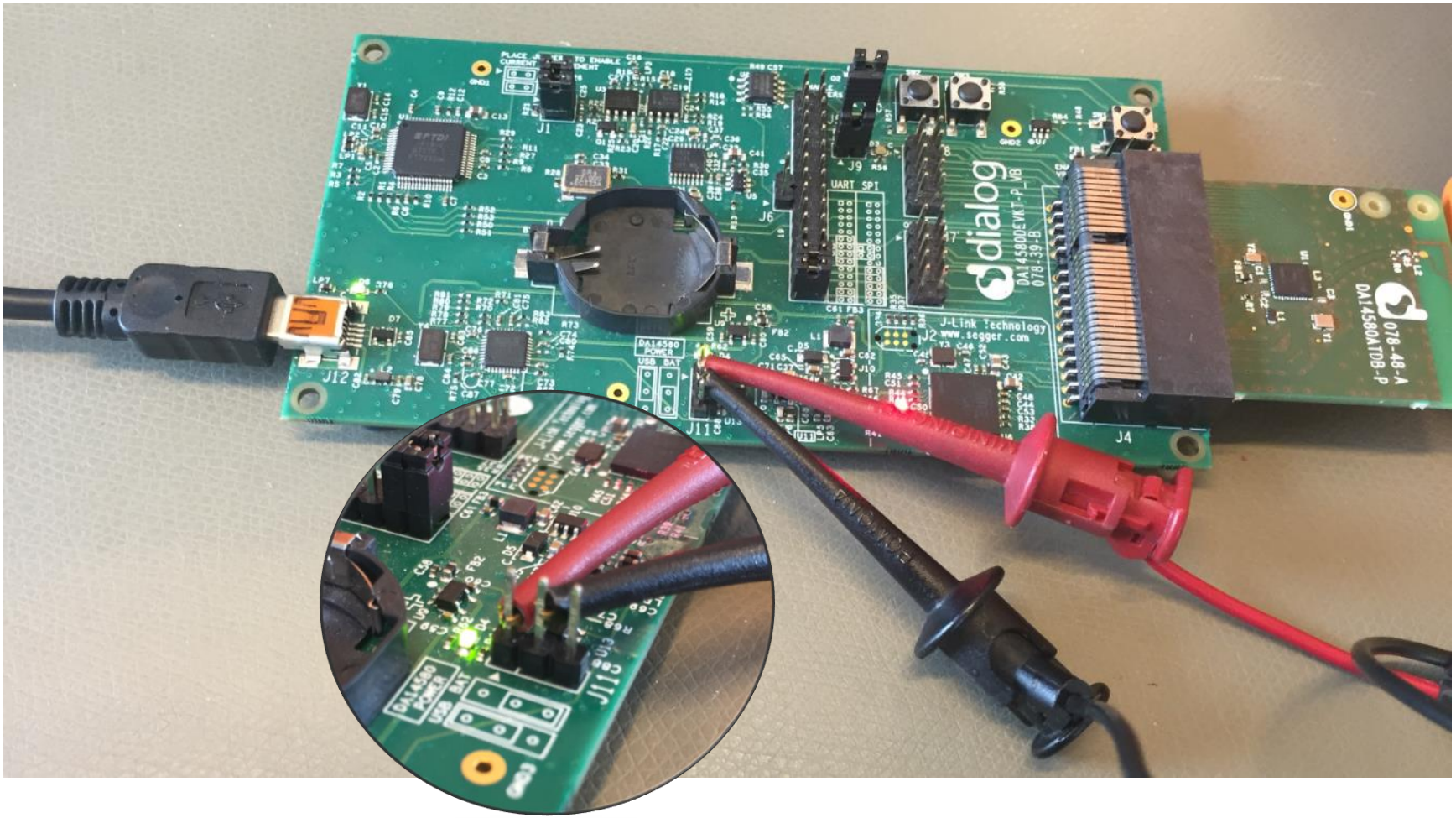
The OTP copy needs ≈ 2 µC



BLE Dialog Semiconductor Sleep modes

Measuring the DEEP sleep mode

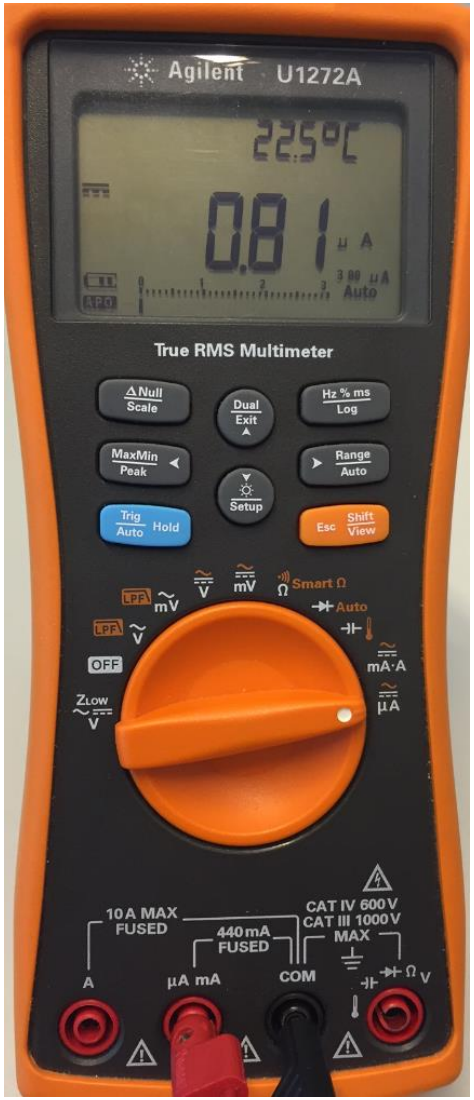
TODO 1 – Connect an Amperemeter as follow to measure the DEEP sleep current.



