Nordic© PC BLE driver concept implemented on a custom board

Space: Vieren Frank
Page location: https://confluence.televic.com/pages/viewpage.action?pageId=131105182

Author: FRV
Date: 10/10/2018

Disclaimer / Confidentiality / Copyright

All information in this document is subject to change without notice.
All information in this document is confidential.
No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Televic.
© 2018 Televic NV. All rights reserved.

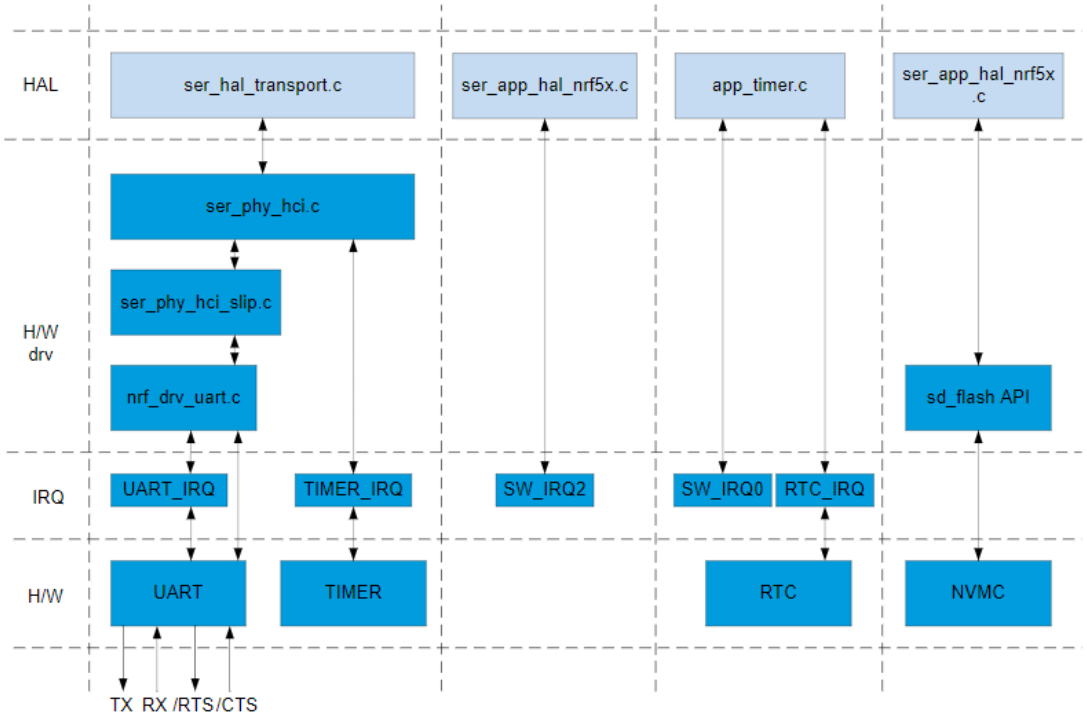
Table of Contents

1 Introduction

The use of the PC BLE driver should make it much easier to implement the UART (HCI) communication between the connectivity board (Nordic nRF5 DK) and the custom application board (Exceet iMX6UL-S). No longer the uart driver (cfr. nrf_drv_uart.c) for the custom board must be written as required when using the Nordic serialization concept as documented in the SDK (See the screenshot below).

UART_HCI PHY (reliable UART)

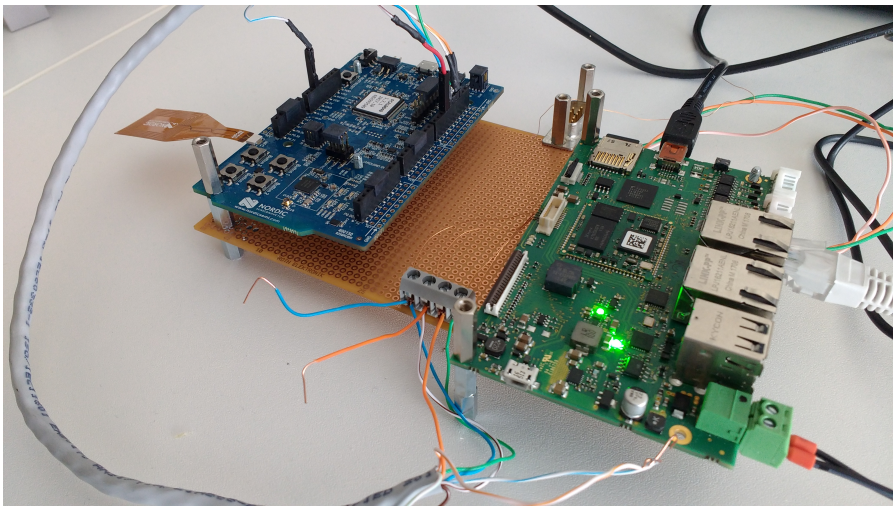
Similarly as for UART PHY, you must provide a UART hardware driver for UART_HCI PHY to substitute the `nrf_drv_uart.c` driver (shown in Figure 2). This driver must implement part of the UART driver - legacy layer API. UART_HCI has two upper-layer drivers that must be ported: `ser_phy_hci_slip.c` and `ser_phy_hci.c`. The latter uses the hardware TIMER and its interrupt TIMER_IRQ.



