



DesignNews

Embedded Studio Primer

DAY 3: Nordic SDK and Embedded Studio

Sponsored by



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.
- Submit questions for the lecturer using the Q&A widget. They will follow-up after the lecture portion concludes.



Fred Eady

Visit 'Lecturer Profile' in your console for more details.

Embedded Studio Primer

Nordic SDK and Embedded Studio

ble_app_uart_pca10040_s132 - SEGGER Embedded Studio for ARM V4.52c (64-bit) - Licensed to Frederick Eady - Ongoing Systems LLC (Stopped)

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

Disassembly

```

main
00029D96 F001F95B bl 0x0002B050 <nrf_ble_gatt_init
-- main.c -- 512
APP_ERROR_CHECK(err_code);
00029DA0 B108 cbz r0, 0x00029DA0
00029DA2 F7DF8E6 bl 0x00026F6C <app_error_handler
-- main.c -- 513
err_code = nrf_ble_gatt_att_mtu_periph_set(&m_ga
00029DA8 4805 ldr r0, =0x20003888 <m_gatt>
00029DA2 21F7 movs r1, #0xF7
00029DA4 F001F968 bl 0x0002B078 <nrf_ble_gatt_att
-- main.c -- 515
APP_ERROR_CHECK(err_code);
00029DA8 B118 cbz r0, 0x00029DB2
-- main.c -- 516
}
00029DA0 E8BD4008 pop.w {r3, lr}
-- main.c -- 515
APP_ERROR_CHECK(err_code);
00029DAE F7FDB8DD b.w 0x00026F6C <app_error_handle
-- main.c -- 516
}
00029DB2 B008 pop {r3, pc}
00029B6D .word 0x00029B6D
00029B8B .word 0x20003888
-- main.c -- 741
{
bool erase_bonds;
// Initialize.
uart_init();
00029DBC B500 push {lr}
-- main.c -- 623
app_uart_comm_params_t const comm_params =
00029DBE 4D6D ldr r5, =0x0002D77C
00029DC0 CD0F ldm r5!, {r0-r3}
-- main.c -- 737
/**@brief Application main function.
*/
int main(void)
{
00029DC2 B0B1 sub sp, sp, #0xC4
-- main.c -- 623
app_uart_comm_params_t const comm_params =
00029DC4 AC05 add r4, sp, #20
00029DC6 C40F stm r4!, {r0-r3}
00029DC8 E8950003 ldm r5, {r0-r1}
-- main.c -- 634
.baud_rate = NRF_UARTE_BAUDRATE_115200
#endif
);
APP_UART_FIFO_INIT(&comm_params,
00029DCC 4B6A ldr r3, =0x200038DC <rx_buf.1198
00029DCE 4A6B ldr r2, =0x200039DC <tx_buf.1198
00029DD0 9301 str r3, [sp, #4]
00029DD2 F44F7380 mov.w r3, #0x100
-- main.c -- 623
app_uart_comm_params_t const comm_params =
00029DD6 E8B40003 stm r4, {r0-r1}
-- main.c -- 634
.baud_rate = NRF_UARTE_BAUDRATE_115200

```

main.c app_error_weak.c

```

int main()
{
    if (NRF_LOG_PROCESS() == false)
    {
        nrf_pwr_mgmt_run();
    }
}

/**@brief Function for starting advertising.
*/
static void advertising_start(void)
{
    uint32_t err_code = ble_advertising_start(&advertising, BLE_ADV_MODE_FAST);
    APP_ERROR_CHECK(err_code);
}

/**@brief Application main function.
*/
int main(void)
{
    bool erase_bonds;

    // Initialize.
    uart_init();
    log_init();
    timers_init();
    buttons_leds_init(&erase_bonds);
    power_management_init();
    ble_stack_init();
    gap_params_init();
    gatt_init();
    services_init();
    advertising_init();
    conn_params_init();

    // Start execution.
    printf("\r\nUART started.\r\n");
    //NRF_LOG_INFO("Debug logging for UART over RTT started.");
    advertising_start();

    // Enter main loop.
    for (;;)
    {
        idle_state_handle();
    }
}

