

Cortex-M0 BLE Solution for AB1602 controller

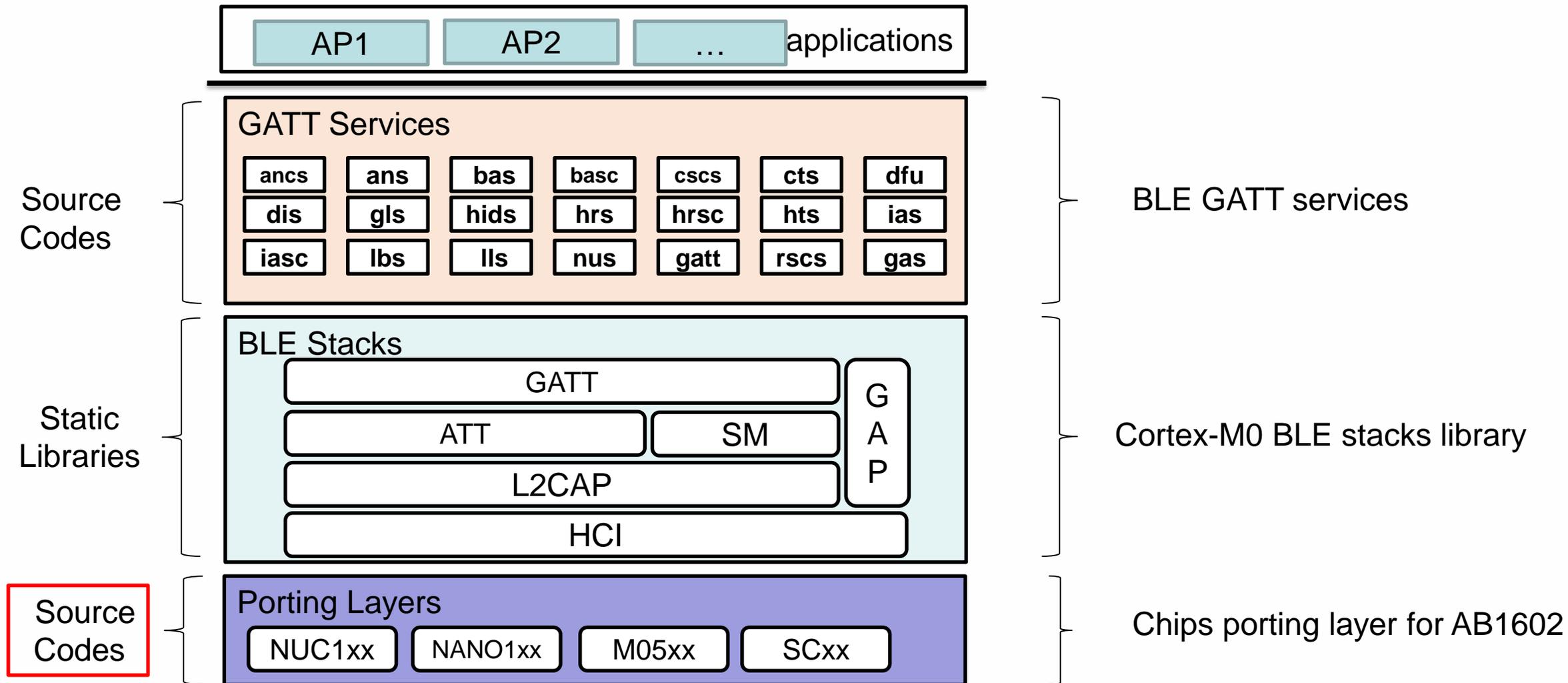
Tallplay

2017/07/11

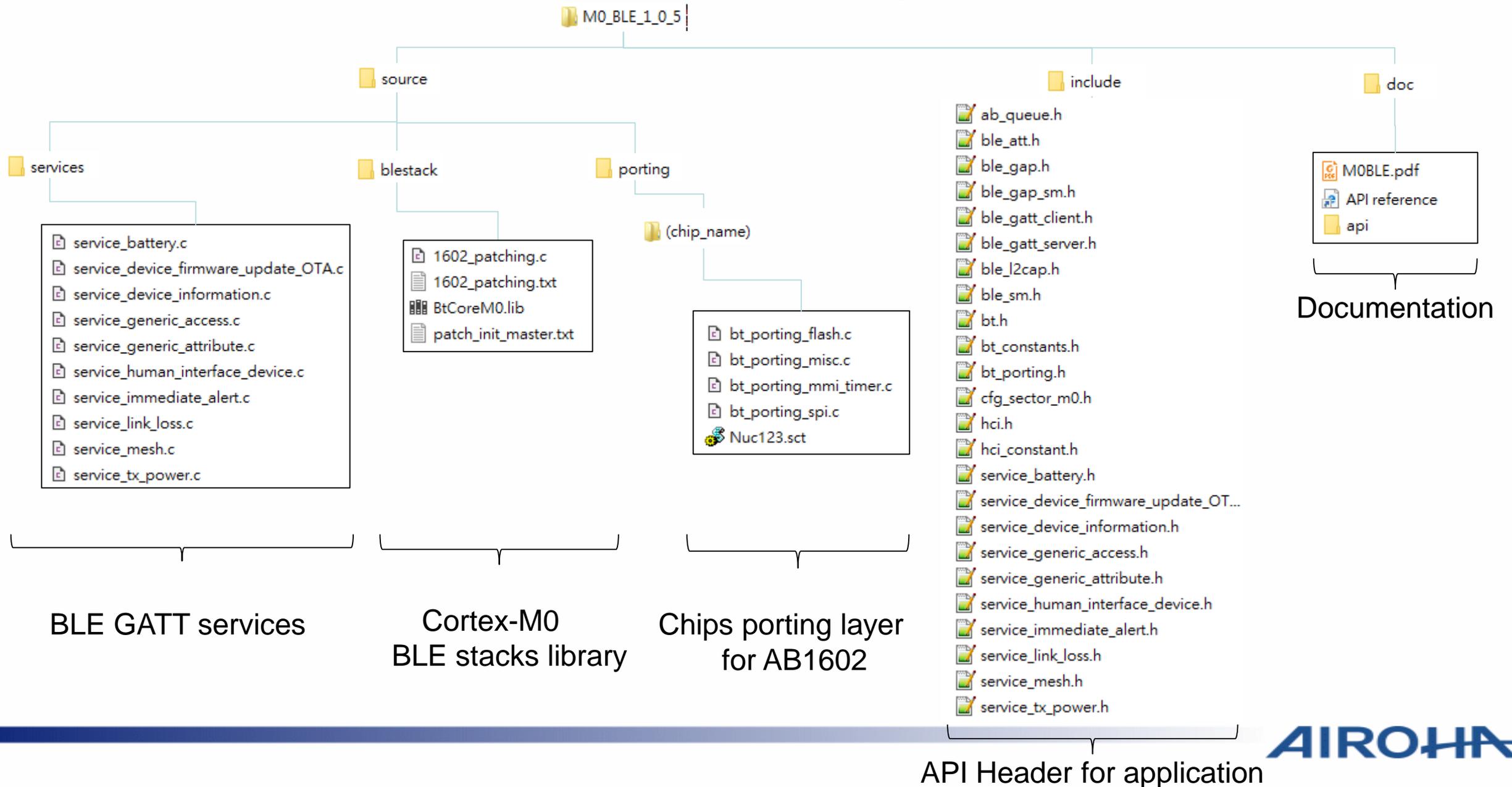
Agenda

- Introduction
 - Solution Architecture
 - File Hierarchy
 - Architecture vs Hierarchy
- Installation and Environment Setup

