Dialog SDK 5.0.x/6.0.x Tutorial

Sleep mode configurations and power measurement

2017 March

…personal
…portable
…connected

Sleep mode configurations and power measurement

Sleep mode configurations and power measurement

Sleep modes overview

Sleep modes overview

The DA14580/1/2/3 has 2 sleep modes available:

- EXTENDED sleep mode (see bloc diagram next section): There is no OTP copy.
  - ➢ DA14580/1/2/3: Only the System RAM 42 kB & Retention RAM remain switched on.

- DEEP sleep mode (see bloc diagram next slide): There is OTP copy (if boot from OTP).
  - ➢ DA14580/1/2/3: Only the Retention RAM remains switched on.

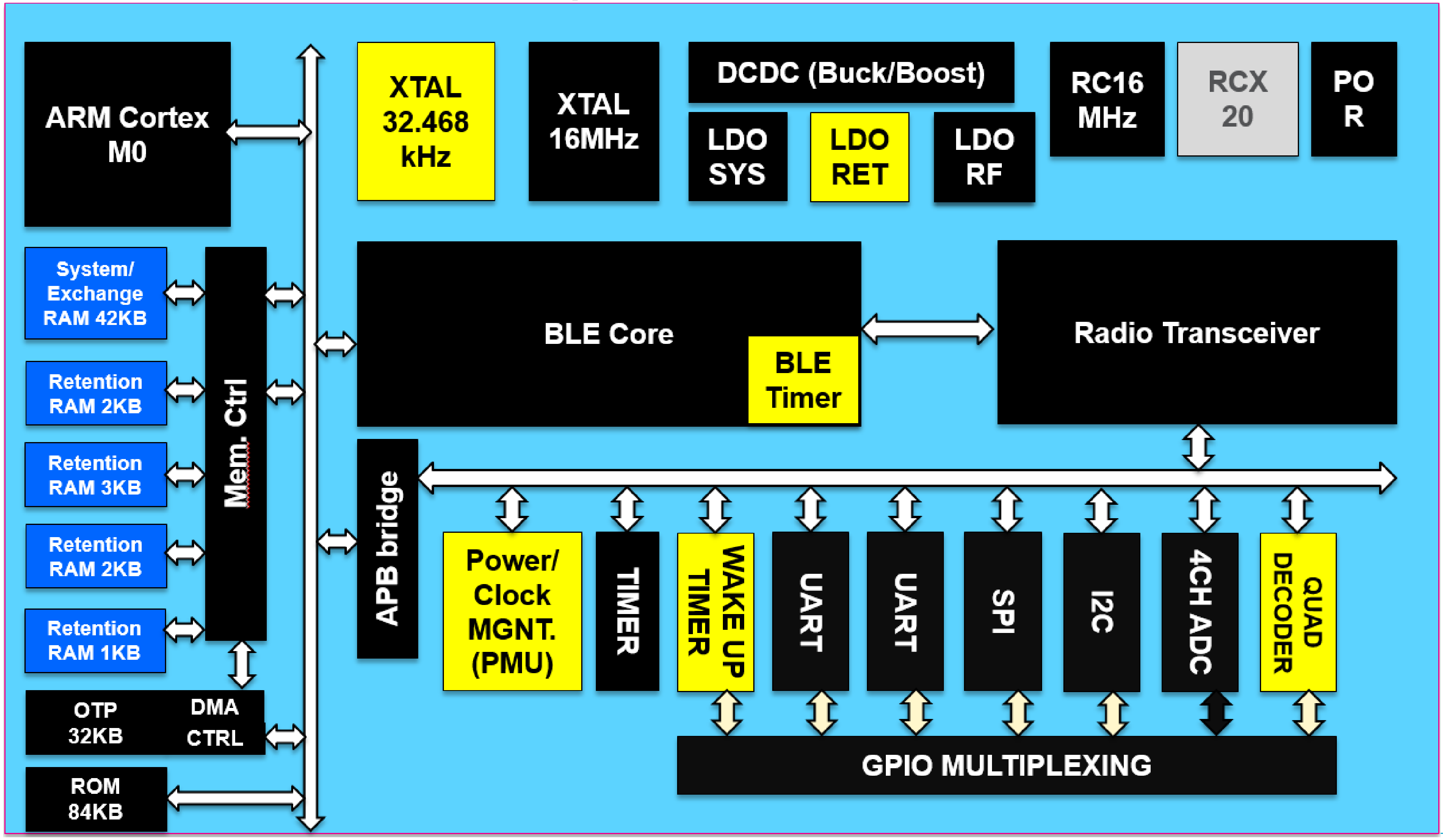
  Note: The OTP must be burnt to be able to measure the DEEP sleep current.

No matter which sleep mode is used, the DA14580/1/2/3 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).

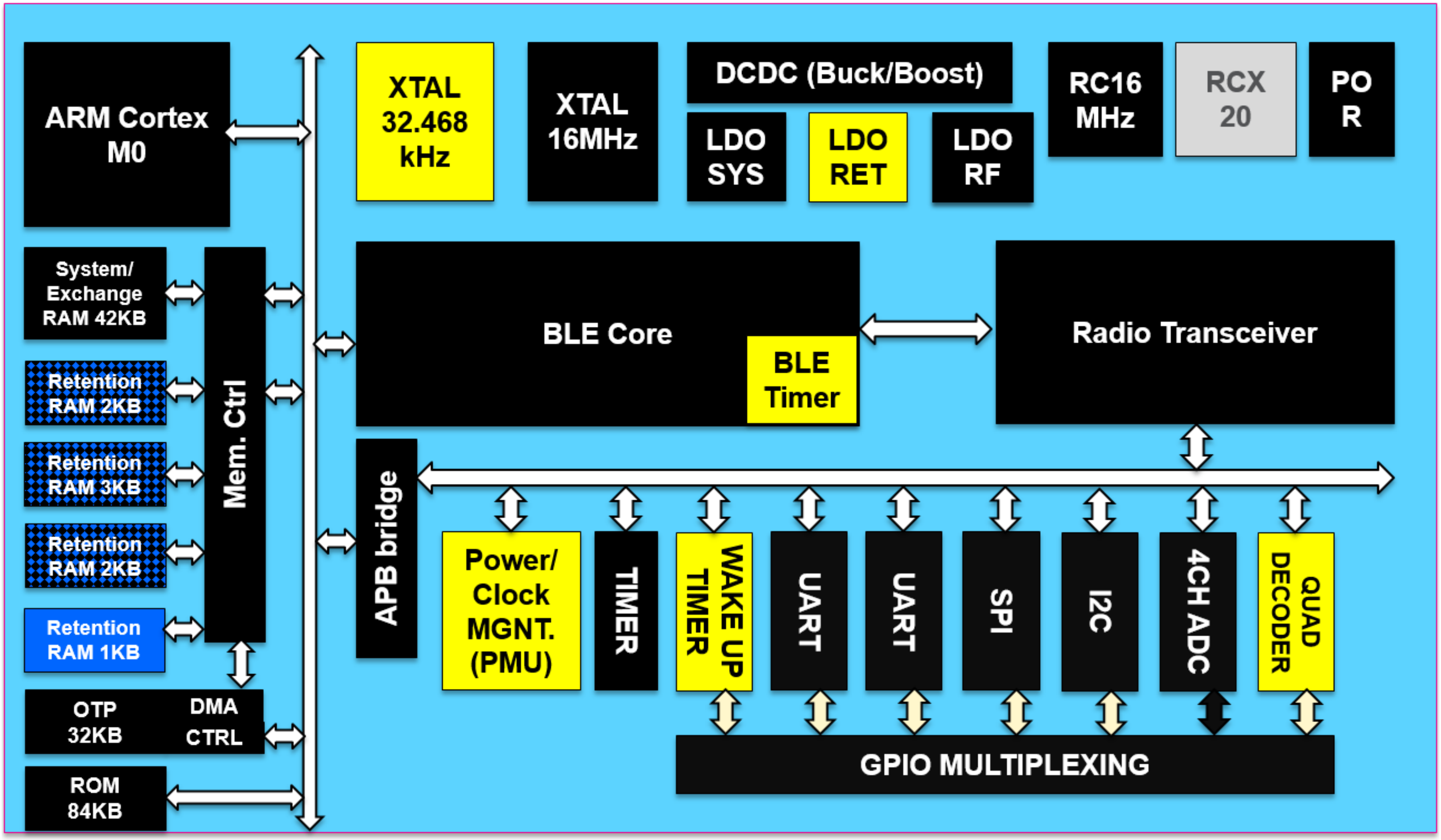
Sleep modes overview

DA14580/1/2/3: EXTENDED sleep mode:



