



Nordic Tools

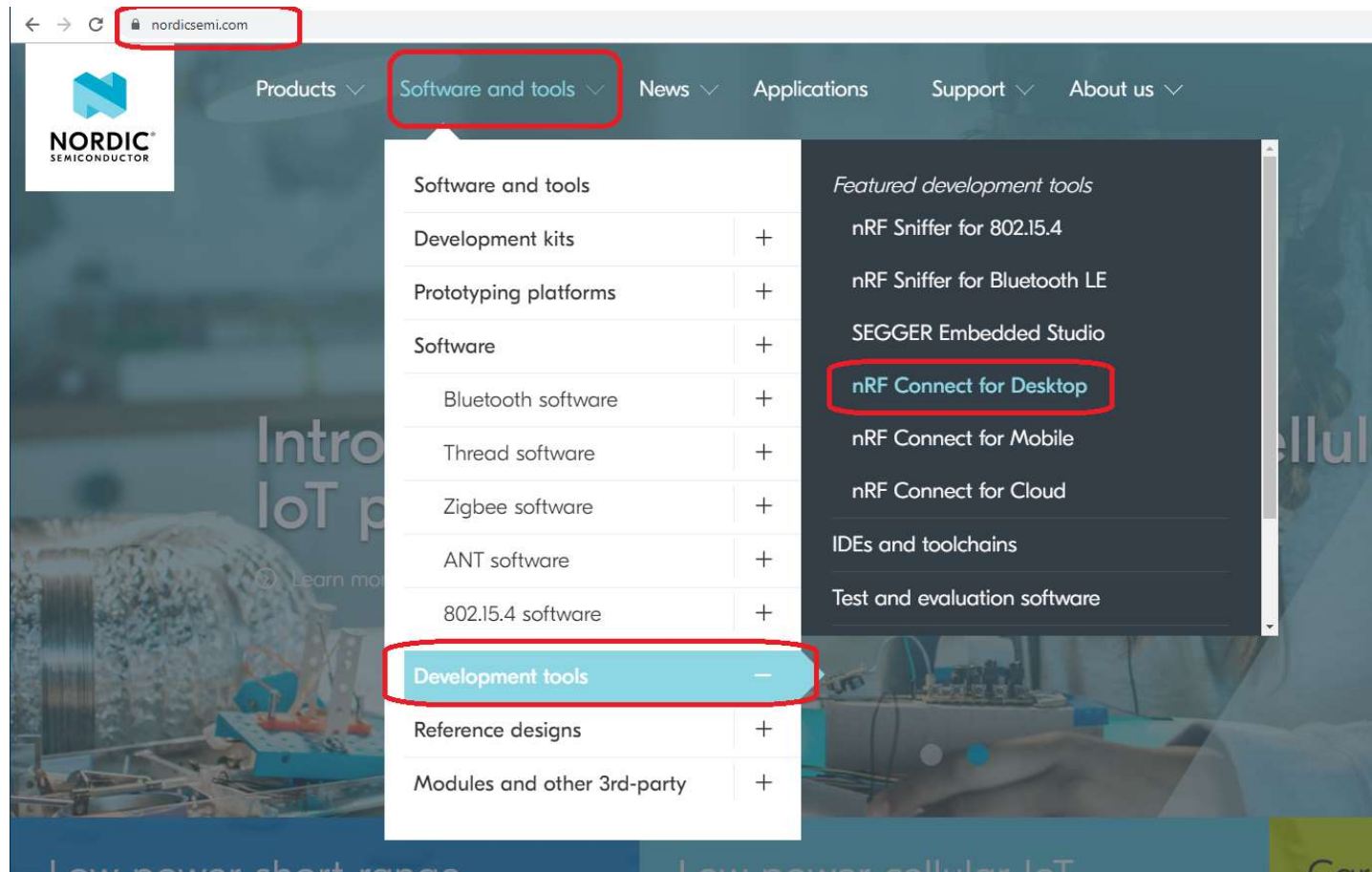
Introduction to the different tools used for development on Nordic's BLE solutions.