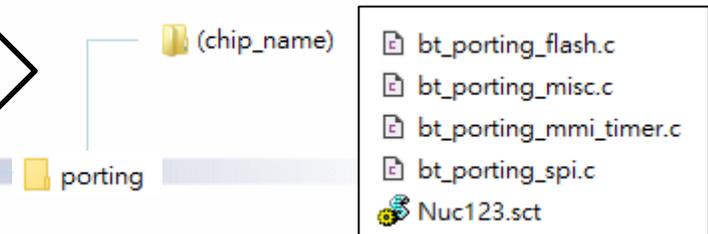
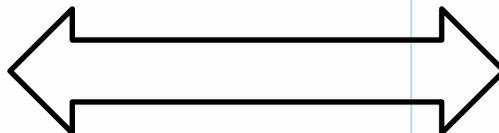
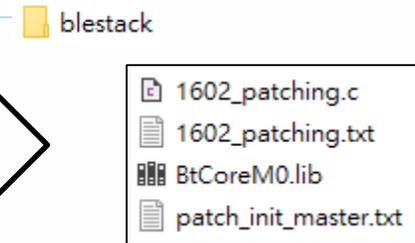
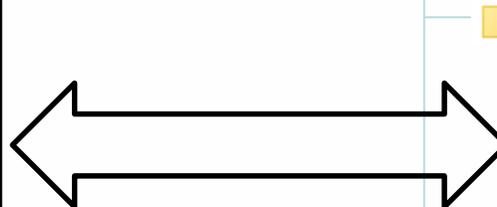
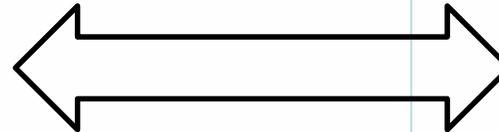
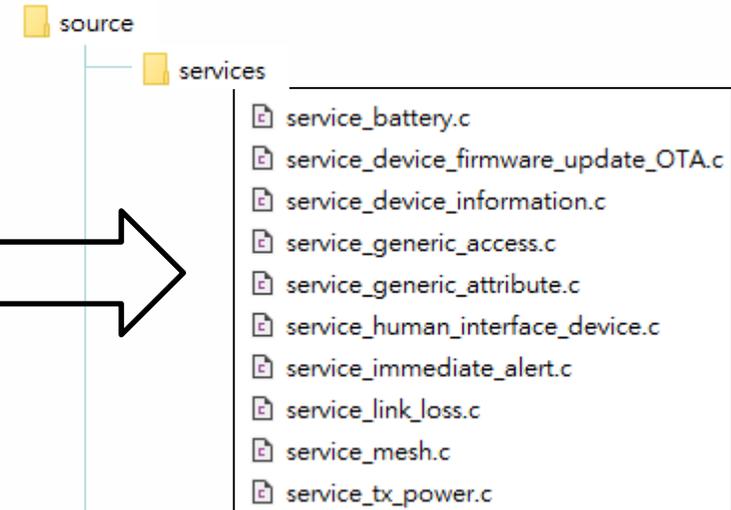
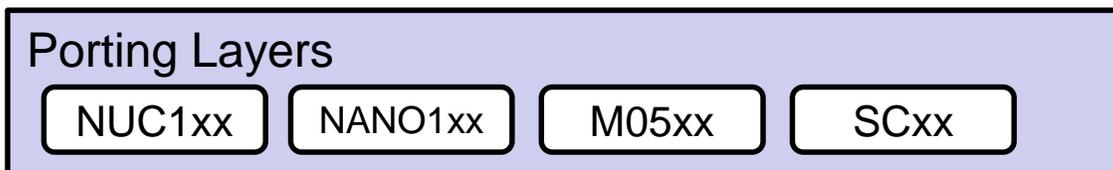
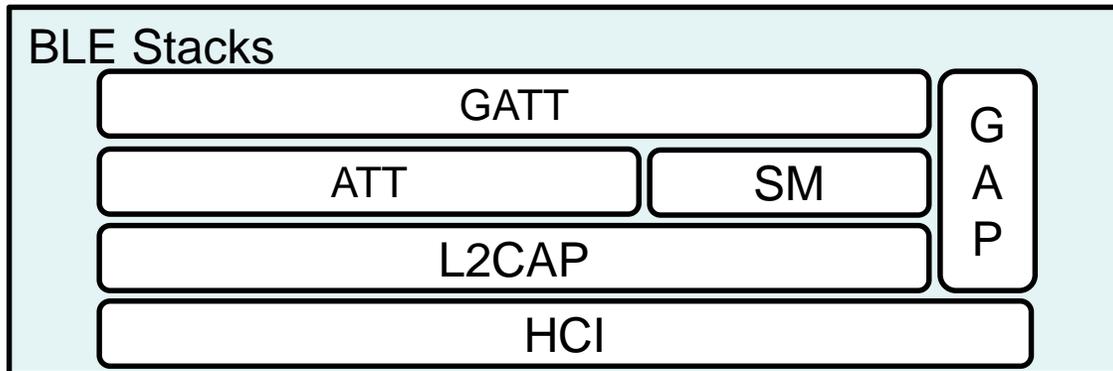
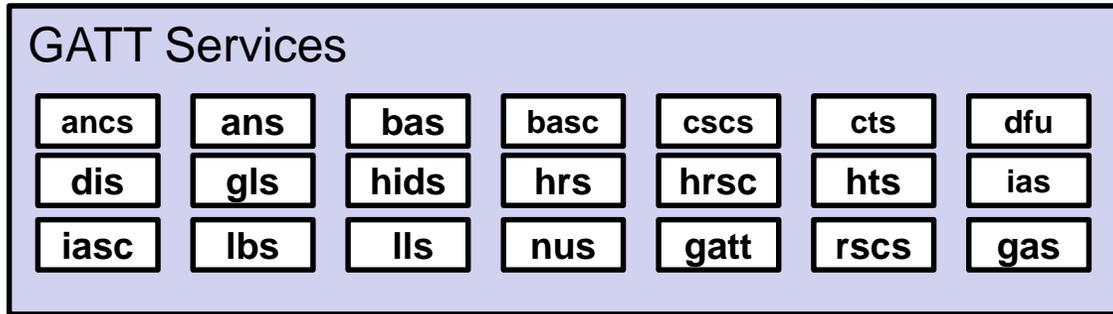
Solution Architecture