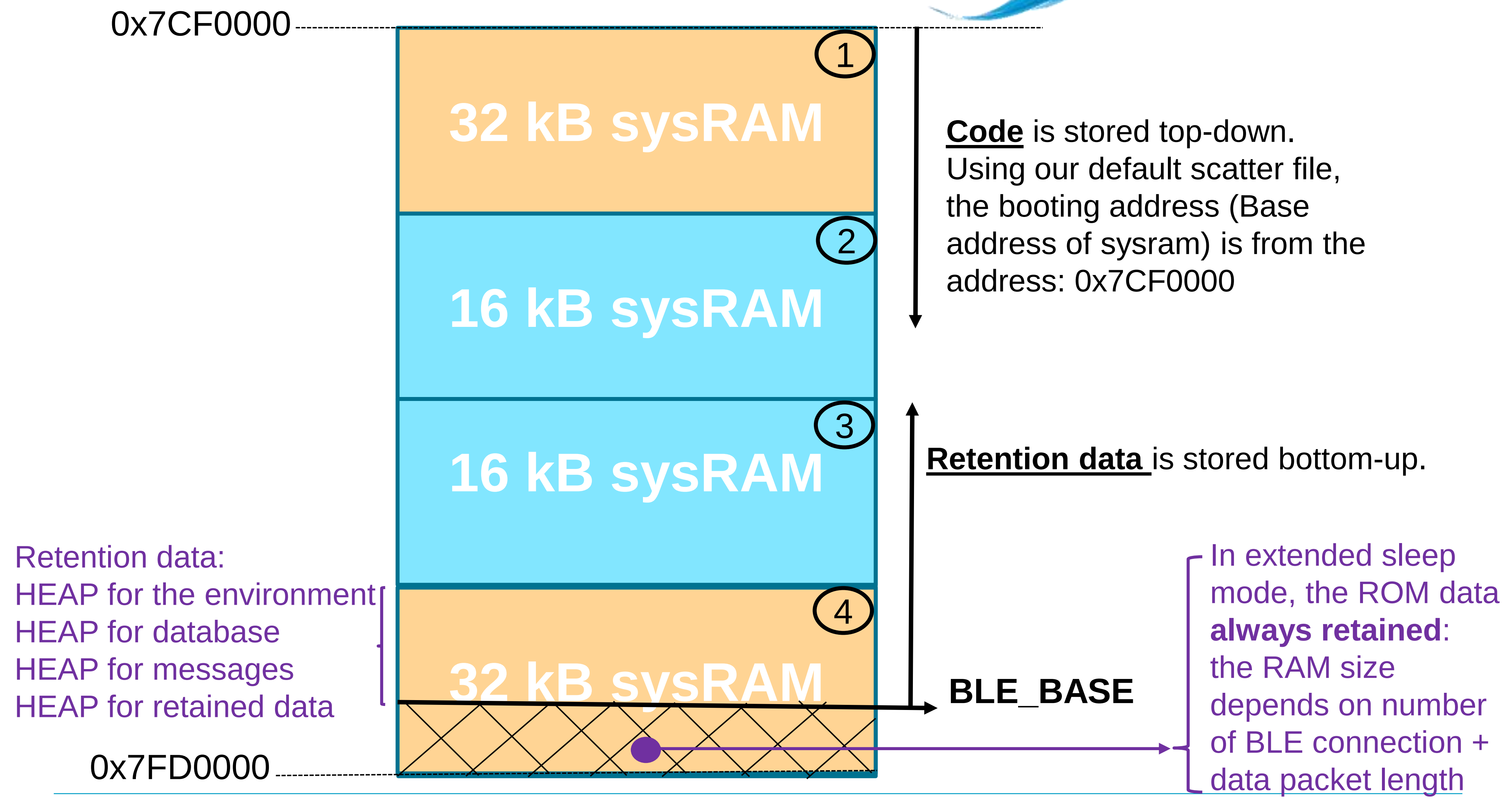
Sleep modes overview

DA14580/1/2/3: DEEP sleep mode:



Sleep modes overview

Memory map of the DA14585/6



Code is stored top-down.
Using our default scatter file, the booting address (Base address of sysram) is from the address: 0x7CF0000

Retention data is stored bottom-up.

Retention data:
HEAP for the environment
HEAP for database
HEAP for messages
HEAP for retained data

In extended sleep mode, the ROM data **always retained**: the RAM size depends on number of BLE connection + data packet length

Sleep modes overview

The DA14585/6 has 3 sleep modes available:

- EXTENDED sleep without OTP copy
  - DA14585/6: Only the System RAM corresponding of the image size stays switched on + the 32kB (block 4 in the previous slide).

- EXTENDED sleep with OTP copy: There is OTP copy (if boot from OTP).
  - DA14585/6: Only 32kB (block 4 from the previous slide) of the System RAM remains switched on.

  <u>Note:</u> The OTP must be burnt to be able to measure the DEEP sleep current.

- DEEP sleep mode: There is OTP copy (if boot from OTP)
  - ➢ DA14585/6: Only the wakeup controller or the POR circuit remains switched on depending on the option selected.

  This mode can be used for the shipping or hibernation mode.
  A BLE connection cannot be maintained.

Sleep modes overview

In the EXTENDED sleep modes, the DA14585/6 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).

When the DEEP sleep mode is selected, the DA14585/6 can be woken up in 1 way:

- Asynchronously, via an external interrupt (input):
  - From the Power On Reset (POR) circuit
  - From the wakeup controller

Sleep modes overview

Sleep mode features (DA1458x):

1) External processor solution (via GTL interface):