Edvin Holmseth

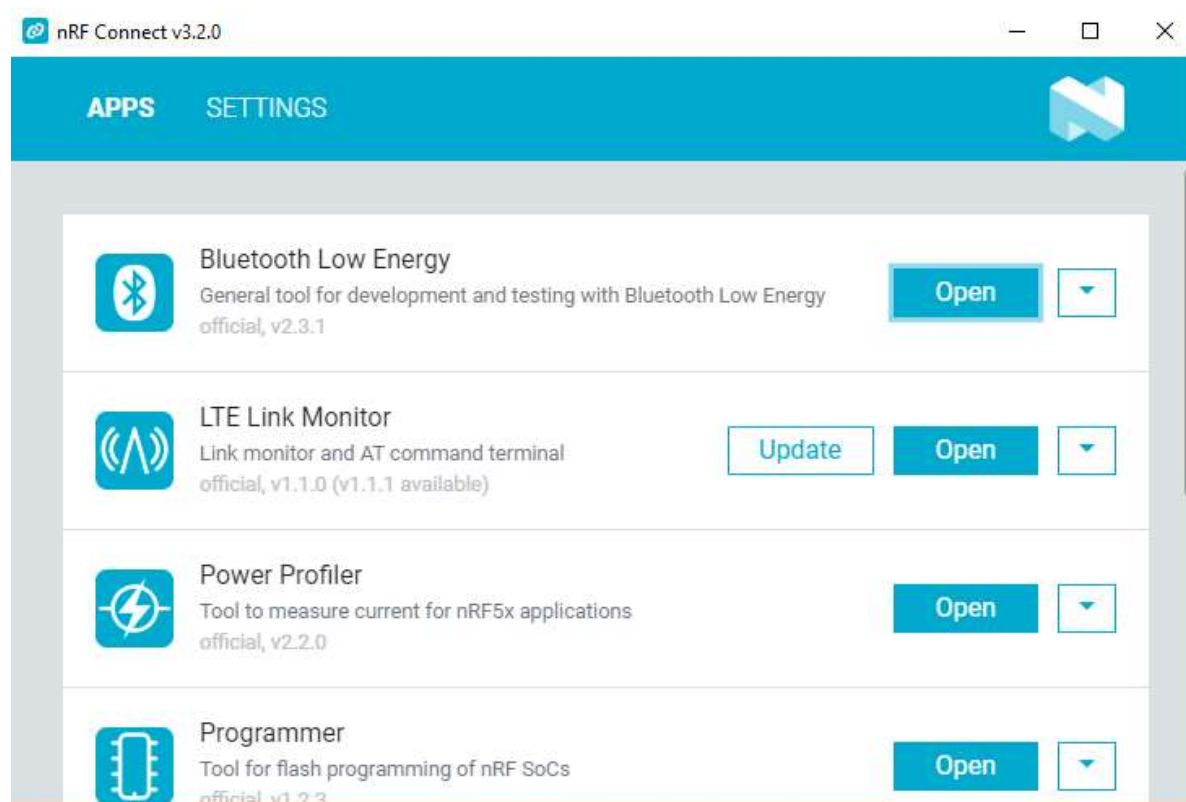
Uni. Of Strathclyde

January 2020

nRF Connect for Desktop



nRF Connect for Desktop





Select device ▾



Connection Map

Server Setup



000960068019 PCA10090

- Serial port: COM14
- Serial port: COM15
- Serial port: COM16

Discovered devices

▶ Start scan

Clear

Options

Log



11:41:40.377 Application data folder: C:\Users\edho\AppData\Roaming\nrfconnect\pc-nrfconnect-ble