File Hierarchy

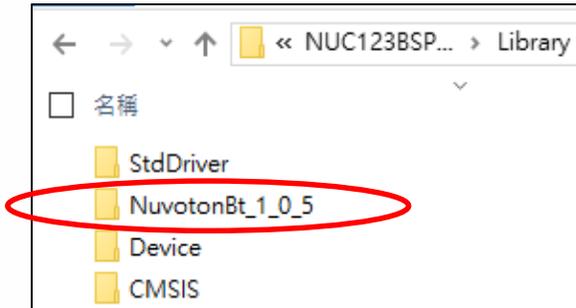
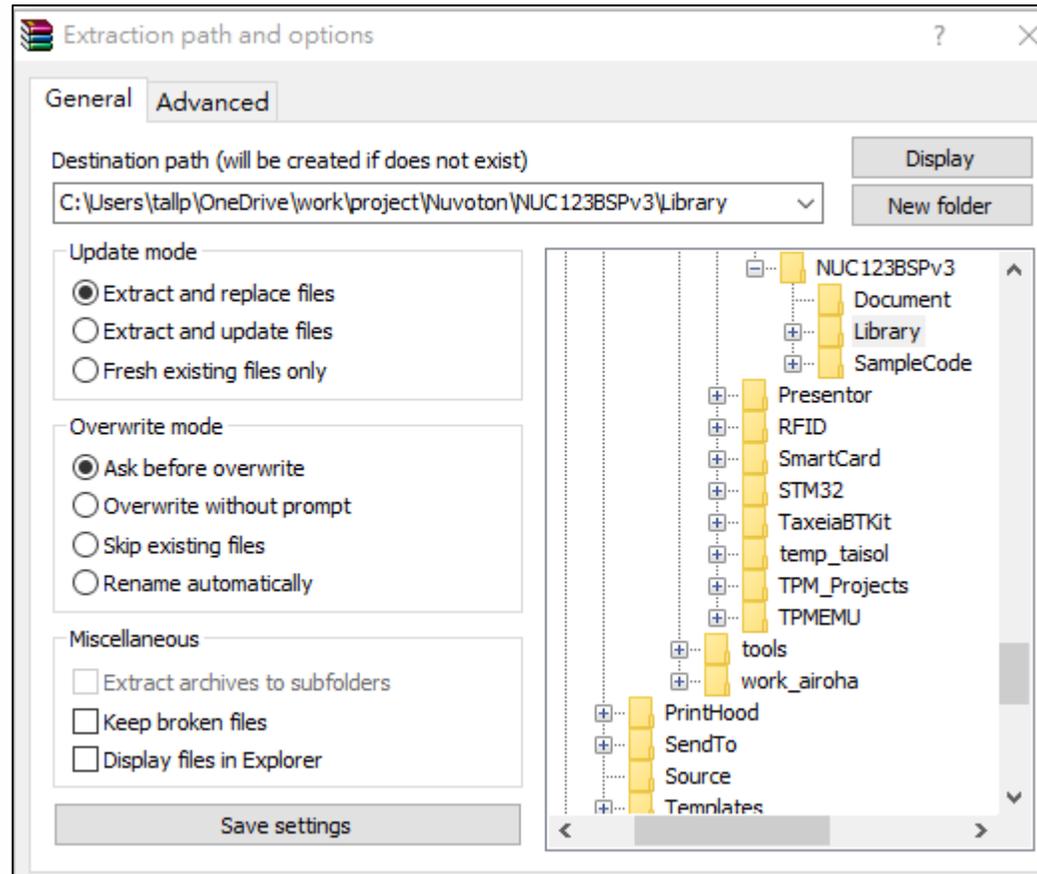
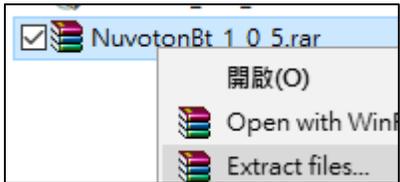