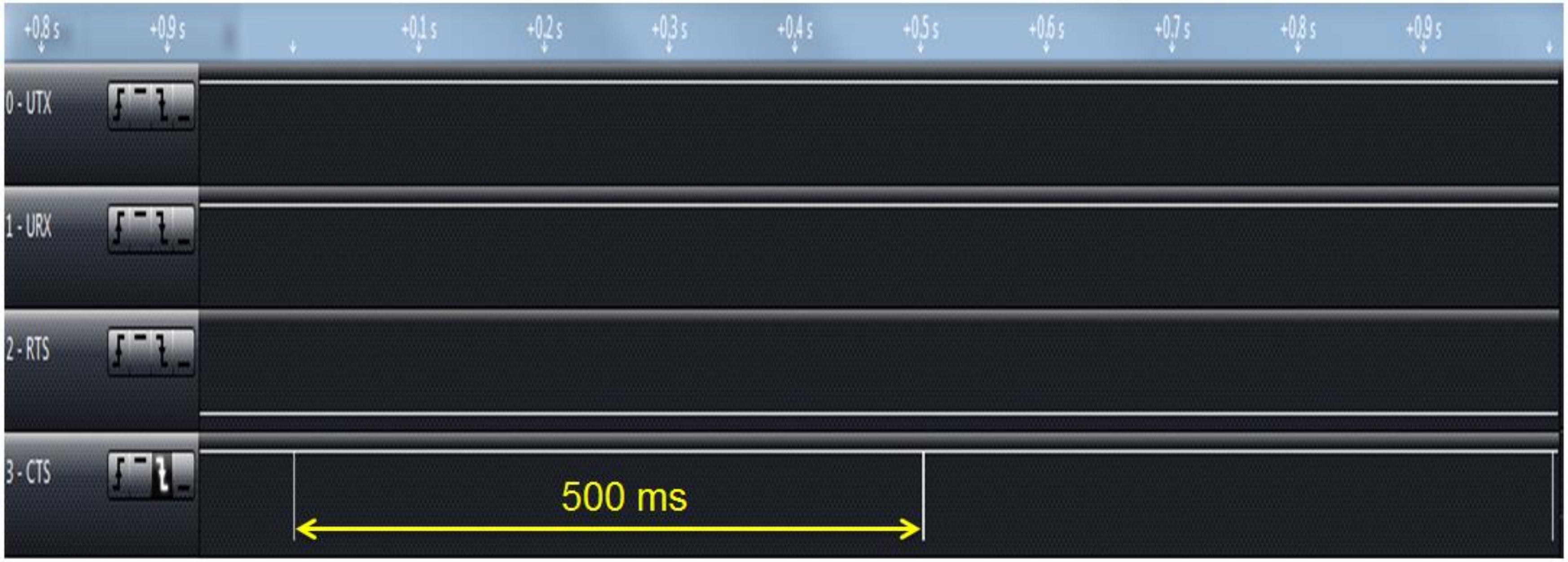
A periodic wake-up period is used to poll the flow control of the GTL interface using the following #define:

```
#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.



Sleep modes overview

Sleep mode features (DA1458x):

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 modes

EXTENDED Sleep modes

Setting the EXTENDED sleep mode (DA14580) or Extended sleep mode without OTP copy (DA14585)

TODO 1 - open the proximity reporter project from:

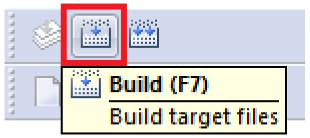
projects\target_apps\ble_examples\prox_reporter\Keil_5

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 :



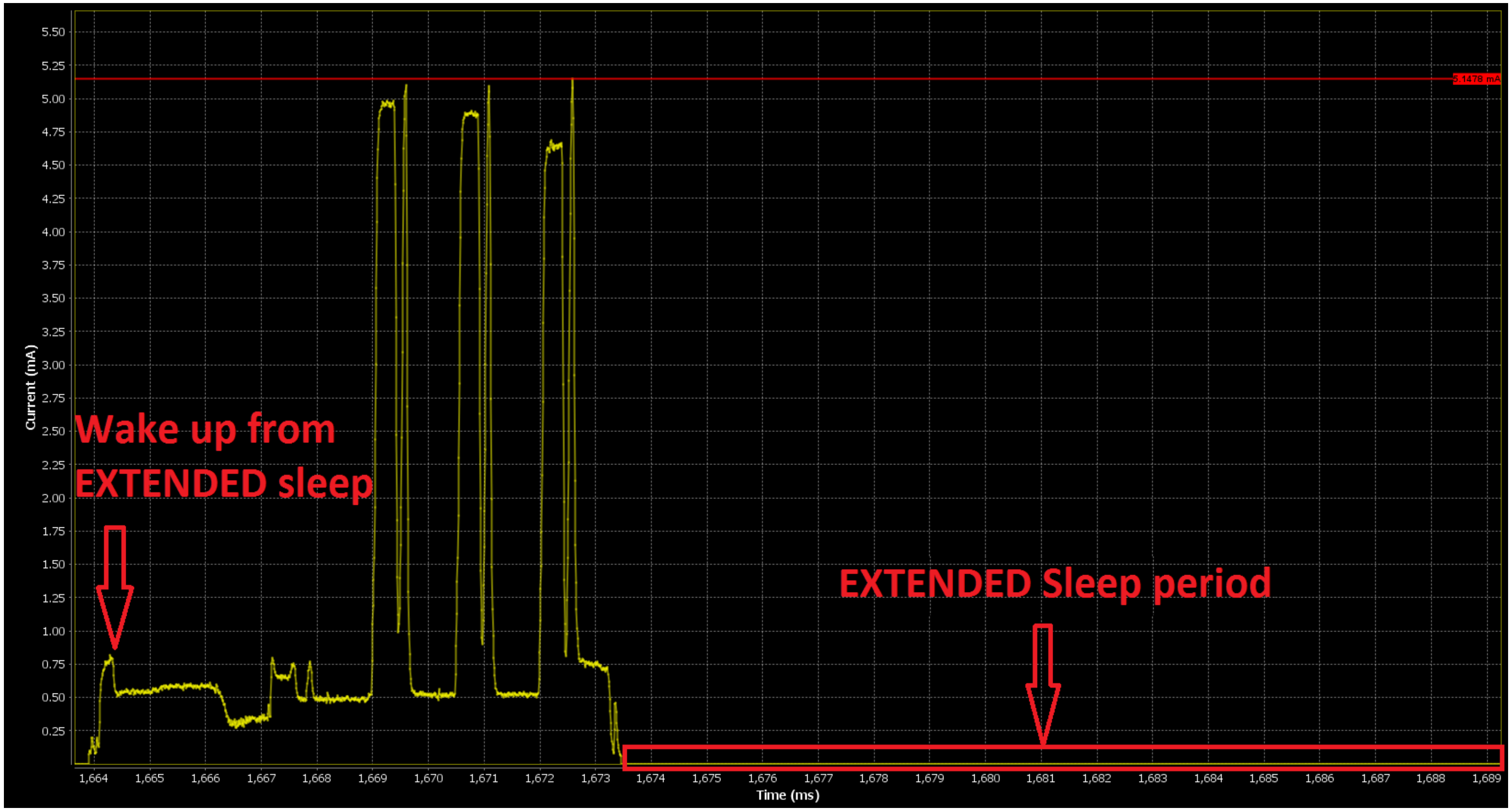
TODO 5 - Connect a PRO board to the PC.

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.

EXTENDED Sleep modes

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

You should see:



EXTENDED Sleep modes

Measuring the EXTENDED sleep mode (DA14580/1/2/3)

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.

EXTENDED Sleep modes

Measuring the Extended sleep mode without OTP copy (DA14585/6)

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

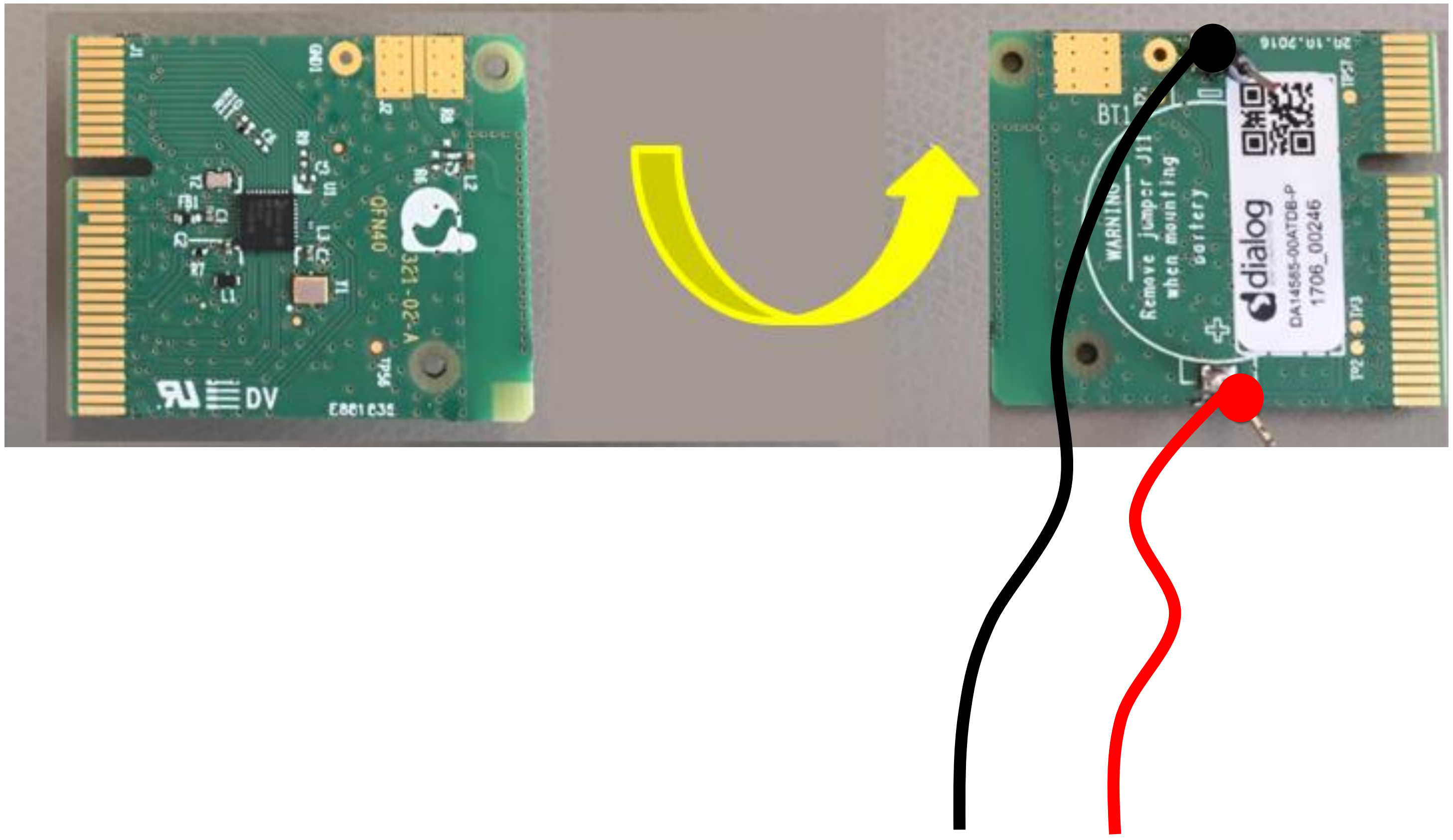
TODO 2 – Change the **.intv_max & .int_min** variable (as shown below) to 60000 (=6 sec) of the **advertise_configuration user_adv_conf** structure in order to have a bigger advertising interval This will leave us some time to measure the EXTENDED sleep current.