BLE Dialog Semiconductor Sleep modes

TODO 5 – Measure the **DEEP sleep** current.

It should be around **800 nA**. In our case, we measure 810 nA.



It is **NOT RECOMMENDED** to use our SmartSnippets tool to measure currents lower than 100 µA.

BLE Dialog Semiconductor Sleep modes

Powering down individual retention memory cells

BLE Dialog Semiconductor Sleep modes

The Powering down individual retention memory cells can be only be done in EXTENDED sleep mode.

TODO 1 - open the proximity reporter project from:

projects\target_apps\ble_examples\prox_reporter\Keil_5

TODO 2 – Please find **void SystemInit (void)** procedure

TODO 3 – Change **SetBits16(PMU_CTRL_REG, RETENTION_MODE, 0xF);**

to **SetBits16(PMU_CTRL_REG, RETENTION_MODE, 0x3);**

TODO 4 – Please find **static const struct advertise_configuration user_undirected_advertise_conf**

TODO 5 – Change * .intv = 1100, * ----------> * .intv = 11000, *

TODO 6 – Change to **sleep_state_t app_default_sleep_mode=ARCH_EXT_SLEEP_ON;**

TODO 7 – Build the code and download the binary to the device.

BLE Dialog Semiconductor Sleep modes

A very precise equipment such as the Agilent 34461A 6 1/2 Digit Multimeter has been used to measure the sleep current.

Results:

Depending on the configuration below used, some energy can be saved:

SetBits16(PMU_CTRL_REG, RETENTION_MODE, 0xF); Extended sleep mode current consumption: **2,037 µA**
SetBits16(PMU_CTRL_REG, RETENTION_MODE, 0x3); Extended sleep mode current consumption: **1,957 µA**
The difference is 80 nA

BLE Dialog Semiconductor Sleep modes

Conclusion

BLE Dialog Semiconductor Sleep modes

CONCLUSION: Differences between EXTENDED & DEEP sleep modes

	EXTENDED sleep	DEEP sleep
Memories switched ON	System RAM 42 kB + 8 kB retention RAM	8 kB retention RAM
Current consumption (BUCK mode, 8 kB retention RAM active, external 32kHz crystal used)	≈ 1.4 µA	≈ 800 nA
OTP content copied?	OTP content is **not copied** to SRAM when boot up from extended sleep (so **no impact on the energy consumption**)	OTP content **is copied** into SRAM when boot up from deep sleep (**extra energy: 2.6 µC!**)

For a typical application, if **advertising / connection interval** is less than **2 sec**, **EXTENDED sleep** mode is preferable.

Internal **RCX20 oscillator** (**<500 ppm**), **in BUCK mode ONLY** can be used for:

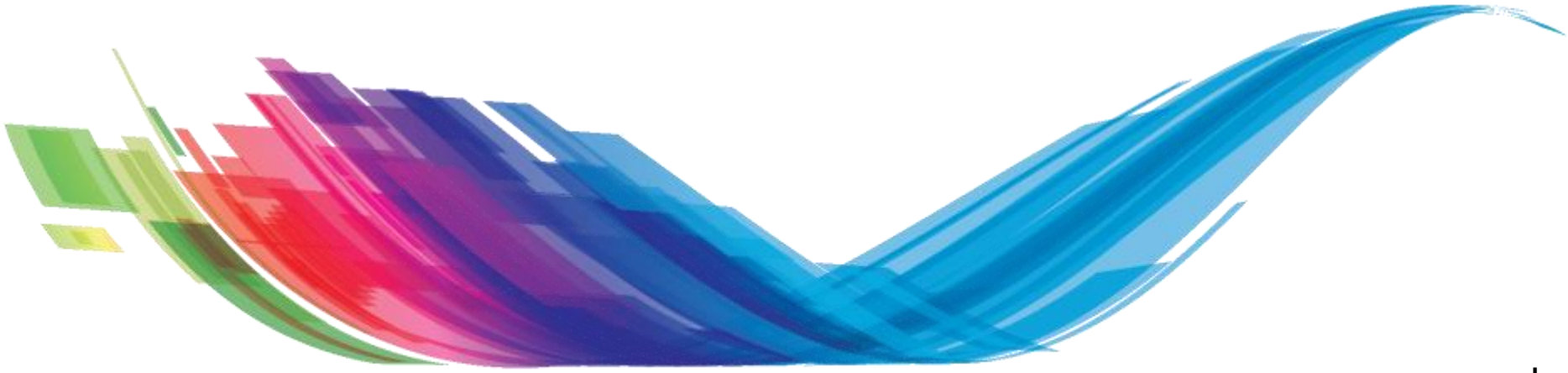
- Counting during both sleep mode
- Counting up to **2 seconds ONLY** while **connected** or during **unlimited time** while **advertising**

BLE Dialog Semiconductor Sleep modes

References

- **Register with Dialog semiconductor to get more development support**
  - http://support.dialog-semiconductor.com/user/register
  - UM-B-006_DA14580_581 Sleep mode configuration

The Power To Be…



…personal
…portable
…connected