Architecture vs Hierarchy



Installation (NUC123 as example)

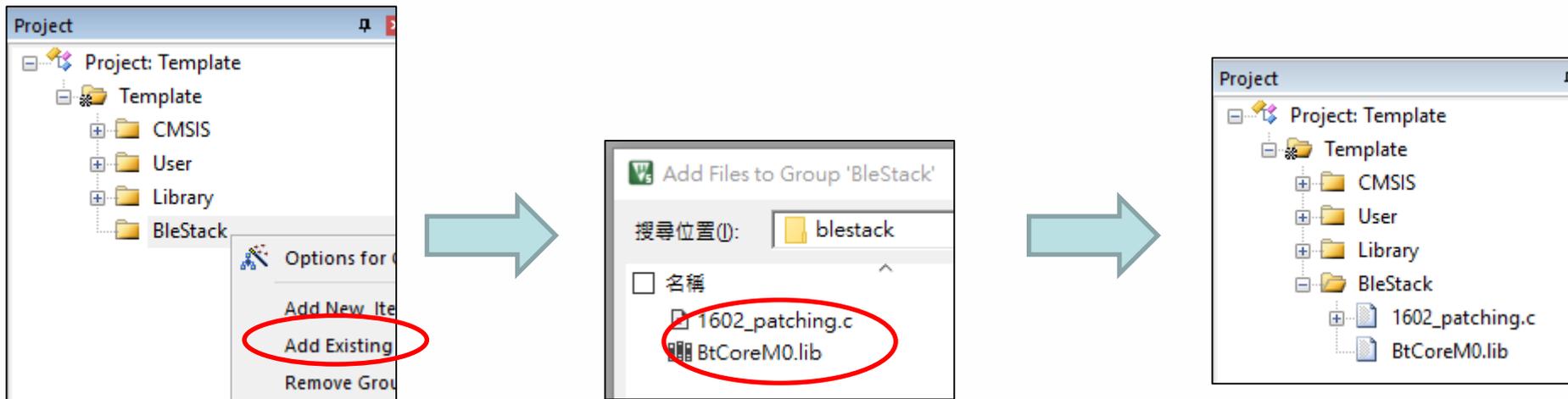
- Extract *MOBLE_x_x_x.rar* to library folder



Setup(1) – Add BLE Core Library (NUC123 as example)

- Add BLE M0 stacks library to project.
 - BtCoreM0.lib
 - 1602_patching.c

NuvotonBt_1_0_5 > source > blestack

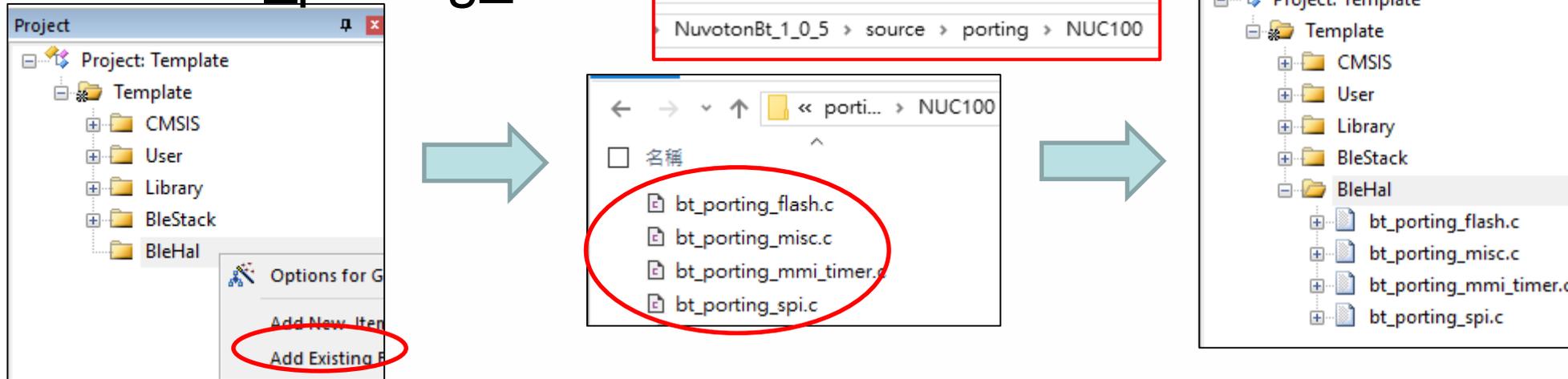


Setup(2) -- add porting layer

(NUC123 as example)