```
static const struct advertise_configuration user_adv_conf = {
    /**
     * Own BD address source of the device:
     * - GAPM_STATIC_ADDR: Public or Private Static Address according to device address configuration
     * - GAPM_GEN_RSLV_ADDR: Generated resolvable private random address
     * - GAPM_GEN_NON_RSLV_ADDR: Generated non-resolvable private random address
     */
    .addr_src = GAPM_STATIC_ADDR,

    /// Minimum interval for advertising
    .intv_min = MS_TO_BLESLOTS(6000),                    // 6000ms

    /// Maximum interval for advertising
    .intv_max = MS_TO_BLESLOTS(6000),                    // 6000ms
```

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

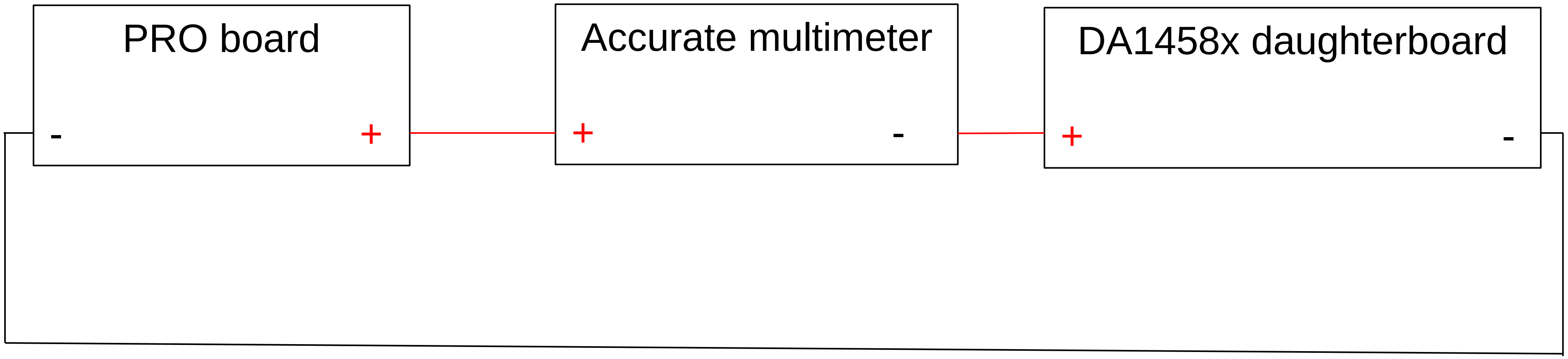
BLE Dialog Semiconductor Sleep modes

TODO 4 - Turn around your DA1458x daughterboard, then solder 2 wires as shown below



BLE Dialog Semiconductor Sleep modes

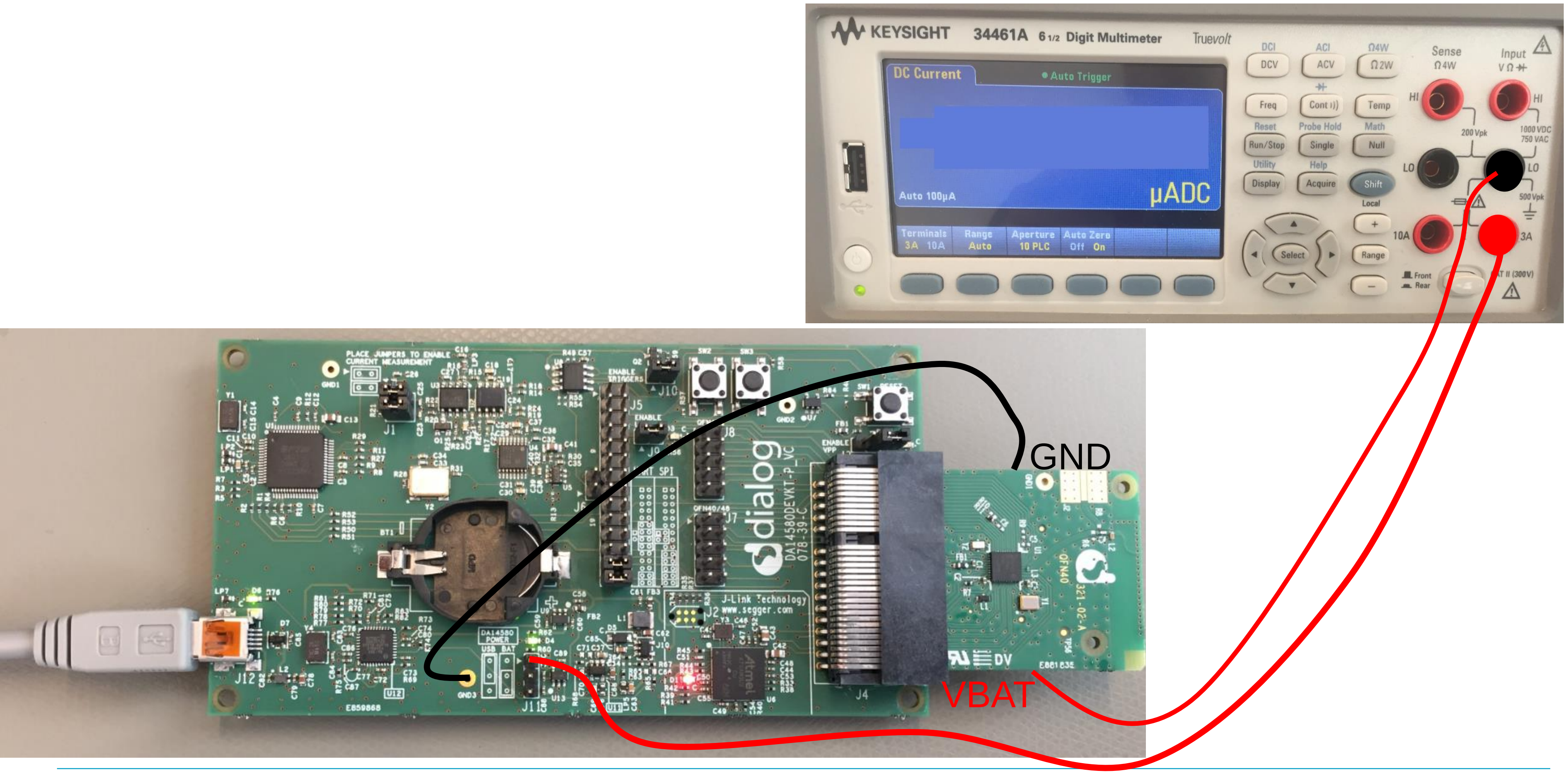
INFO: To measure the sleep currents, we will follow the following block diagram:



BLE Dialog Semiconductor Sleep modes

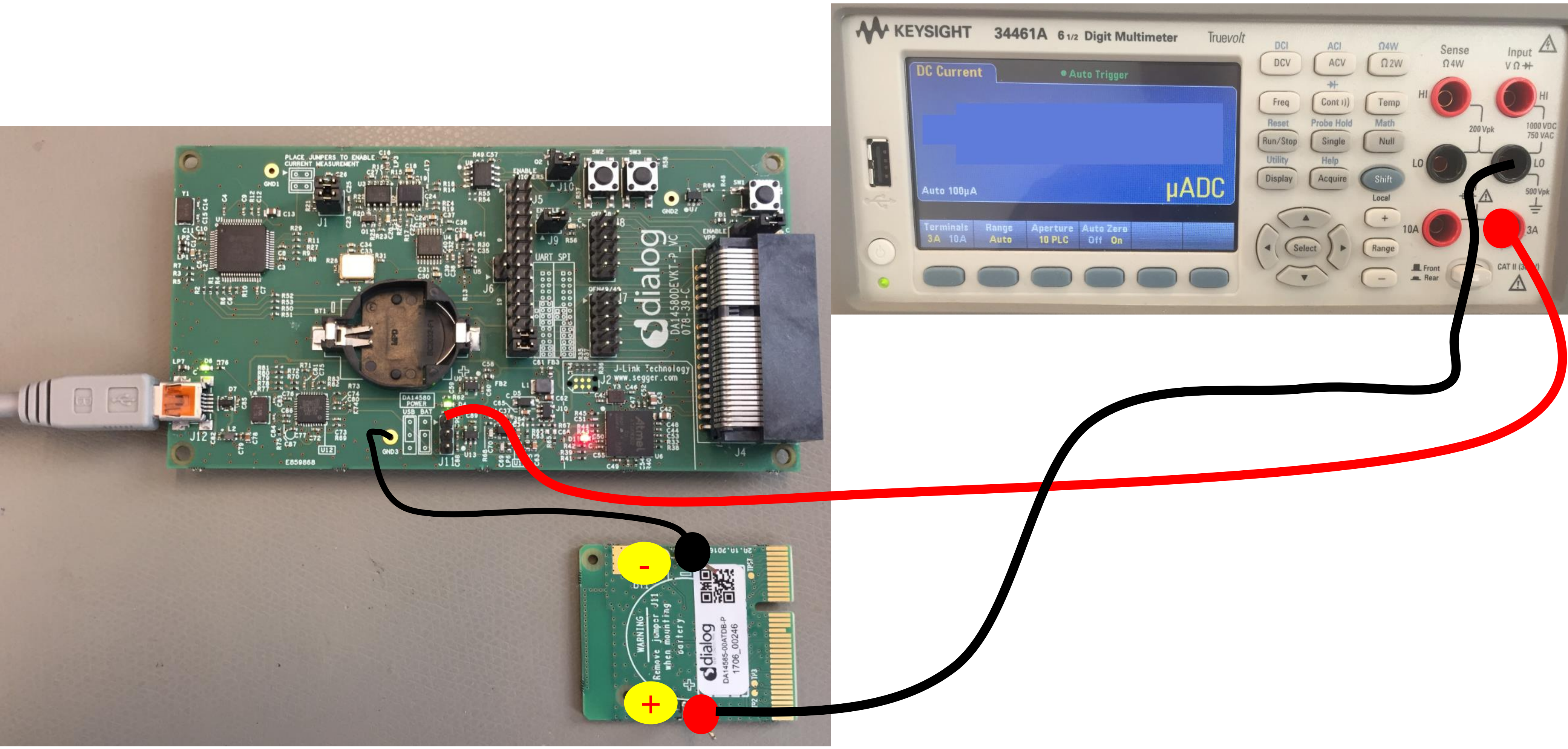
TODO 5 – Connect the daughter board to the J4 connector of the PRO board.

Program the daughter board.



BLE Dialog Semiconductor Sleep modes

TODO 6 – Remove the daughter board from the J4 connecter.



EXTENDED Sleep modes

TODO 7 – Measure the **EXTENDED sleep** current.

For **DA14580:**

We measured: **1.35 µA.**

For **DA14585:**

We measured: **3.4 µA** for extended sleep mode (without OTP copy).

IMPORTANT NOTICE

