nRF52840 Dongle Notes

Some notes on programming the nRF52840 Dongle.

Assumptions:

- using the S140 SoftDevice.
- using the nRF5 SDK 15.2.
- use nRF Connect for programming.

http://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.softdevices52%2Fdita%2Fsoftdevices%2Fs140%2Fs140.html

https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.sdk52.v0.9.2%2Findex.html

https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF-Connect-for-Desktop

Step-by-step guide for Building and Flashing

Building and flashing an SDK example.

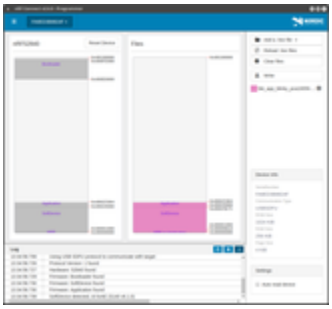
There appear to be only two `ble_peripheral` examples for `pca10059`, the dongle, so we'll go with `ble_app_blinky`.

1. `cd ~/Development/nRF_SDK_15.2.0_9412b96/examples/ble_peripheral/ble_app_blinky/pca10059/s140/armgcc`
2. `make`

```
knud 10:40:57 $make
mkdir _build
cd _build && mkdir nrf52840_xxaa
Assembling file: gcc_startup_nrf52840.S
Compiling file: nrf_log_backend_rtt.c
Compiling file: nrf_log_backend_serial.c
Compiling file: nrf_log_backend_uart.c
Compiling file: nrf_log_default_backends.c
Compiling file: nrf_log_frontend.c
Compiling file: nrf_log_str_formatter.c
Compiling file: app_button.c
Compiling file: app_error.c
Compiling file: app_error_handler_gcc.c
Compiling file: app_error_weak.c
Compiling file: app_scheduler.c
Compiling file: app_timer.c
Compiling file: app_util_platform.c
Compiling file: hardfault_implementation.c
Compiling file: nrf_assert.c
Compiling file: nrf_atfifo.c
Compiling file: nrf_atflags.c
Compiling file: nrf_atomic.c
Compiling file: nrf_balloc.c
Compiling file: nrf_fprintf.c
Compiling file: nrf_fprintf_format.c
Compiling file: nrf_memobj.c
Compiling file: nrf_pwr_mgmt.c
Compiling file: nrf_ringbuf.c
Compiling file: nrf_section_iter.c
Compiling file: nrf_strerror.c
Compiling file: system_nrf52840.c
Compiling file: boards.c
```

```
Compiling file: nrf_drv_clock.c
Compiling file: nrf_drv_uart.c
Compiling file: nrfx_clock.c
Compiling file: nrfx_gpiote.c
Compiling file: nrfx_power_clock.c
Compiling file: nrfx_prs.c
Compiling file: nrfx_uart.c
Compiling file: nrfx_uarte.c
Compiling file: main.c
Compiling file: SEGGER_RTT.c
Compiling file: SEGGER_RTT_Syscalls_GCC.c
Compiling file: SEGGER_RTT_printf.c
Compiling file: ble_advdata.c
Compiling file: ble_conn_params.c
Compiling file: ble_conn_state.c
Compiling file: ble_srv_common.c
Compiling file: nrf_ble_gatt.c
Compiling file: nrf_ble_qwr.c
Compiling file: utf.c
Compiling file: ble_lbs.c
Compiling file: nrf_sdh.c
Compiling file: nrf_sdh_ble.c
Compiling file: nrf_sdh_soc.c
Linking target: _build/nrf52840_xxaa.out
   text    data     bss     dec     hex
filename
  26096     532    2476   29104    71b0
_build/nrf52840_xxaa.out
Preparing: _build/nrf52840_xxaa.hex
Preparing: _build/nrf52840_xxaa.bin
DONE nrf52840_xxaa
```

3. `mergehex -m _build/nrf52840_xxaa.hex ~/Development/nRF5_SDK_15.2.0_9412b96/components/softdevice/s140/hex/s140_nrf52_6.1.0_softdevice.hex -o ble_app_blinky_pca10059.hex`
4. Start `nRF Connect` and launch the `Programmer` application.
5. Select the device (Dongle)
    a. You may need to reset it manually using its reset push button to get it to show up in `nRF Connect`
6. Add the .hex file from step 3
7. The `nRF Connect` window should resemble



8. Click `Write`
9. The Dongle should now be programmed

Related articles

- nRF52840 Dongle Notes
- Standalone Duo Build Environment