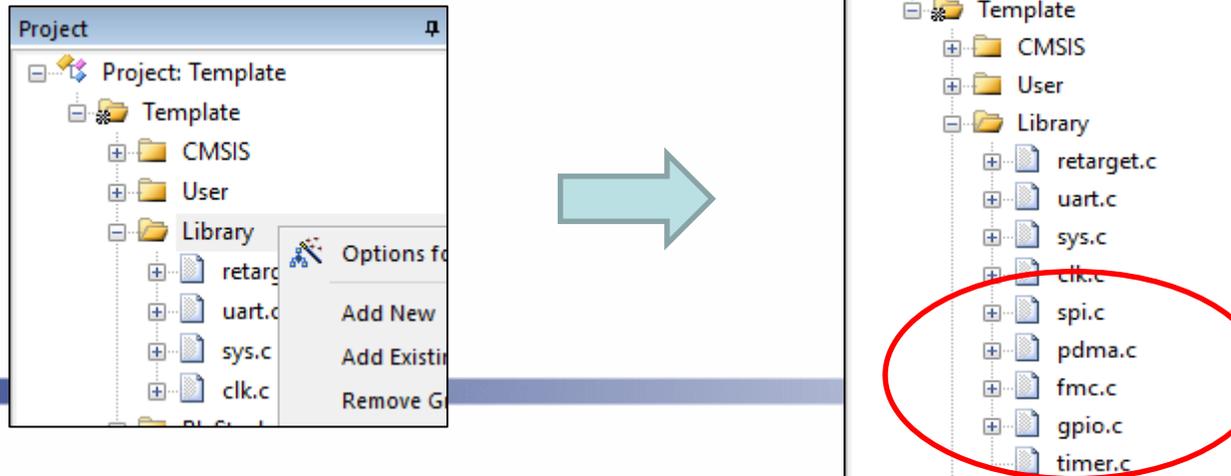
- Add porting layer to project.

– Bt_porting_*.c



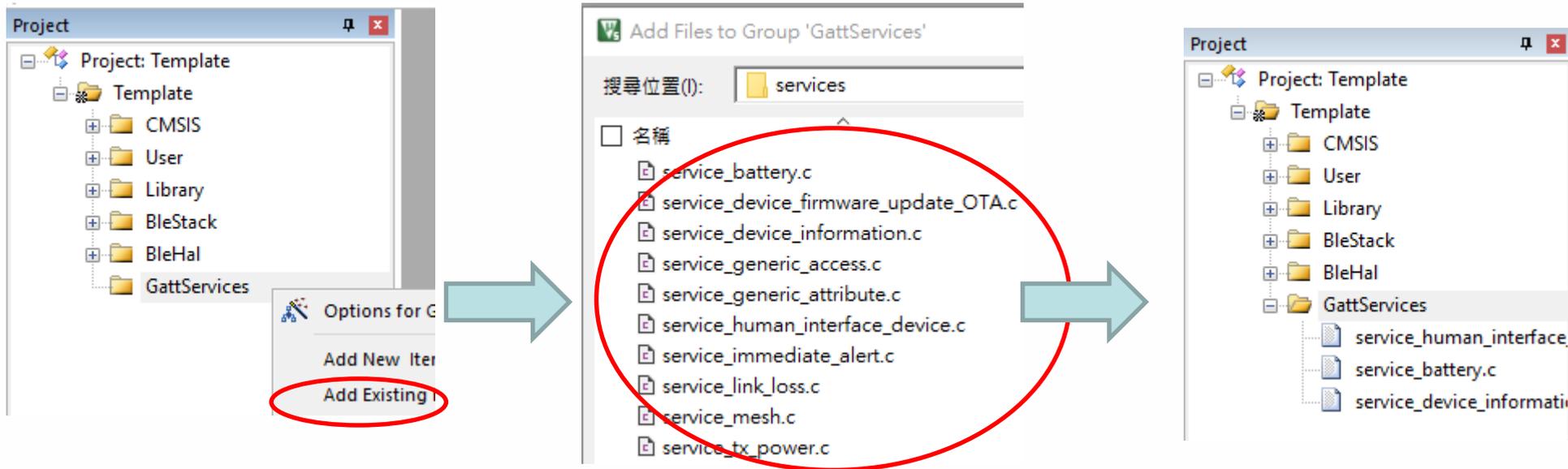
- Add related driver. (NUC123 as example)