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

EXTENDED Sleep modes

Setting the Extended sleep mode with OTP copy (DA145855)

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_OTP_COPY_ON** as shown below:

```
const static sleep_state_t app_default_sleep_mode = ARCH_EXT_SLEEP_OTP_COPY_ON;
```

EXTENDED Sleep modes

Setting the Extended sleep mode with OTP copy (DA14585)

TODO 4 - Change the **.intv_max & .int_min** variable (as shown below) to 60000 (=6 sec) of the **advertise_configuration user_adv_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_adv_conf = {
    /**
     * Own BD address source of the device:
     * - GAPM_STATIC_ADDR: Public or Private Static Address according to device address configuration
     * - GAPM_GEN_RSLV_ADDR: Generated resolvable private random address
     * - GAPM_GEN_NON_RSLV_ADDR: Generated non-resolvable private random address
     */
    .addr_src = GAPM_STATIC_ADDR,

    /// Minimum interval for advertising
    .intv_min = MS_TO_BLESLOTS(6000),                    // 6000ms

    /// Maximum interval for advertising
    .intv_max = MS_TO_BLESLOTS(6000),                    // 6000ms
```

EXTENDED Sleep modes

Setting the Extended sleep mode with OTP copy (DA14585)

TODO 5: In the da1458x_config_advanced.h, we need to have the following:

```
#undef CFG_CODE_LOCATION_EXT
#define CFG_CODE_LOCATION_OTP
```

TODO 6: In the da1458x_config_basic.h, we need to have:

```
#undef CFG_DEVELOPMENT_DEBUG
```

TODO 7 – Connect a PRO board to the PC.

