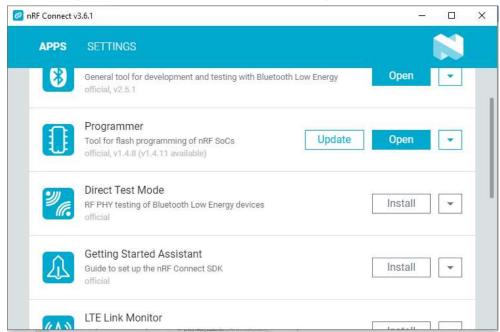
1. Tools

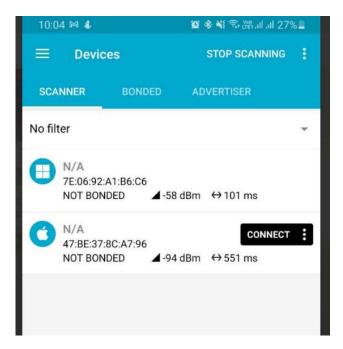
1.1. Tools for nrf code

- Download and install Segger Embedded studio v4.12
 https://www.segger.com/downloads/embedded-studio (Default installation path: C:\Program Files\SEGGER\SEGGER Embedded Studio for ARM 4.12)
- Request a free license for SES http://license.segger.com/Nordic.cgi
- Down load NRF connect tools https://www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Connect-for-desktop



1.2. Tools mobile app

Nrf Connect (in both android and IOS)



1.3. SDK for nrf and soft Device

Download platform s132

 $\underline{https://www.nordicsemi.com/Software-and-tools/Software/nRF5-SDK/Download}$

Remember include "Soft Device": S132



1.4. Hardware tools

- Using PCA10040 development kit
- Debugger CP2103

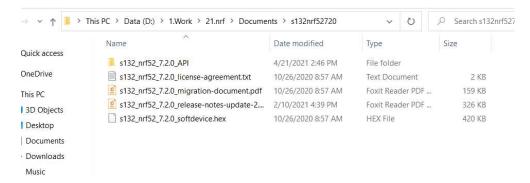
```
workspace_1.6.0 - Flyx_CppTest/Core/user_driver/sensor/sensor_knock.c - STM32CubelDE
 File Edit Source Refactor Navigate Search Project Run Window Help
[ ] ▼ [ ] [ ] ♥ ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼ [ ] ▼
Project Explorer 

□ □ S □ □ □ □ ble_com.c □ uart_com.c □ main.c □ portmacro.h □ app_cli.c
       Assignment2
                                                                                                                                     case 1: // read channel 1
       DWS1000_CubeMX (in DWS1000_exam
                                                                                                                                                if (timer get tick() - knock sensor tick count > 3
   > Flyx_CppTest
  poskat (in poskat_cubeIDE)
                                                                                                                                                              knock sensor state = 2;
        > 🗱 Binaries
                                                                                                                                                              knock_sensor_tick_count = timer_get_tick();
        > 🛍 Includes
                                                                                                                                                              // read sensor channel 1
        > 🐸 Core
                                                                                                                                                             sensor_knock_read_channel_1();
                                                                                                            90
91
92
93
94
95
96
97
98
99
        > 🐸 Drivers
                                                                                                                                                            sensor_knock_set_channel_2();
        > 🗁 Debug
               config.xml
               poskat.cfg
              www.poskat.ioc
                                                                                                                                     case 2:
               poskat.launch
              STM32F407IGHX FLASH.Id
                                                                                                                                                if (timer_get_tick() - knock_sensor_tick_count > 3
               STM32F407IGHX RAM.Id
                                                                                                                                                              knock_sensor_state = 0;
                                                                                                                                                            knock_sensor_tick_count = timer_get_tick();
// read channel 2
                                                                                                            101
                                                                                                                                                              sensor_knock_read_channel_2();
                                                                                                            102
103 //
                                                                                                                                                                   APP_LOGD("send message to device");
                                                                                                            104
105
                                                                                                            106
                                                                                                                                     default:
                                                                                                                                                break;
                                                                                                                                    }
```

1.5. Import and build nrf project

Extract "SDK file" you have downloaded

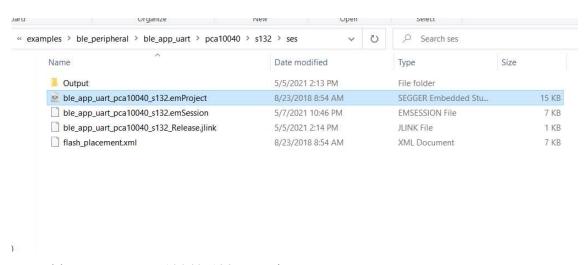
We have 1 folder for soft device is S132



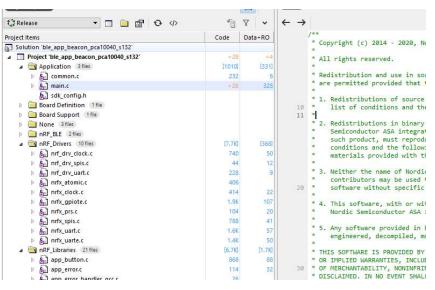
Folder S132

1.5.1. Import

Clone source code from git: https://github.com/ductuanhoang/Phone_Position_Detect "your project folder\examples\ble_peripheral\ble_app_beacon\pca10040\s132\ses"