nRF Connect for Desktop

1. Compile the example found in:
 `SDK15.3.0\examples\ble_peripheral\ble_app_uart\pca10040
 \s132\ses\`
2. Press “Build”->”Build and Debug”
3. Press Green arrow (continue execution)
4. Press scan start in nRF Connect
5. Connect to the device called “Nordic_UART”

nRF5x

E2:74:A1:0E:97:52

Adapter

Generic Access

Generic Attribute

Nordic_UART

C4:34:FE:E0:6C:D3

Peripheral

Generic Access

Generic Attribute

UART over BLE

UART RX

writeWoResp write

UART TX

notify

Discovered devices

Start scan

Clear

Options

Nordic_UART

-26 dBm

C4:34:FE:E0:6C:D3

Connect

<Unknown name>

-33 dBm

38:F3:80:0B:B7:8A

Connect

Zigbee_Door_Lock

-42 dBm

FD:70:5E:50:5A:73

Connect

<Unknown name>

-46 dBm

2E:F6:41:B9:CC:1A

Connect

<Unknown name>

-46 dBm

20:9A:4C:55:84:ED

Connect

<Unknown name>

-54 dBm

28:49:59:FE:7A:6E

Connect

<Unknown name>

-42 dBm

67:35:E8:E1:05:AA

Connect

<Unknown name>

-51 dBm

11:C3:E9:3F:64:E7

Connect

<Unknown name>

-44 dBm

08:CA:3F:BD:B8:65

Connect

<Unknown name>

-64 dBm

Connect

Log

12:33:43.230	Connecting to device
12:33:45.873	Connected to device C4:34:FE:E0:6C:D3
12:33:46.011	ATT MTU changed, new value is 247
12:33:46.879	Attribute value read, handle: 0x03, value (0x): 4E-6F-72-64-69-63-5F-55-41-52-54
12:33:50.422	Connection parameters updated for device C4:34:FE:E0:6C:D3: interval 20ms, timeout 4000ms, latency: 0
12:33:51.733	Attribute value changed, handle: 0x10, value (0x): 01-00
12:33:51.741	Attribute value written, handle: 0x10, value (0x): 01-00

nRF Connect for Desktop

1. Open a UART Terminal.
PuTTY or Termite are good options
2. Write «Hello» and press send.
3. Monitor that the UART TX characteristic says:



48:65:6C:6C:6F:0D
'H' 'E' 'L' 'L' 'O' '\n'

Is ASCII code for:

nRF Connect for Mobile

- A Mobile version of nRF Connect for Desktop
- Works for Android and Apple
 - Can be found in Google Play Store and App Store

nRF Connect for Mobile

- A Mobile version of nRF Connect for Desktop
- Works for Android and Apple
 - Can be found in Google Play Store and App Store

Logging module in Nordic's SDK

- Most of the examples in the SDK comes with the logging module enabled.
- When you debug, you can monitor the logging near the bottom of the screen

Debugging Mode

ble_app_uart_pca10040_s132 - SEGGER Embedded Studio V3.30 (64-bit) - Licensed to Nordic Semiconductor - Justin Lee (Running)

File Edit View Search Navigate Project Build Debug Target Tools Window Help

Disassembly main.c Running...

