Introduction

The example runs on Nordic nRF52832 SoC. The block diagram of this example is as follows:



In this example, nRF52832 acts as a forwarder, to forward command packets and return packets between the NSP32 module and the Android phone. The Android app can wirelessly (through bluetooth) control NSP32 module and get the spectrum data.

Note: This example must run in conjunction with the "Android SpectrumMeter" APK installed on an Android phone. Please check "NSP32 Java API for Android / desktop" for getting this Android project and APK.

API Source File Location

- The API source files are located under [/examples/nRF52/SpectrumMeter/NanoLambdaNSP32/], along with nRF52Adaptor.h and nRF52Adaptor.cpp tailored for nRF52.
- 2) NSP32CWrapper.h and NSP32CWrapper.c are also provided under the same folder, to enable NSP32 API being called from C source code (i.e. main.c).

Hardware Setup

- 1) Tested on
 - nRF52832

- 2) Setup
 - (a) using NSP32m DBK Development Board Kit with J3 disconnected.



(b) or manually configure your NSP32m with your nRF52 in the following way:



Table below is the pin connections between NSP32 and nRF52832 (on PCA10040 Development Kit board).

| GPIO | Hardware | NSP32 Pin | nRF52832 |
|-------|----------|-----------|----------|
| Power | VDD | VDD3V3 | VDD |
| | GND | GND | GND |

| SPI Signal | Wakeup/Reset | RST | P0.16 |
|------------|--------------|-------|-------|
| | SPI SSEL | SS | P0.22 |
| | SPI MOSI | MOSI | P0.23 |
| | SPI MISO | MISO | P0.24 |
| | SPI SCK | SCK | P0.25 |
| | Ready | Ready | P0.14 |

Status LED

Connect nRF52832 P0.18 to a resistor and a LED as follows.



Pre-built hex

- A pre-built hex file is located at [/examples/nRF52/SpectrumMeter/pca10040/s132/arm5_no_packs/_build/nrf528 32_xxaa.hex].
- 2) You can program the hex to nRF52832 by flashers (e.g. nRFgo or J-Flash).
- 3) The example requires Nordic SoftDevice present on nRF52832. Make sure you program the "S132 v6.1.0 SoftDevice (provided with nRF SDK 15.2.0)" before hand.

Software Setup

1) Runs on

Keil uVision5

2) Setup

If you need to rebuild the hex file, or want to modify the example code, you can follow these steps:

- 1. Install Keil5 IDE.
- 2. Download "nRF SDK 15.2.0" and extract it to your hard drive. Say the extracted path is {SDK}.
- Put the [/examples/nRF52/SpectrumMeter] folder under [{SDK}/examples/ble_peripheral/].
- 4. Open [{SDK}/examples/ble_peripheral/SpectrumMeter/pca10040/s132/arm5_no_p

acks/image_transfer_demo_pca10040_s132.uvprojx] in Keil5.

 Build the project, and you will get the hex file at [{SDK}/examples/ble_peripheral/SpectrumMeter/pca10040/s132/arm5_no_p acks/_build/nrf52832_xxaa.hex].

Project Disclamation

This example is modified from a demo project from Nordic, which could be found at [https://github.com/NordicPlayground/nrf52-ble-image-transfer-demo]. We try to do minimum modifications and keep the original codes as much as possible.

Run the Example

To run the example, program the hex file and Nordic SoftDevice to nRF52832. Then power on.

- When BLE is advertising, you shall see the nRF status LED blinking.
- After Android app connects nRF, the nRF status LED will keep lighting.
- If the advertising times out (i.e., Android app does not connect within 180 seconds), nRF will go system-off.

Note: Users need to wake it up through the reset pin or use "power-off \rightarrow power-on" procedure.

Appendix

nRF52832 programming guide (using nRFgo studio)

Hardware interface: SEGGER J-LINK SWD

Software: nRFgo studio

| Step1. P | rogram 'SoftDev | vice' (.hex file) | Step2. Program 'Application' (.hex file) | | |
|---|--|--|--|---|--|
| 🛤 nRFgo Studio | | - D X | 🛤 nRFgo Studio | | - 🗆 X |
| File View nRF8001 Setup Help | | | File View nRF8001 Setup Help | | |
| Feature X * 2.4 GHz * Front-End Tests TX carrier wave output RX constant carrier/LD leakage | SBOGER to use: 22410034 v Retroit | 2 | Feature X * 2.4 GHz * From-End Tests TXC carrier wave output RXC constant carrier/LO leakage | SEGGER to use: 29480924 * Refault | 2 |
| TX/RX channel sweep RX sensitivity W Bluetooth nRF8001 Configuration Dispatcher | Region 1 (Application) | Program.2010evice Program.Applantion. Program.Boofcoder Programming of SoftDevice on aRF5x device Prot program. (2,a152,6.1.0,a0Mevice.hex) Brown. | TX/RX channel sweep RX sensitivity W Bluetooth nRF8001 Configuration Dispatcher | Region 1 (Application) Size . 100 MB | Program.SofDevice Program Application Program.Boodooder Programming of Application on nRF5x device File to program ProclaminEFInetTSR72_com.lex Brown. |
| Trace Translator Direct Test Mode #RF8002 | Address: 0x20000 | DefDevice nice (AD): 152 C DefDevice nice (AD): 152 C Enable DefDevice posterion (UICR CLENE): | Trace Translator Direct Test Mode NRF8002 | Address: 0x26000 | Lock value chy from rescheck |
| Device Meeger X Notherbacks INF5x Programming nRF24LU1+ Bootloaders | Region (CSchDawice) Bue: 132 AB Pitmware Unbuewn (H. Bollind) | Traty Real | Device Mesager × Motherboards INPSs Programming INPS4LU1+ Booffoaders | Region O (SoftDurice) Sile: 152 MB Permenen: Unknown Od Bullhar) | Tropus Tudy Red |
| | Erns el | | | Eros al |] |

Programming nRF52832 via J-LINK



Android API example