- Spi
- Pdma.c
- Fmc.c
- Gpio.c
- Timer.c



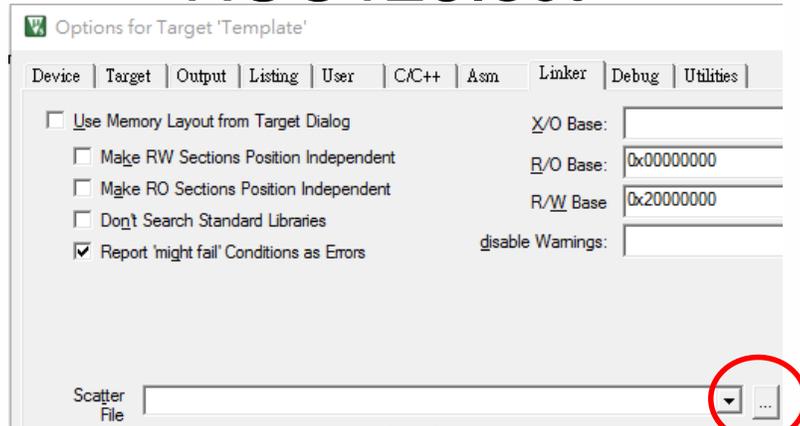
Setup(3) -- Add GATT Services (NUC123 as example)

- Add needed GATT services to `NuvotonBt_1_0_5 > source > services`
 - `Service_xxxx.c`

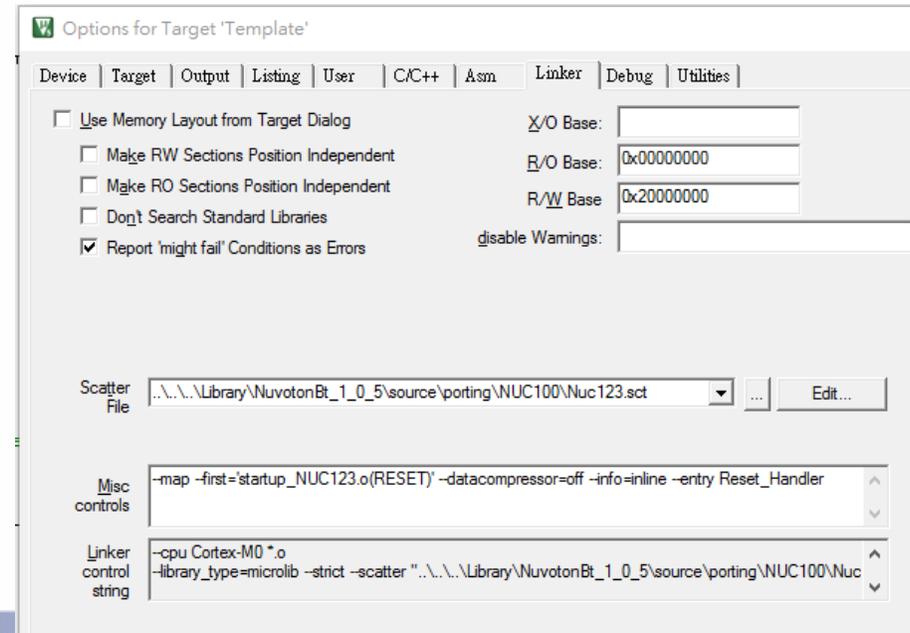
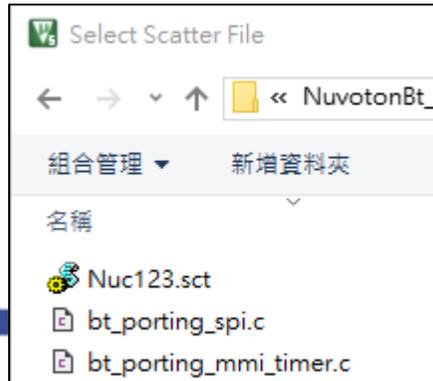


Setup(4) – Set scatter file(NUC123 as example)