UART HCI means use of H5 protocol for UART communication, makes it more reliable, SLIP encoding.

2 HW setup

Nordic nRF5 DK as connectivity board (left on picture below, Exceet evk right on picture below) with UART connection between both boards.



3 SW Installation steps

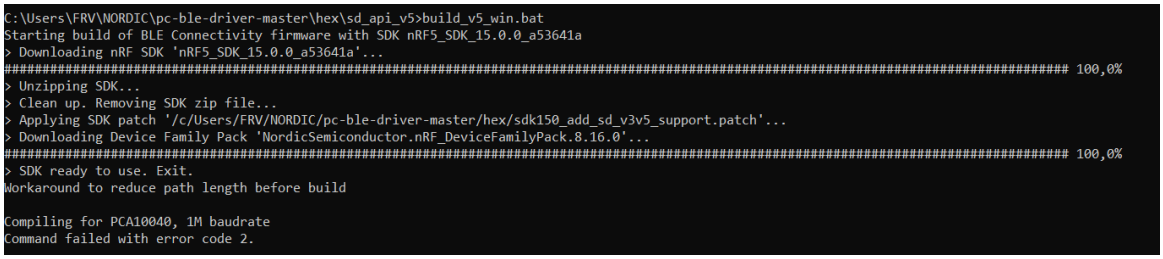
3.1 Application board

1. Prepare the Exceet iMX6 UL-S board with a valid Linux environment to run the PC BLE driver applic : Exceet Yocto + Boost libraries and Udev libraries, create bootable image file on sd-card
2. Download the PC BLE driver sources from the Nordic GIT HUB project : https://github.com/NordicSemiconductor/pc-ble-driver
3. Build PC BLE driver applic (HR monitor : pc-ble-driver-master\examples\heart_rate_monitor\main.c) via qt-creator (use of toolchain created by Yocto for cross compiling)
   Remark : currently HWFC is not set, Boost throws an exception when enabling HWFC, so the RTS/CTS lines are not used. This must be tackled to improve reliability of the UART communication between both boards (mainly preventing message flooding).
4. Deploy the PC BLE driver applic to the Exceet board and use as program arguments : /dev/ttymxc2 115200

3.2 Connectivity board

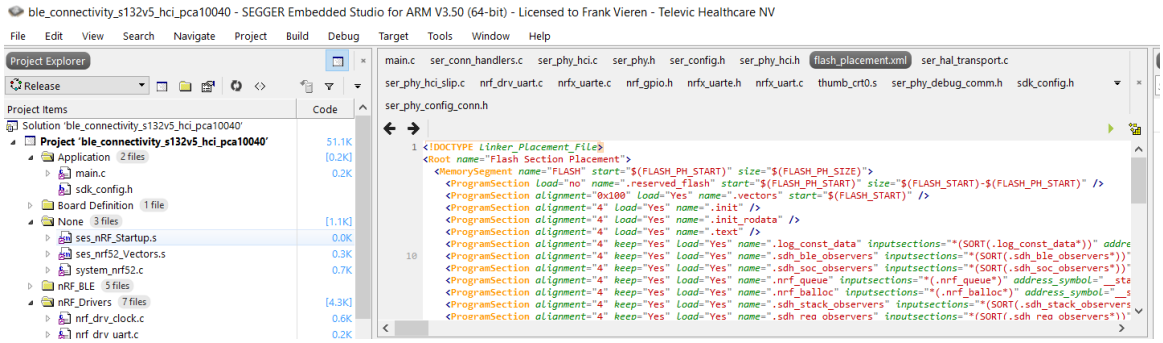
1. Run the Windows batch file, **build_v5_win.bat,** under the folder : ..\pc-ble-driver-master\hex\sd_api_v5:
   Downloads the correct SDK v15 sources (this is not the latest SDK v15 version)
   Applies the patch to these SDK v15 files

   The batch file exits before completely finishing when building the Connectivity s132_hci applic via the KEIL compiler.
   The KEIL compiler/linker fails when exceeding the linkage size for the evaluation license version of KEIL IDE ("non licensed").