```

Globals

Expression	Value
SEGGER_RTT	<struct>
log_mempool	<struct>
m_nrf_log_app_button_logs_data_c	<struct>
m_nrf_log_app_logs_data_const	<struct>
m_nrf_log_app_timer_logs_data_co	<struct>
m_nrf_log_ble_nus_logs_data_cons	<struct>
m_nrf_log_clock_logs_data_const	<struct>
m_nrf_log_CLOCK_logs_data_const	<struct>
m_nrf_log_GPIOTE_logs_data_const	<struct>
m_nrf_log_nrf_ble_gatt_logs_data	<struct>
m_nrf_log_nrf_sdh_ble_logs_data	<struct>
m_nrf_log_nrf_sdh_logs_data_cons	<struct>
m_nrf_log_nrf_sdh_soc_logs_data	<struct>
m_nrf_log_PRS_logs_data_const	<struct>
m_nrf_log_pwr_mgmt_logs_data_con	<struct>
m_nrf_log_sortlist_logs_data_con	<struct>
m_nrf_log_UART_logs_data_const	<struct>
m_nrf_log_UARTE_logs_data_const	<struct>
nrf_drv_uart_use_easy_dma	""
nrf_log_backend_rtt_api	<struct>
nrf_nvic_state	<struct>
SystemCoreClock	0x03d09000

Output Show: Target

- Preparing target Completed
- Downloading: 148.9 KB in 0.2s Download succ 528.1 KB/s
- Downloading: 31.8 KB in 0.0s Download succ 362.3 KB/s

Call Stack

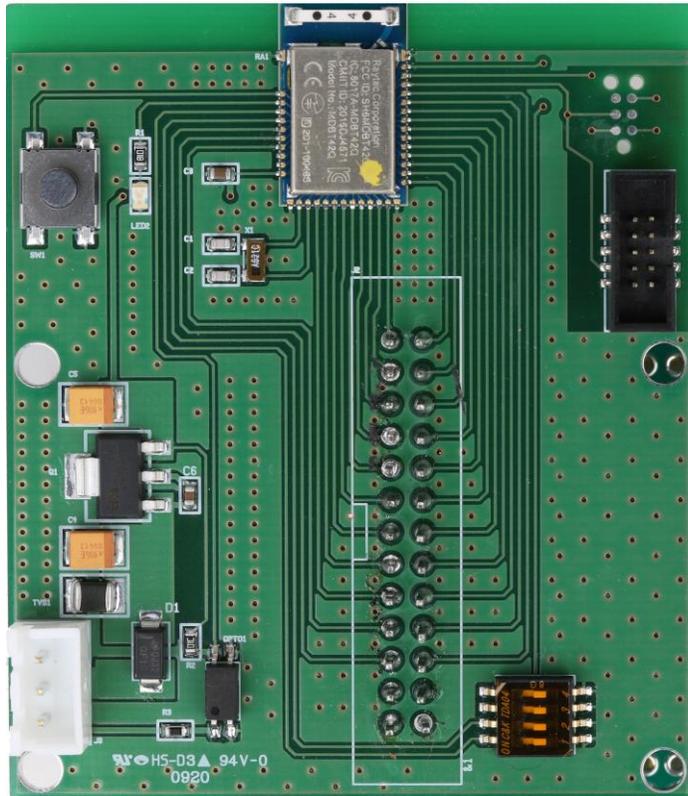
Function	Call Address
int main()	0x00029DBC 0x000262B2