- Setup scatter file
– NUC123.sct

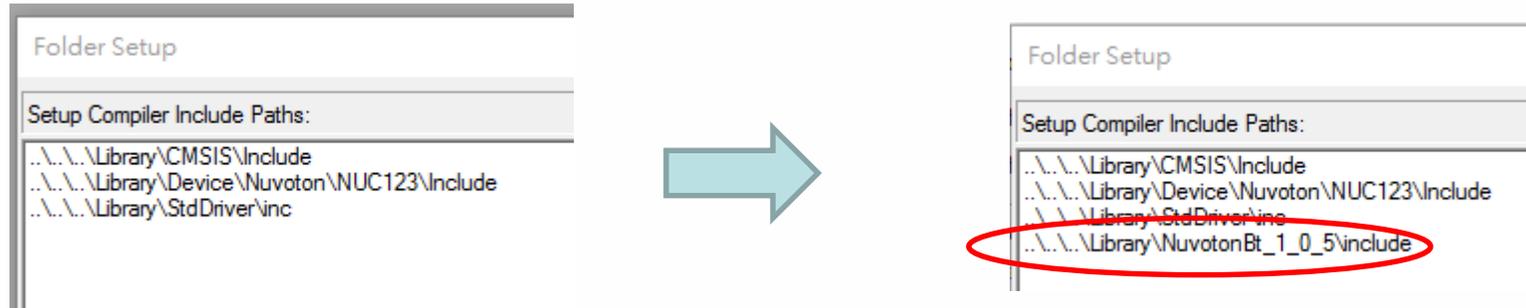


NuvotonBt_1_0_5 > source > porting > NUC100



Setup(5) – Set Include Path and Size of HEAP/STACK

- Setup compiler include path:



- Setup size of HEAP/STACK
 - Stack 1024 bytes
 - Heap 16384 bytes

```
startup_NUC123.s
10 ; User may overwrite stack size setti
11 IF :LNOT: :DEF: Stack_Size
12 Stack_Size EQU 0x00000400
13 ENDIF
14 AREA STACK, NOINIT, RE
15 Stack_Mem SPACE Stack_Size
16 __initial_sp
17
18
19 ; <h> Heap Configuration
20 ; <o> Heap Size (in Bytes) <0x0-0xFFFFF>
21 ; </h>
22
23 IF :LNOT: :DEF: Heap_Size
24 Heap_Size EQU 0x00004000
25 ENDIF
26
```

Setup(6) --- AB1602 Initialization

1. Calling initialization function before main loop.
 - `Airoha_PortingInit();`
 - `Bt_InitEx2(bt_evt_hdl, ...);`
2. Add .h header file
 - `Bt.h` (for `Bt_InitEx2(...)` prototype)
 - `Cfg_sector_m0.h`
 - `Bt_porting.h` (For `Airoha_PortingInit()` prototype)
3. Add BT event callback function.
 - `bt_evt_hdl(bt_evt_t*)`

```
int main()
{
    int8_t ch;

    /* Unlock protected registers */
    SYS_UnlockReg();

    SYS_Init();

    /*airoha porting init*/
    Airoha_PortingInit();

    /* Lock protected registers */
    SYS_LockReg();

    /* Init UART0 to 115200-8n1 for print message */
    UART_Open(UART0, 115200);

    /* Bt Init*/
    BT_InitEx2(bt_evt_hdl, BT_INIT_DATA, BT_INIT_DATA_NUM,
              BT_LOG_HCI_CMD | BT_LOG_HCI_EVT);
}
```

```
*****
#include <stdio.h>
#include "NIC123.h"
#include "bt.h"
#include "cfg_sector_m0.h"
#include "bt_porting.h"
*****
```

```
void bt_evt_hdl(bt_evt_t *evt)
{
    switch(evt->evt_id)
    {
        case BT_MISC_EVENT_INITED:
            printf("init = %d\n", evt->evt.misc_evt.param.inited.is_success);
            break;
    }
}
```

Setup(7) – Test

- Check if AB1602 initialization success.
 - If success, then user could start develop application.

```
Build Output
.\obj\template.axf: Warning: L63
Program Size: Code=40084 RO-data
Finished: 0 information, 1 warni
After Build - User command #1: f
After Build - User command #2: f
".\obj\template.axf" - 0 Error(s)
Build Time Elapsed: 00:00:03
```

```
COM5:115200baud - Tera Term VT
文件(F) 編輯(E) 設定(S) 控制(O) 視窗(W) 幫助(H)
Simple Demo Code
Please Input Any Key
Input: [READ_BUFFER_SIZE]
[Unknown(0xFF) ]
01 30 0e 31 00 .0.1.
[COMMAND_COMPLETE <READ_BUFFER_SIZE FAIL> ]
01 05 10 0c ....
[LE_READ_BUF_SIZE]
[COMMAND_COMPLETE <LE_READ_BUF_SIZE OK> ]
01 02 20 00 ff 00 08 .. ....
[LE_SET_EVENT_MASK]
9f 01 00 00 00 00 00 00 .....
init = 1
```

Q&A