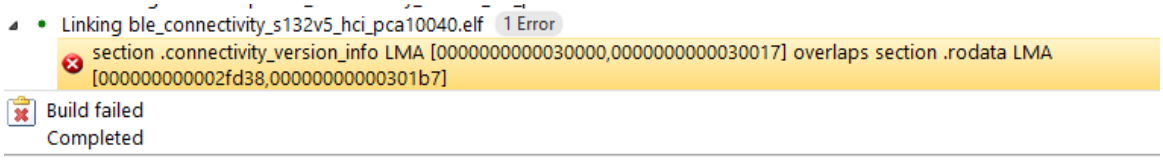


2. Adapt (revert) UART pin defines in the header file : .. \sdk\s150\components\serialization\common\transport\ser_phy\config**ser_phy_config_conn.h**

   as not the standard UART pins (for PC virtual COM port) are to be used in our setup.
   #define SER_PHY_UART_RX SER_CON_RX_PIN
   #define SER_PHY_UART_TX SER_CON_TX_PIN
   #define SER_PHY_UART_CTS SER_CON_CTS_PIN
   #define SER_PHY_UART_RTS SER_CON_RTS_PIN
   Knowing:
   // serialization CONNECTIVITY board
   #define SER_CON_RX_PIN 24 // UART RX pin number.
   #define SER_CON_TX_PIN 23 // UART TX pin number.
   #define SER_CON_CTS_PIN 25 // UART Clear To Send pin number. Not used if HWFC is set to false.
   #define SER_CON_RTS_PIN 2 // UART Request To Send pin number. Not used if HWFC is set to false.
3. Adapt the BAUDRATE and HWFC defines in header file : .. \sdk\s150\components\serialization\common**ser_config.h**
   #define **SER_PHY_UART_FLOW_CTRL NRF_UART_HWFC_DISABLED**
   #define SER_PHY_UART_PARITY NRF_UART_PARITY_EXCLUDED
   #define SER_PHY_UART_BAUDRATE_VAL **115200**
   Remark:
   As the Boost UART function in PC BLE Driver applic throws an exception when setting the flow control to HWFC as value, the HWFC is also disabled in the connectivity application.
   See also : https://stackoverflow.com/questions/28274367/how-to-make-boostasioserial-port-baseflow-control-use-hardware-flow-contro
   So far no flooding of messages experienced when HWFC is not enabled.
   BTW currently when HWFC is enabled in the connectivity application no bi-directional communication between both applics, the PC BLE driver applic will exit when it receives no responses after sending packets.
4. Manually build the Connectivity applic via SEGGER compiler (use of SEGGER IDE) : load the project **ble_connectivity_s132v5_hci_pca10040** from ... \sdk\s150\examples\connectivity\ble_connectivity\pca10040\ser_s132v5_hci\ses into SEGGER IDE

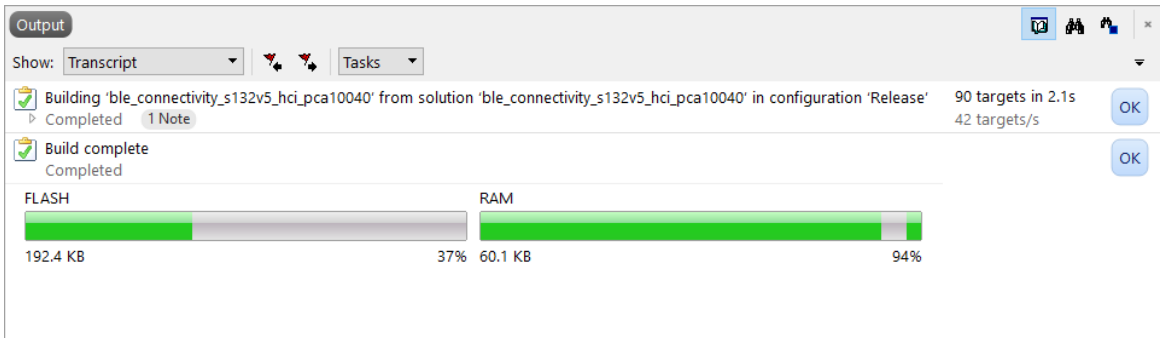


5. During the linkage process the **flash_placement** XML file gives linkage errors, use flash_placement file from the Nordic Dev Zone ticket (See : https://devzone.nordicsemi.com/f/nordic-q-a/37123/linker-issues-in-the-connectivity-firmware)



Use this XML file:
flash_placement.xml

6. Build and deploy the Connectivity applic to Nordic nRF5 DK (chip PCA10040/nRF52832), debug the applic to start it on the Connectivity Nordic DK board.



4 Result

Launch the Nordic nRF Toolbox on a Smart Phone and select the HRM applic, the simulated HR can be monitored after addressing the right BLE device, Nordic HRM.

5 Conclusion

Pros:

- Very nice concept compared to having to implement yourself the UART driver functionality in plain C to support the HAL transport layer. In other words simplifies the implementation process.
- Nice C++ implementation.

Contras:

- Use of a patch, API version 5 PC BLE driver sources not in sync with the SDK v15 API 5.
- Struggling when building the patched project, **ble_connectivity_s132v5_hci_pca10040** in SEGGER IDE (flash_placement, debug build fails), this is not the case with the non-patched SDK v15 version.
  Strange because normally full support expected for SEGGER.
- Not part of the Nordic SDK.
- Limited documentation, as far as I know no direct reference on Nordic web page cfr. SDK.