CortexM4 on J-Link 31,827 Cycles Built OK INS R/O Ln 202 Col 1

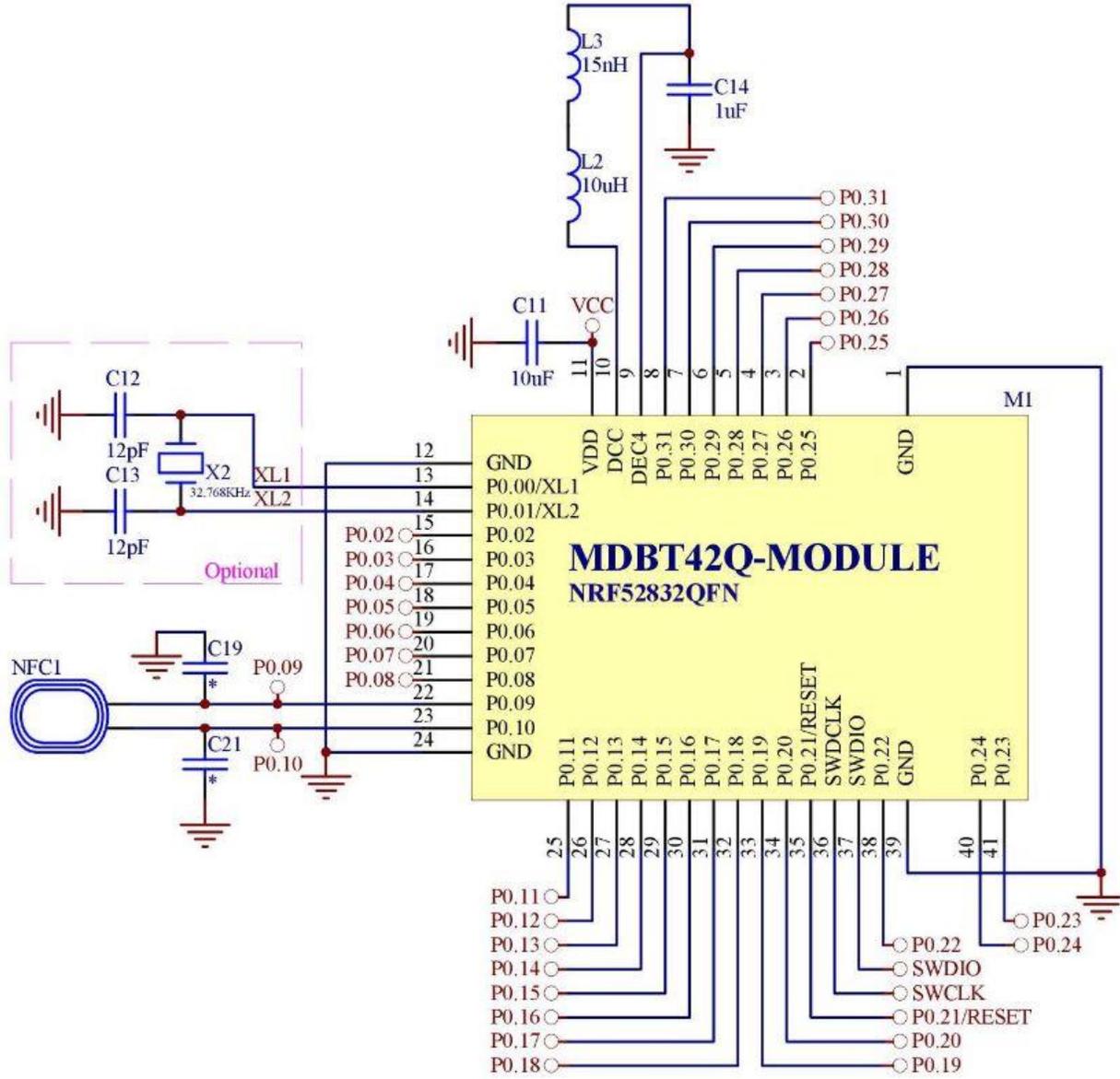
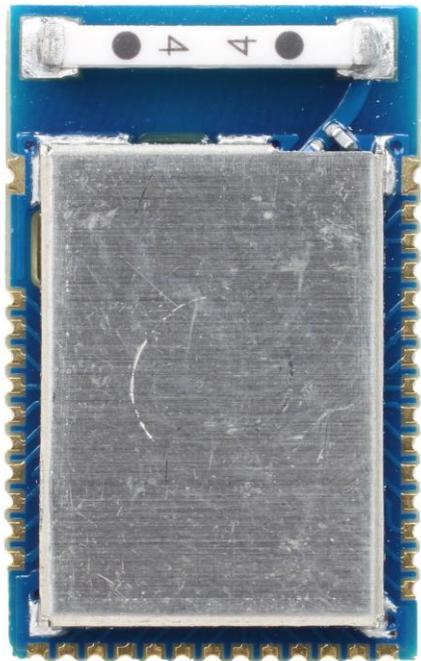
AGENDA