TODO 8 – Burn the OTP using the SmartSnippets tool.

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

EXTENDED Sleep modes

TODO 9 – Measure the **EXTENDED sleep** current.

For **DA14585:**

In our case, we measure **2.8 µA** for extended sleep mode with OTP copy.

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

BLE Dialog Semiconductor Sleep modes

DEEP Sleep modes

DEEP sleep modes

Setting the DEEP sleep mode (DA14580)

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

DEEP sleep modes

Setting the DEEP sleep mode (DA14580)

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 board to the PC.

TODO 10 – Burn the OTP using the SmartSnippets tool.

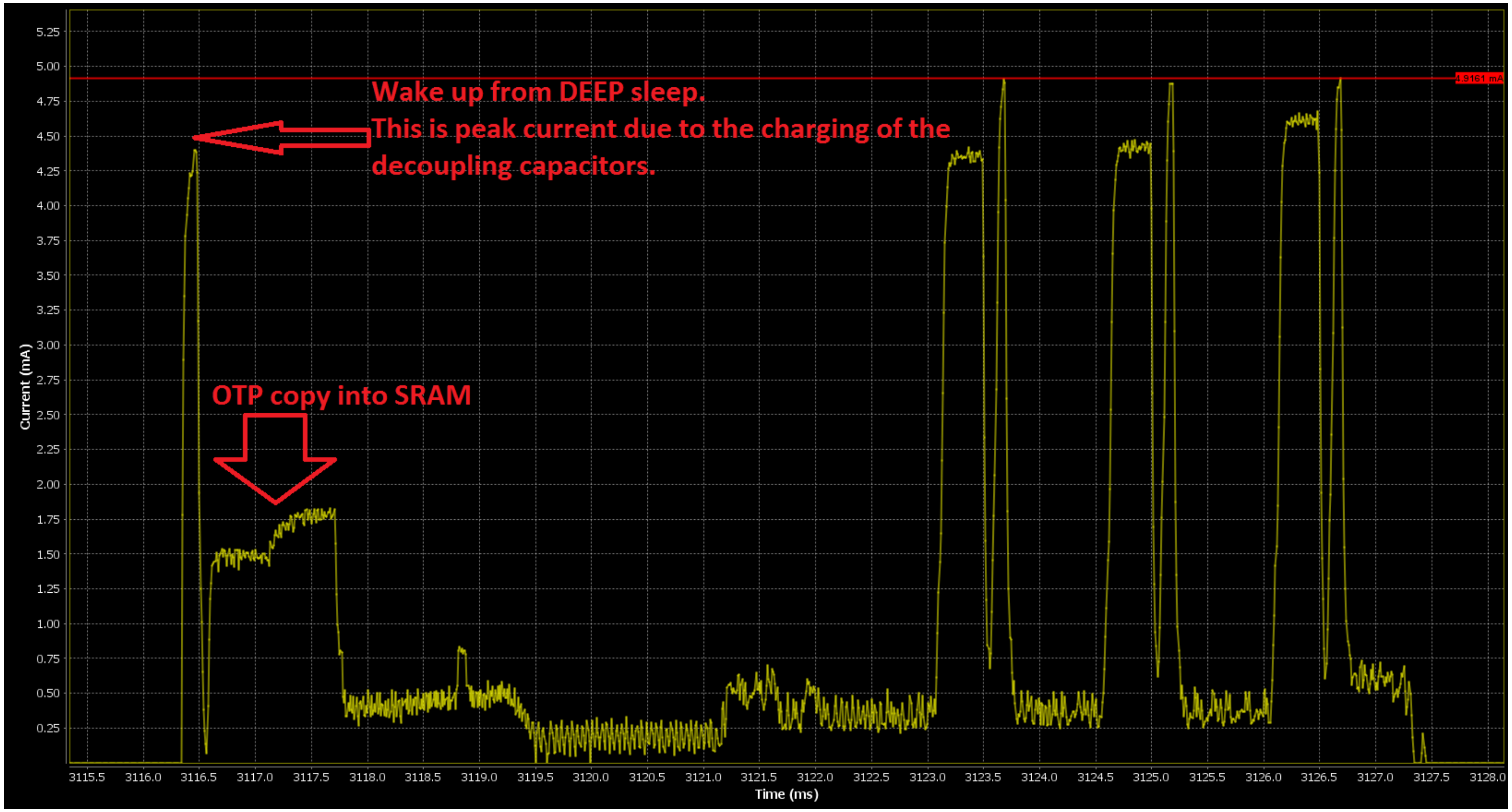
Description on how to burn the OTP is mentioned in the User guide from the Help tab.

DEEP sleep modes

Setting the DEEP sleep mode (DA14580)

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

You should see:

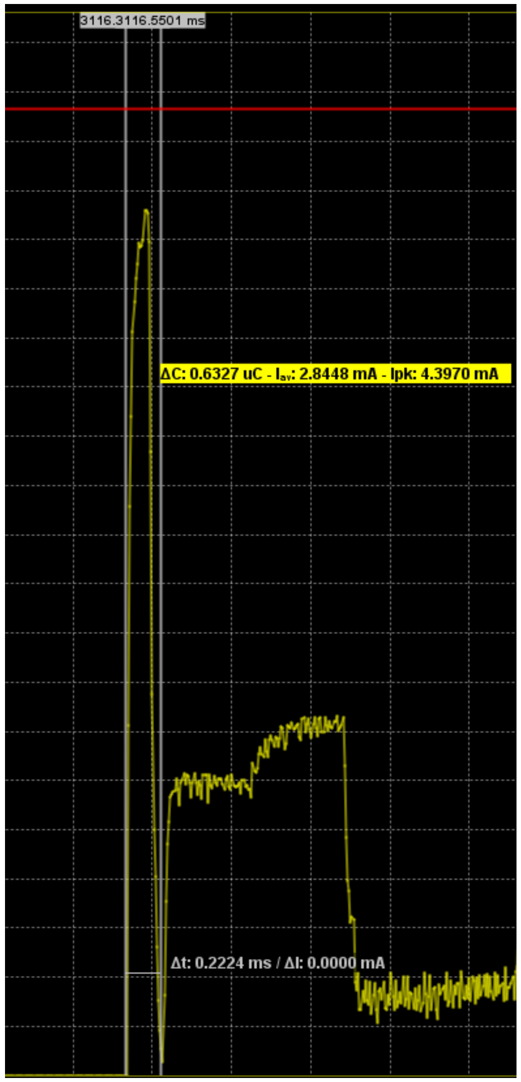


DEEP sleep modes

Setting the DEEP sleep mode (DA14580)

Measurements:

Current peak due to the charging caps needs ≈ 0.6 µC

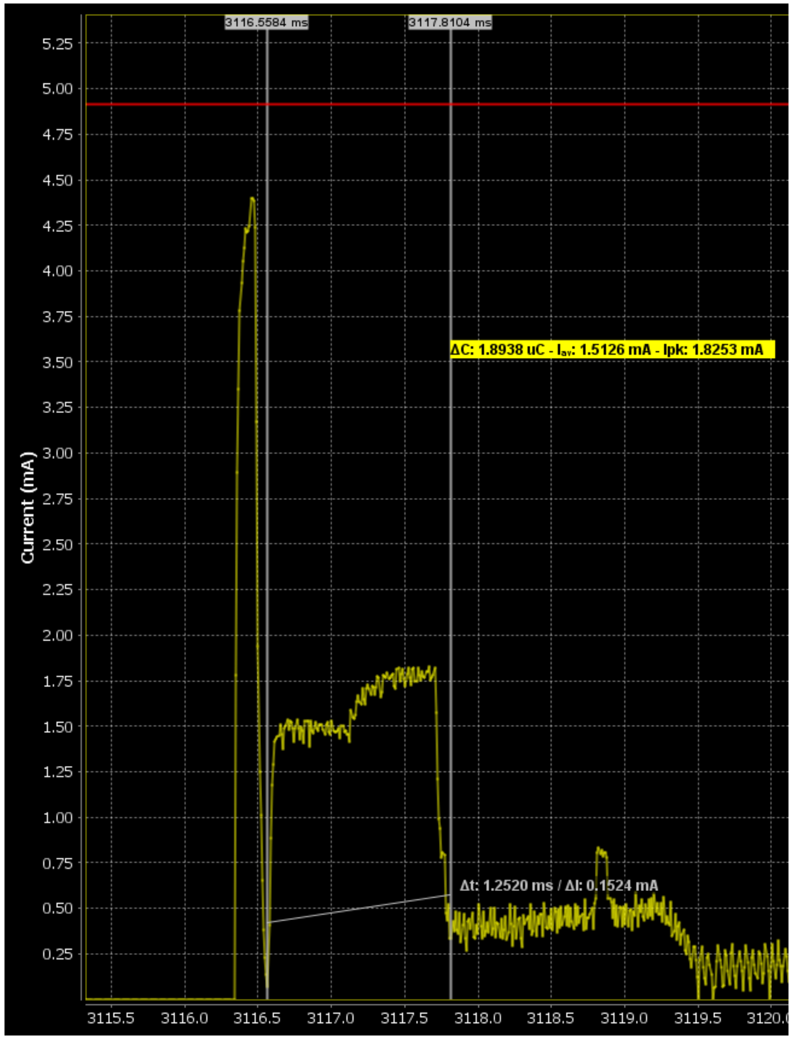


DEEP sleep modes

Setting the DEEP sleep mode (DA14580)

Measurements:

The OTP copy needs ≈ 2 μC



DEEP sleep modes

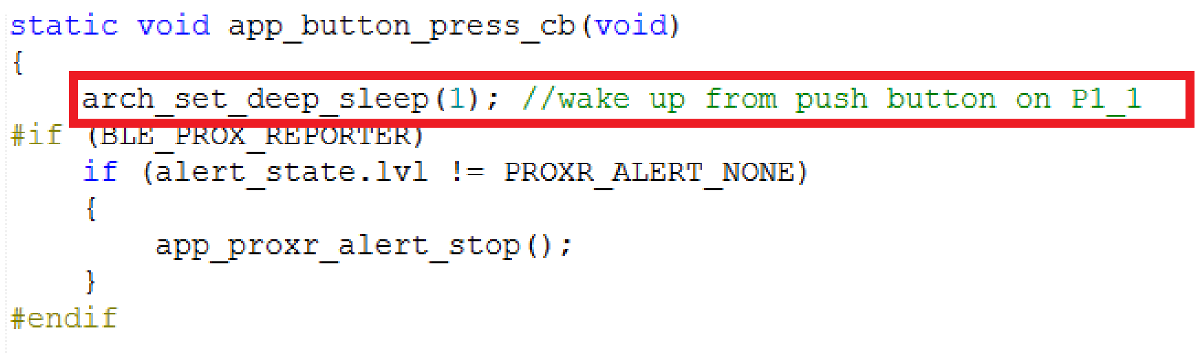
Setting the DEEP sleep mode (DA14585/6)

TODO 1 - open the proximity reporter project from:

projects\target_apps\ble_examples\prox_reporter\Keil_5

TODO 2 – Open the file /* @file **user_proxr.c** */ which is under the user_app folder.

TODO 3 – Add the function arch_set_deep_sleep(1) in the app_button_press_cb(void)



```
static void app_button_press_cb(void)
{
    arch_set_deep_sleep(1); //wake up from push button on P1_1
#if (BLE_PROX_REPORTER)
    if (alert_state.lvl != PROXR_ALERT_NONE)
    {
        app_proxr_alert_stop();
    }
#endif
```

IMPORTANT NOTICE

arch_set_deep_sleep(0): An external interrupt can wake-up the DA14585/6.

arch_set_deep_sleep(1): The POR can wake-up the DA14585/6.

DEEP sleep modes

Measuring the DEEP sleep mode (DA1458x)

`Please follow the same steps as shown earlier.`

DEEP sleep modes

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

For DA14580:

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

For DA14585:

we have measured **571nA** using: arch_set_deep_sleep(1): Interrupt wake-up

we have measured **569nA** using: arch_set_deep_sleep(0): POR wake-up

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

Powering down individual retention memory cells

The Powering down individual retention memory cells can be only be done in EXTENDED sleep mode (DA14580/1/3).

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 with the sleep modes

Conclusion with the sleep modes

CONCLUSION: DA14580: 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	≈ 810 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**

Conclusion with the sleep modes

CONCLUSION: DA14585: Differences between EXTENDED & DEEP sleep modes

	EXTENDED sleep without OTP copy	**EXTENDED sleep with OTP copy**	**DEEP sleep**
Memories switched ON	RAM size of the image + 32 kB RAM (block 4)	32 kB RAM (block 4)	None
Current consumption (BUCK mode, external 32kHz crystal used)	≈ 3.4 μA	≈ 2.8 μA	≈ 571 nA (Ext wakeup) ≈ 569 nA (POR wakeup)
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	OTP content **is copied** into SRAM when boot up from deep sleep