Go/Stop

Breakpoints

Breakpoint Line

Registers

Register

Call Stack

Call Stack

Debug Terminal

RTT Terminal

```
main
0002656C 4632 mov r2, r6
0002656E 4641 mov r1, r8
00026570 F7DF99E bl 0x00023880 <
00026574 4630 mov r0, r6
00026576 F7DFBC7 bl 0x00023D08 <
0002657A 3401 adds r4, #1
0002657C E7DC b 0x00026538
0002657E E8BD81F0 pop.w {r4-r8, p
00026582 BF00 nop
00026584 20002A68 .word 0x20002A6
00026586 0002939F .word 0x0002939
<gatt_init>
0002658C B508 push {r3, lr}
0002658E 4908 ldr r1, =0x0002
00026590 4808 ldr r0, =0x2000
00026592 F000FFED bl 0x00027570 <
00026596 B108 cbz r0, 0x00026
00026598 F7DFB86 bl 0x00023D08 <
0002659C 2148 movs r1, #0x40
0002659E 4805 ldr r0, =0x2000
000265A0 F000FFFA bl 0x00027598 <
000265A4 B118 cbz r0, 0x00026
000265A6 E8BD4008 pop.w {r3, lr}
000265AA F7DFBAD pop {r3, pc}
000265AE B008
000265B0 00026309 .word 0x000263D
000265B2 200034E8 .word 0x200034E
<main>
000265B8 B530 push {r4-r5, lr}
000265BA B0AD sub sp, sp, #0x
000265BC F7DFE5E bl 0x0002427C <
000265C0 B108 cbz r0, 0x00026
000265C2 F7DFBA1 bl 0x00023D08 <
000265C6 4D60 ldr r5, =0x0002
000265C8 CDF0 ldr r5!, {r0-r3
000265CA AC04 add r4, sp, #16
000265CC C40F stm r4!, {r0-r3
000265CE E8950003 ldr r5, {r0-r1}
000265D0 4B68 ldr r3, =0x2000
000265D4 4A68 ldr r2, =0x2000
000265D6 9300 str r3, [sp]
000265D8 F44F7300 mov.w r3, #0x10
000265DC E8B40003 stm r4, {r0-r1}
000265E0 9301 str r3, [sp, #4
000265E2 9202 str r2, [sp, #8
000265E4 9303 str r3, [sp, #1
000265E6 4A68 ldr r2, =0x0002
000265E8 2107 movs r3, #7
000265EA 4669 mov r1, sp
000265EC A804 add r0, sp, #16
000265EE F7DFF41 bl 0x00024474 <
000265F0 2000 .word 0x2000
000265F2 B108 cbz r0, 0x00026
000265F4 F7DFB88 bl 0x00023D08 <
000265F6 2000 .word 0x2000
000265F8 F7CFFA9 bl 0x00023550 <
000265FE B108 cbz r0, 0x00026

// Initialize.
err_code = app_timer_init();
APP_ERROR_CHECK(err_code);

uart_init();
log_init();

buttons_leds_init(&erase_bonds);
ble_stack_init();
gap_params_init();
gatt_init();
services_init();
advertising_init();
conn_params_init();

printf("\n\nUART Start!\n\n");
NRF_LOG_INFO("UART Start!");
err_code = ble_advertising_start(&advertising, BLE_ADV_MODE_FAST);
APP_ERROR_CHECK(err_code);

// Enter main Loop.
for (;;)
{
    UNUSED_RETURN_VALUE(NRF_LOG_PROCESS());
    power_manage();
}
```

main.c, line 701

V7M Exceptions

- BusFault
- ExceptionEntryReturnFault
- HardFault
- MemManage
- UsageFault_Coprocessor
- UsageFault_StateError

Groups

Name Value

CPU - Current Context

Name	Value
r0	0x00000000
r1	0x00000000
r2	0x000265b9
r3	0x00000000
r4	0x10001000
r5	0x00000000
r6	0x00000000
r7	0x00000000
r8	0x00000000
r9	0x00000000
r10	0x00000000
r11	0x00000000
r12	0x20010000
sp(r13)	0x20010000
lr(r14)	0x00023193
pc(r15)	0x000265b8
apsr	0x61000000

Function

Call Address

Running...

<warning> nrf_sdh ble: RAM starts at 0x20002A68, can be adjusted to 0x200027C0

<warning> nrf_sdh ble: RAM size can be adjusted to 0x0000

<info> app: UART Start!



Nordic Tools

Introduction to the different tools used for development on Nordic's BLE solutions.

Edvin Holmseth

Uni. Of Strathclyde

January 2020