**BCM20736S/SDK2.0.1 on EMRF-20736S-BOB:**  Configured for High Adv Interval = 32slots = 32 \* .625ms = 20ms for 10 seconds and Low Adv Interval = 1024slots = 1024 \*.625ms = 640ms. The device will enter DeepSleep if after 30 seconds it has not made a connection OR if after it has connected to a device the connection has been lost. Details are shown below. Board is being supplied 3.3V and is being regulated down to 1.8V to supply BCM20736S. Fine Timer is configured to 500ms.

In my\_App.c I have placed DeepSleep instructions in 2 function calls, **void my\_App\_connection\_down(void)** and void **my\_APP\_advertisement\_stopped(void)** as shown below where the ‘else’ statements are.

**void** **my\_APP\_connection\_down**(**void**)

{

ble\_trace1("handle:%d\n", my\_APP\_connection\_handle);

memset (my\_APP\_remote\_addr, 0, 6);

my\_APP\_connection\_handle = 0;

// If we are configured to stay connected, disconnection was caused by the

// peer, start low advertisements, so that peer can connect when it wakes up.

**if** (my\_APP\_stay\_connected)

{

bleprofile\_Discoverable(*LOW\_UNDIRECTED\_DISCOVERABLE*, my\_APP\_hostinfo.bdaddr);

ble\_trace2("ADV start: %08x%04x\n",

(my\_APP\_hostinfo.bdaddr[5] << 24 ) + (my\_APP\_hostinfo.bdaddr[4] <<16) +

(my\_APP\_hostinfo.bdaddr[3] << 8 ) + my\_APP\_hostinfo.bdaddr[2],

(my\_APP\_hostinfo.bdaddr[1] << 8 ) + my\_APP\_hostinfo.bdaddr[0]);

}

**else**

{

ble\_trace0("Entering DeepSleep - Connection Lost \n");

bleapputils\_delayUs(500);

// devlpm\_enterLowPowerMode(); //Puts device into normal Sleep Mode ~37uA

bleprofile\_PrepareHidOff(); //Puts device into Deep Sleep ~1.33uA

}

}

**void** **my\_APP\_advertisement\_stopped**(**void**)

{

ble\_trace0("ADV stop!!!!");

ble\_trace1("my\_APP\_stay\_connected = %d\n", my\_APP\_stay\_connected);

// If we are configured to stay connected, disconnection was caused by the

// peer, start low advertisements, so that peer can connect when it wakes up.

**if** (my\_APP\_stay\_connected)

{

bleprofile\_Discoverable(*LOW\_UNDIRECTED\_DISCOVERABLE*, my\_APP\_hostinfo.bdaddr);

ble\_trace2("ADV start: %08x%04x\n",

(my\_APP\_hostinfo.bdaddr[5] << 24 ) + (my\_APP\_hostinfo.bdaddr[4] <<16) +

(my\_APP\_hostinfo.bdaddr[3] << 8 ) + my\_APP\_hostinfo.bdaddr[2],

(my\_APP\_hostinfo.bdaddr[1] << 8 ) + my\_APP\_hostinfo.bdaddr[0]);

}

**else**

{

ble\_trace0("Entering DeepSleep - AdvStopped \n");

bleapputils\_delayUs(500);

bleprofile\_Discoverable(*NO\_DISCOVERABLE*, NULL);

// devlpm\_enterLowPowerMode(); //Puts device into normal Sleep Mode ~37uA

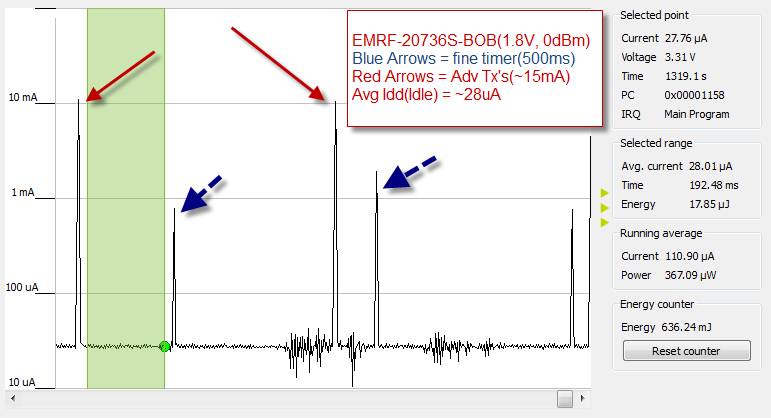
bleprofile\_PrepareHidOff(); //Puts device into DeepSleep ~1.33uA

}

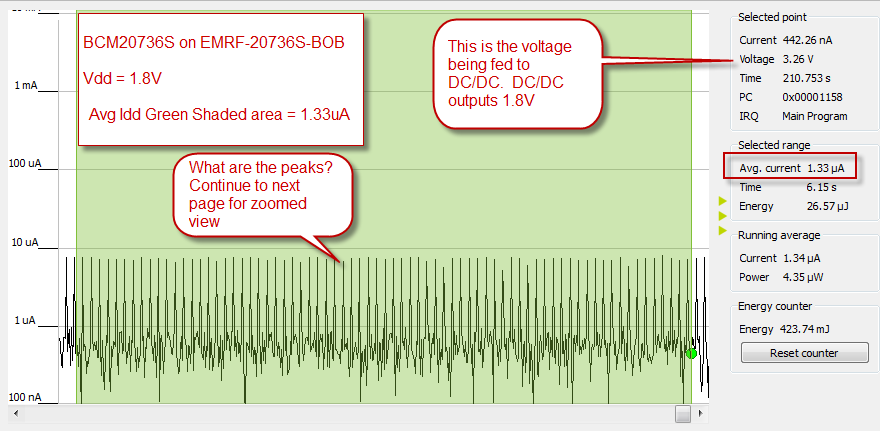
}

During **low\_undirect\_adv\_interval** which is set to 1024 slots = 1024 \* .625ms = 640ms

We are looking at this waveform vs. the high\_undirect\_adv\_interval as we can see the ‘Idle’ current much clearer since the Adv intervals are more spread out. Keep in mind as long as Sleep is enabled in the BCM20736A1.cgs file the device will automatically go into Sleep/Idle mode when it can such as in between Advertising events.



**HIDOff:** Device will enter HIDOff after it has Advertised for >30seconds and no connection OR after it has connected to a device and the connection has been lost. This is how the my\_App.c firmware works which is a simple modified version of the hello\_sensor app in SDK2.0.1. Device will wake-up after pressing a GPIO that has been defined as an Interrupt, such as P0 which is defined in platform.h as the Push-button.



**HIDOff Peaks:** Zoomed in view to see pulses that are occurring every ~82ms with peaks of >120uA’s.