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

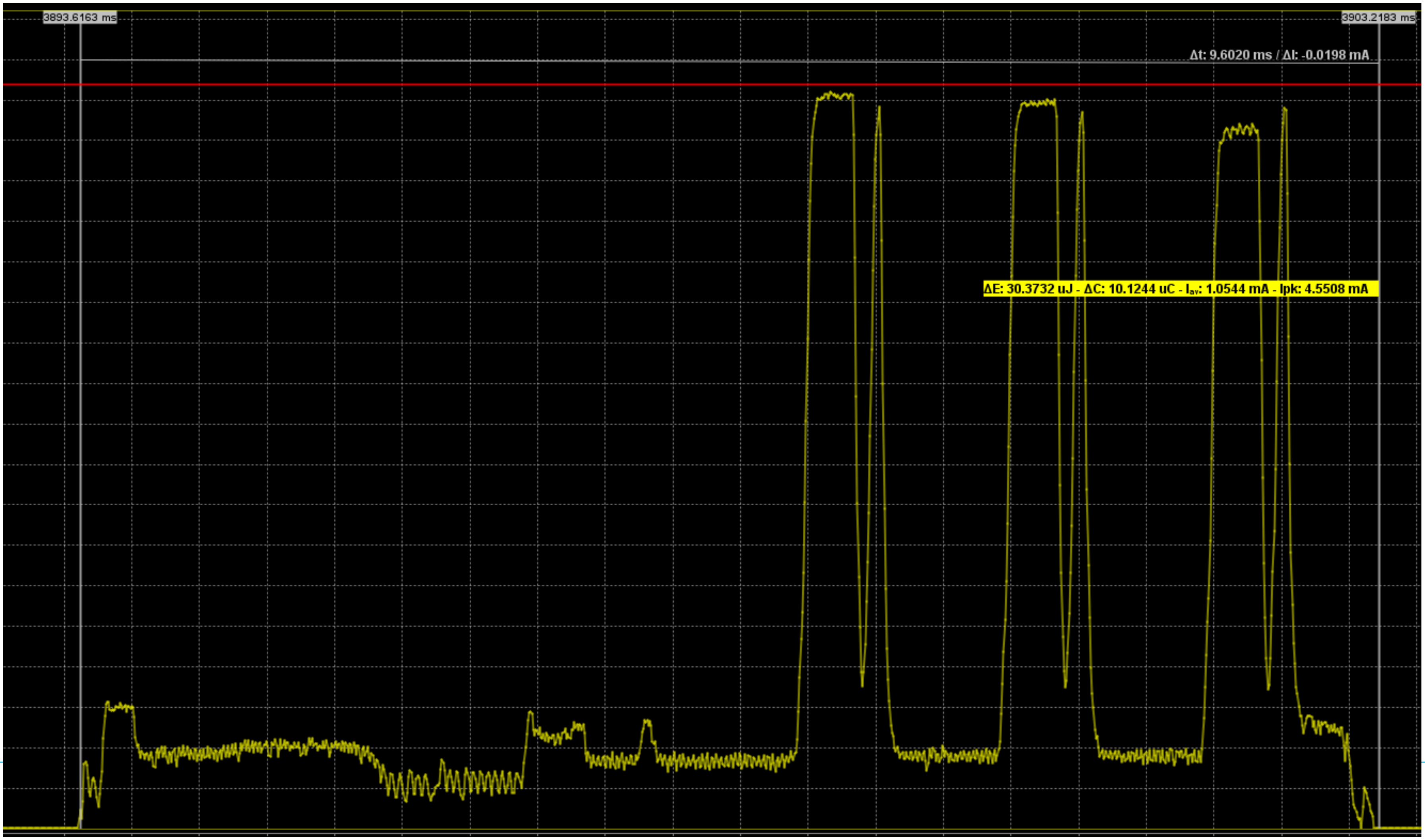
BLE Dialog Semiconductor Sleep modes

Power consumption in active mode

Power consumption in active mode

Power consumption of the Active mode DA14580=30.4uJ

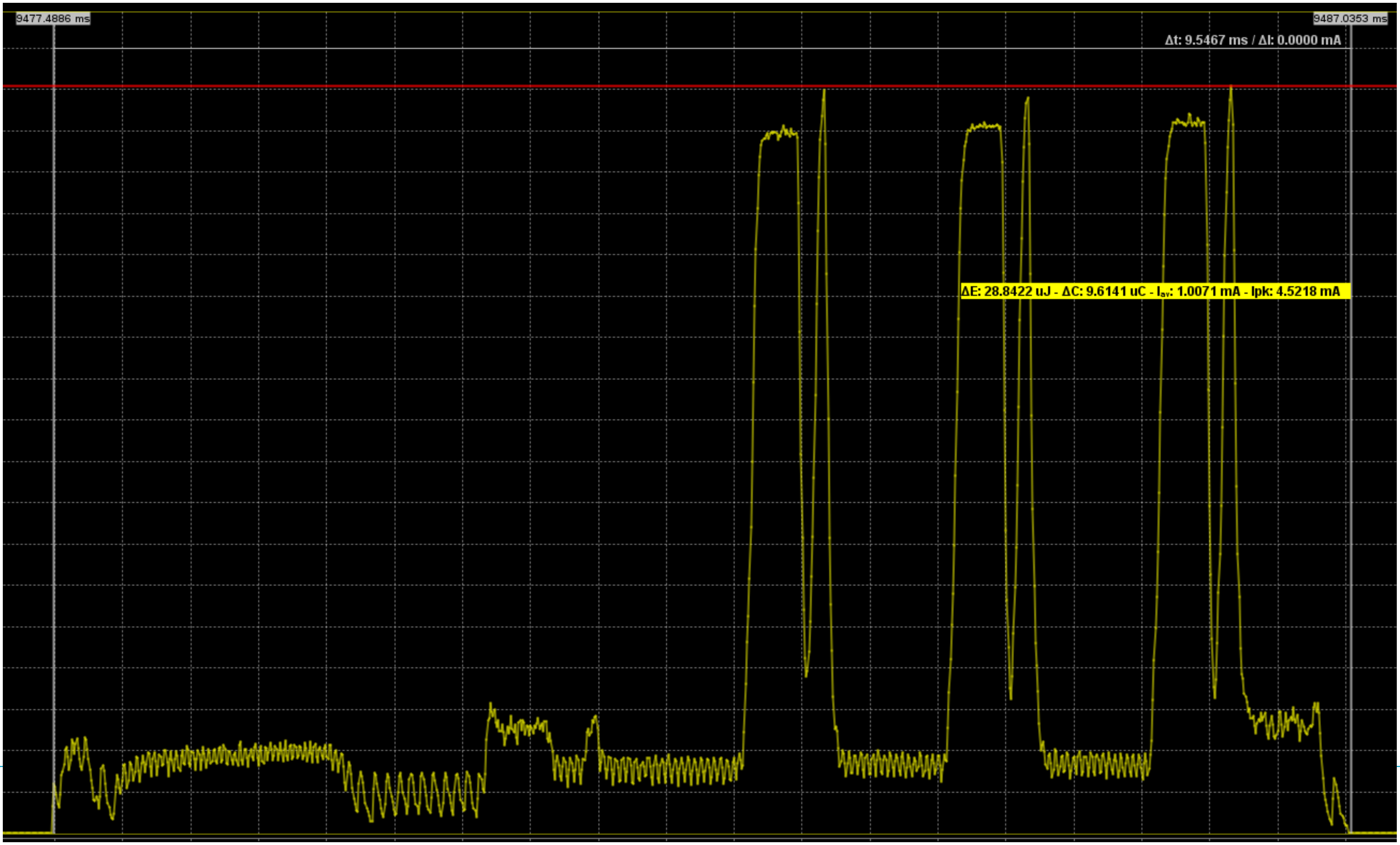
Proximity reporter SW used. Extended sleep mode selected. Output power= 0 dBm. Power supply = 3.3V



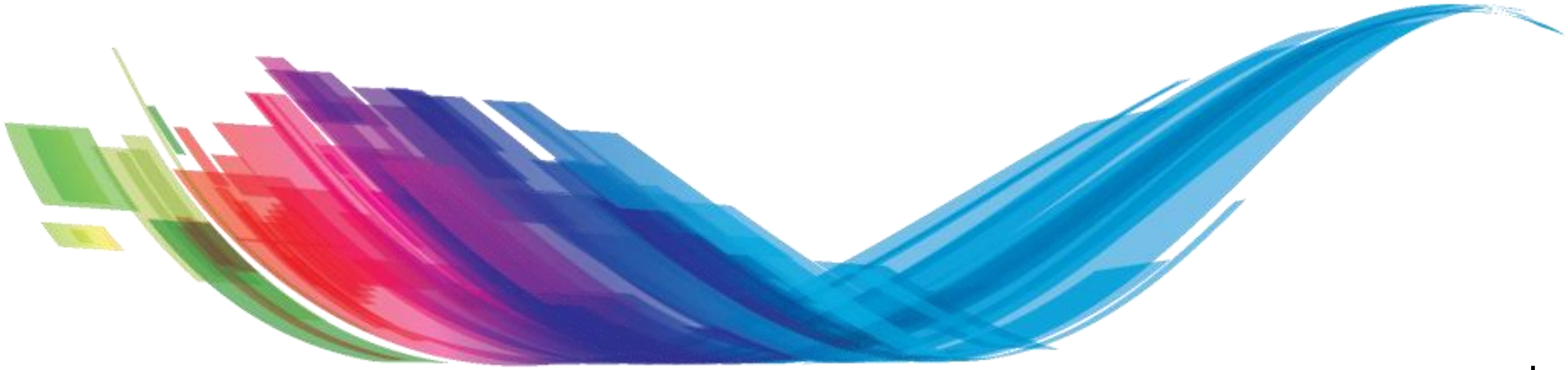
Power consumption in active mode

Power consumption of the Active mode DA14585=28.8uJ

Proximity reporter SW used. Extended sleep mode selected.
Output power= 0 dBm. Power supply = 3.3V



The Power To Be…



…personal
…portable
…connected