Open ble app_uart_pca10040s132emProject

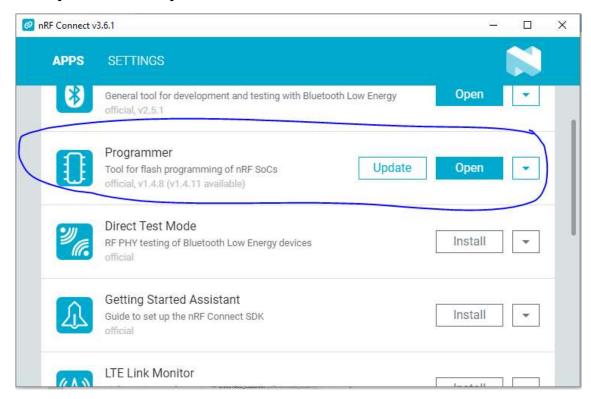


- Right click project and build
- F5 flash project

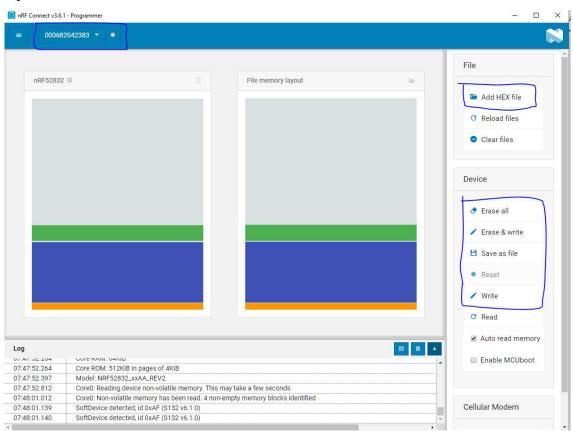
2. Download the code

• We must download soft sdk (S132) first (if you haven't run ble chip before)

2.1. Open NRF desktop tools



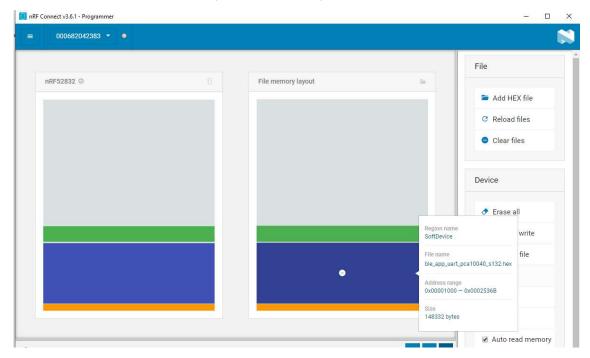
Open or install first



- Chose the board ID (left side)
- Add hex file:
 - o The first "ble_app_beacon_pca10040_s132.hex" at link "Phone_Position_Detect\examples\ble_peripheral\ble_app_beacon\pca1 0040\s132\ses\Output\Release\Exe\"
 - o Hex file for software device "S132" go to folder S132 before
- You can see the "file memory layout" at the tool
- Download to board: "Write"

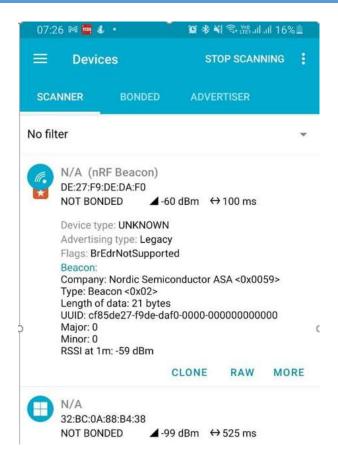
Note:

You can use the segger compiler for download new soure after you do all step above, because the device need S132 (software device) first and one time



3. Run

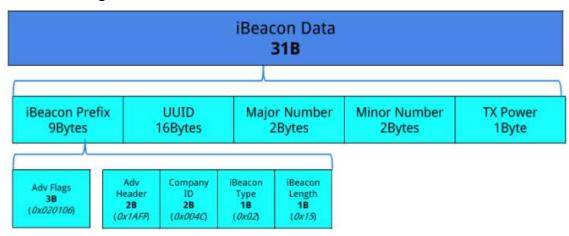
Open IOS or Android app



4. Message format

Because we custom message ADV for iBeacon so you can see the message difference with normal message.

Normal message:



Company ID: 0x0059

We custom UUID and Major Number and Minor Number

UUID = cf85-de27-f9de-daf0-00000...

- 0xCf85 (2 bytes) is UUID for Device grouping
- 0xde27f9dedaf0 (8 bytes) is MAC of the device
- 0x0000 ...(12 bytes) (include 12 bytes of the lastest message from SBC) include major and minor number

TX power can be set by SBC message