- Coding the BLE Peripheral
- Coding the B4I BLE Central



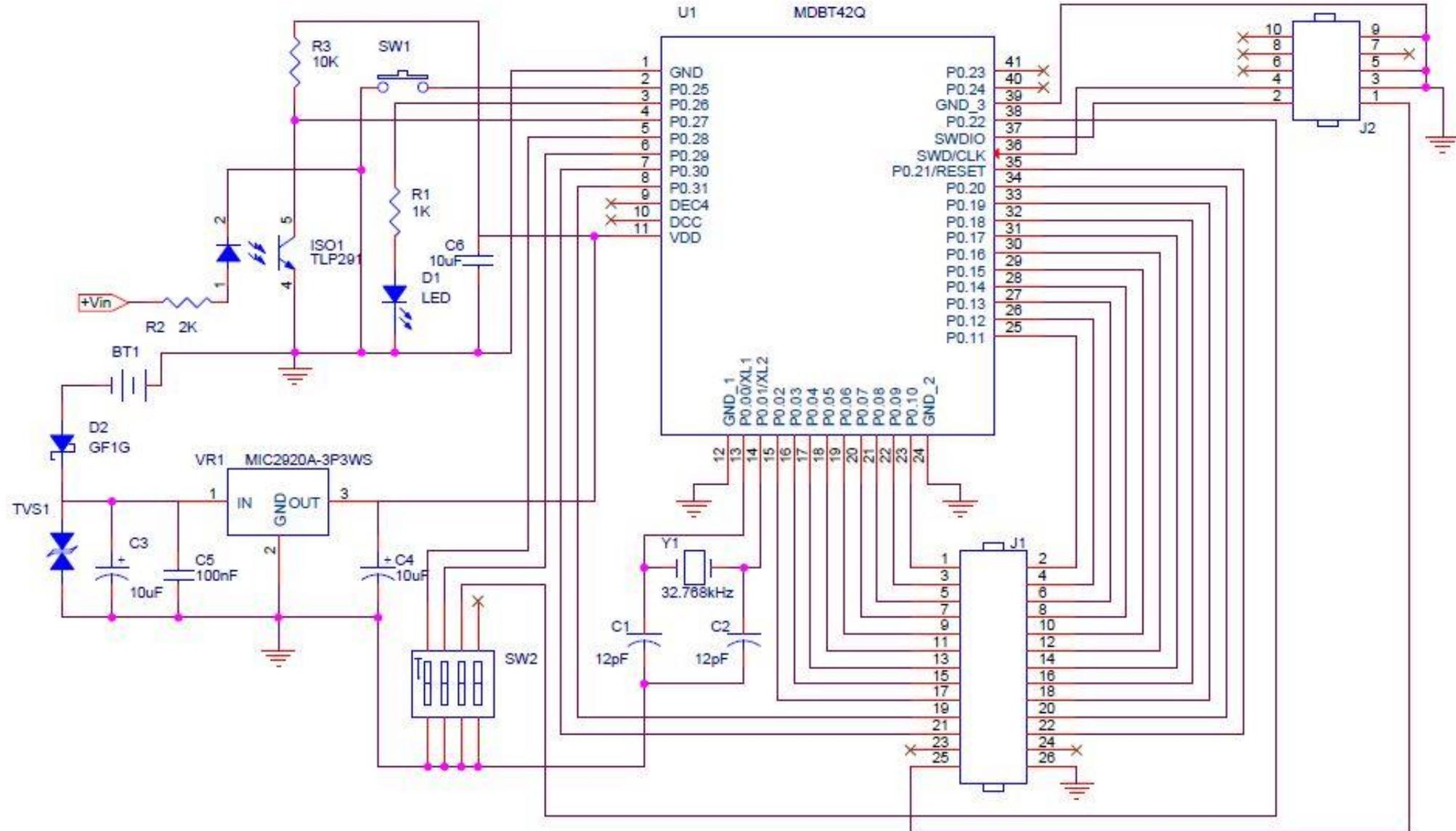
Embedded Studio Primer

Coding the BLE Peripheral



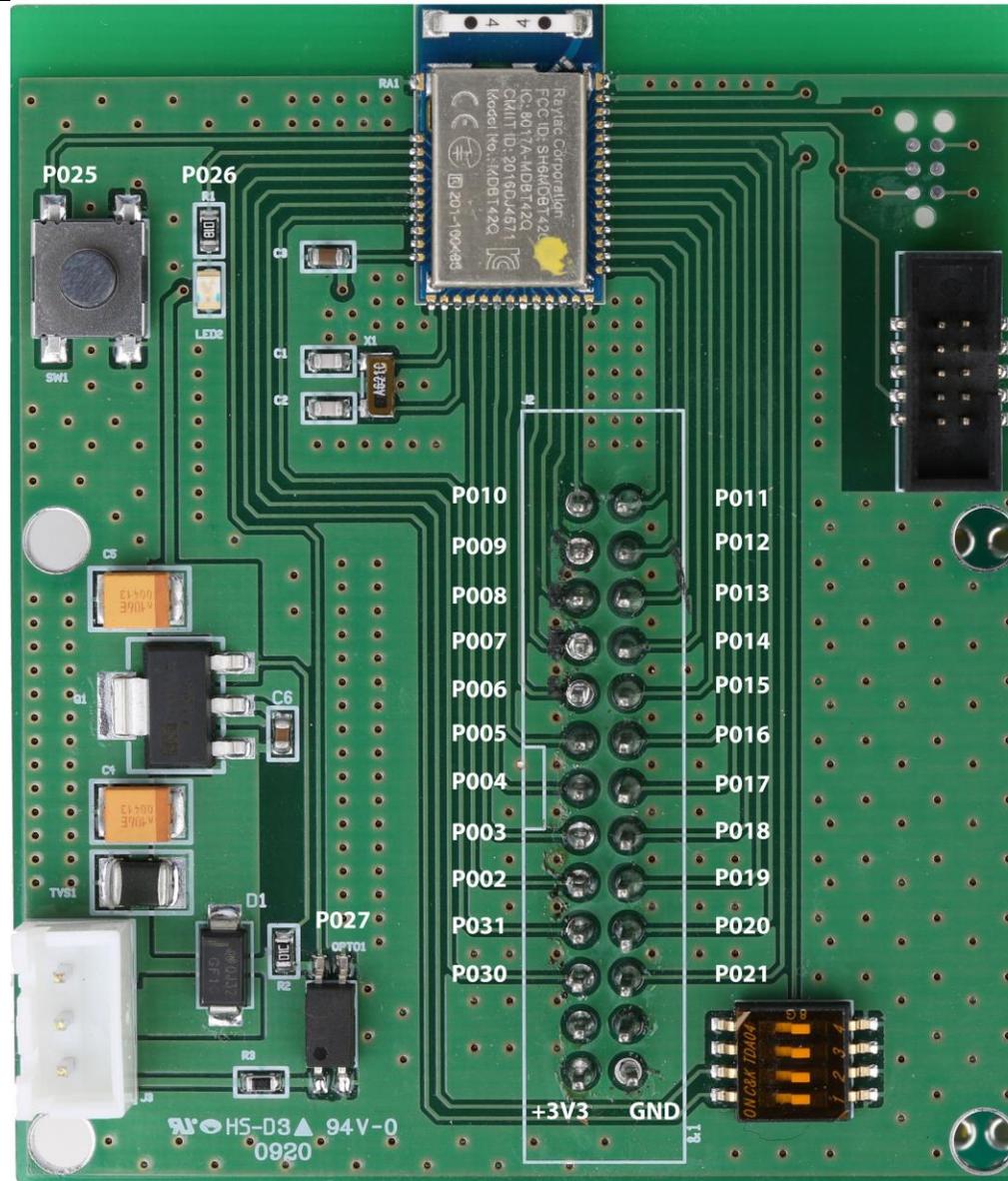
Embedded Studio Primer

Coding the BLE Peripheral



Embedded Studio Primer

Coding the BLE Peripheral



Embedded Studio Primer

Coding the BLE Peripheral

```
#define DEVICE_NAME           "Nordic_UART"           /**< Name of device. Will be included in the advertising data. */
