

Optimizing Power on nRF52 Designs



[notsane0](#) 30 May 2018

As customers reach the end of their design cycle, their attention generally falls on power optimization, especially for battery powered designs. The following are hints for achieving the lowest possible power in your design.

System

1. By enabling and populating the DC/DC converter, the current consumption of your product will be much lower. Depending on your design, you can see as high as a 35% reduction in power usage. The higher the duty cycle is, the greater the savings will be from the DC/DC converter.
2. 2-10% power savings can be achieved by enabling/populating the 32kHz LFCLK crystal. In very low duty cycle applications, like a long-interval beacon, the savings may be much greater.
3. Make sure to provide a way to measure nRF52 current. If the nRF52 power is combined with other circuitry, you will not be able to measure it independently and get meaningful results.
4. Pick the right battery for your design. nRF52832 pulse currents are 7mA (with DCDC) and 12mA (w/o DCDC). Proper decoupling can mitigate the effects of these pulses on your battery, but high pulse currents and low temperatures can significantly reduce a coin cell's overall capacity.
5. Review the device errata and note what applies to your design's power consumption. This may influence your serial port or other peripheral choices.
6. The nRF52 devices are optimized for running at 3V (although the '840 will accept 5V). None of the internal circuitry runs at these voltages except the GPIO, so running at lower voltages will not result in reduced power consumption.
7. Only use RAM_RETENTION for the memory required to be retained. Using more draws unnecessary power.
8. There are only two power modes on the nRF52: SYSTEM_ON and SYSTEM_OFF.
 - a. SYSTEM_OFF: This state draws 300nA (nRF52832) or 400nA (nRF52840) and can be woken by a reset, GPIO interrupt or NFC Field (add 100nA). The wake state is as if a reset has occurred. See



the reset behavior table in the ['832 product spec](#) and ['840 product spec](#).

- b. SYSTEM_ON: There are Constant latency and Low power sub-modes. In the default Low power sub-mode, normally the lowest power is 1.9uA (nRF52832) or 1.5uA(nRF52840) and includes the LFCLK and RTC. This is the normal state between BLE connection events. CPU restarts on timer, peripheral or pin interrupts. For more information on the sub-modes, see the product spec.

Hardware

1. Be aware that single-pin GPIOTE interrupts may use more power than Port GPIOTE interrupts depending on the scenario.
2. The default (from reset) state of all GPIO is as a disconnected input. Unless you've previously configured them to something else, you can leave unused pins in their default state and physically disconnected to draw no current.
3. At 0dBm with the DC/DC on, the nRF52832 transmitter draws 5.3mA. Increasing the TX power to +4dBm adds only 2.2mA. Decreasing it to -40 dBm saves only 2.6mA.
4. Holding the reset pin low is not a low power state
5. Consider adding a GPIO controlled power switch (ex. high side p-channel MOSFET) for sensors that have high leakage current in order to reduce SYSTEM_OFF current

Software

1. Note the `power_manage()` or `sd_app_evt_wait()` call inside the `main.c` `while(1)` loop in the SDK examples. Running this instruction addresses pending interrupts and race conditions and allows the CPU to be properly idled. If your code does not reach this instruction, the CPU will remain ON and drawing power.
2. Use app timers rather than software delays where possible. Software delay loops will keep the CPU running and drawing power.

Debugging

1. The nRF52 may remain in debug interface (DIF) mode if the SWD debugger connection is not disconnected properly. When the device is in debug interface mode, it will draw additional current since  it keeps a certain set of peripherals ON at all times. Performing a pin reset, given pin reset has been

enabled on the part and it is accessible, is a common way of making sure DIF mode is not enabled.

2. Use the right tool to measure your current. Use the [online power profiler](#) and Product Spec numbers to estimate your average current. Use the [PPK2](#) or another [correct method](#) to measure it. Be aware that PPK measurements will be inaccurate at very, very low static current levels.
3. Make sure to power your design with a battery when measuring current, since a power supply may introduce unknown variables.
4. Here's a list of the most common power scenarios and how to recognize them.
 - a. **You measure the current at about 4mA.** The CPU is running ... either because you're using it intentionally or that you haven't run the `power_manage()`; That instruction (rather, the ones contained in its wrapper) are the ones that shut down the CPU.
 - b. **You think the part is in SYSTEM_OFF mode, but the part is drawing much more than a few hundred nA.** The part isn't actually in SYSTEM_OFF mode and something has been left running ... the CPU, a peripheral. Make sure your shutdown procedure actually turns everything off. Cycle power to make sure the SWD port isn't inadvertently on. Another common issue is being in debug interface (DIF) mode, either by having the debugger connected or not power cycling after programming.
 - c. **You think you're in SYSTEM_ON mode with the CPU off but you measure 100's of uA.** You've left a peripheral on, maybe intentionally. If you left logging enabled through the UART, this will also use this amount of current. Turn off UART logging in the `sdk_config.h` file in favor of RTT. RTT will not use current unless the debugger is connected.
 - d. **You're measuring 10's of mA.** It's nearly impossible for the nRF52 to draw this much current. It's almost certain that you're measuring current for surrounding circuitry as well.



[JohnMeg](#) *over 4 years ago*

Thank you, this is a good article. I ran into these requirements few months ago. Actually the main blocker I find out preventing me to go to SYSTEM_OFF mode is the easyDMA. If a peripheral is using the **DMA you cannot go to System OFF**. In my case it was related to the PWM.

9 comments 1 member is here



[helloGodNodic](#) *over 4 years ago*

thanks! Simple and useful.



[modles](#) *over 3 years ago*