#define NUS_SERVICE_UUID_TYPE BLE_UUID_TYPE_VENDOR_BEGIN /**< UUID type for the Nordic UART Service (vendor specific). */

#define APP_BLE_OBSERVER_PRIO 3                      /**< Application's BLE observer priority. You shouldn't need to modify this value. */

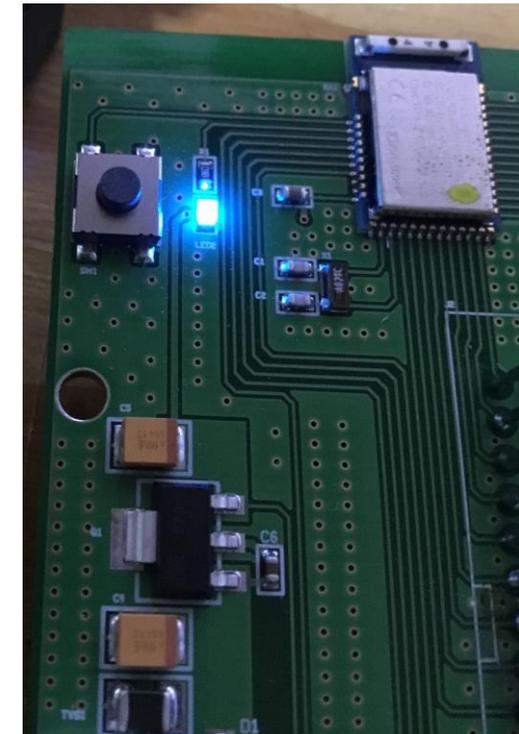
#define APP_ADV_INTERVAL      64                     /**< The advertising interval (in units of 0.625 ms. This value corresponds to 40 ms). */

#define APP_ADV_DURATION      0                      /**< The advertising duration 0 = advertise never times out */
```

```
static void advertising_init(void)
{
    uint32_t      err_code;
    ble_advertising_init_t init;

    memset(&init, 0, sizeof(init));

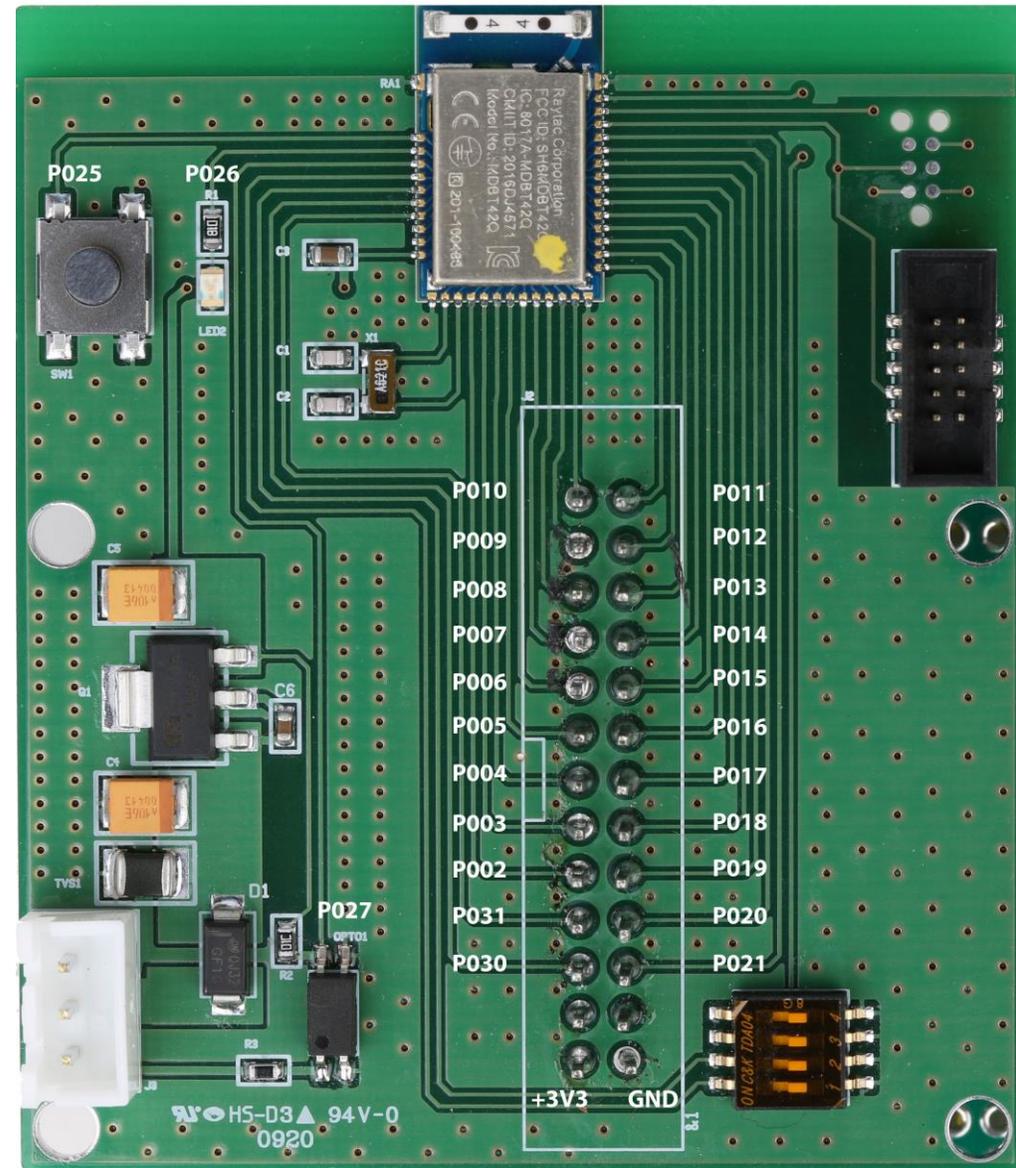
    init.advdata.name_type      = BLE_ADVDATA_FULL_NAME;
    init.advdata.include_appearance = false;
    init.advdata.flags          = BLE_GAP_ADV_FLAGS_LE_ONLY_GENERAL_DISC_MODE;
```



Embedded Studio Primer

Coding the BLE Peripheral

```
static void nus_data_handler(ble_nus_evt_t * p_evt)
{
    if (p_evt->type == BLE_NUS_EVT_RX_DATA)
    {
        switch(p_evt->params.rx_data.p_data[0])
        {
            case 0:
                NRF_GPIO->OUTCLR = 0x000E001C; //0000 0000 0000 1110 0000 0000 0001 1100
                NRF_GPIO->OUTSET = 0x00000004;
                printf("P0.02 = ON\r\n");
                break;
            case 1:
                NRF_GPIO->OUTCLR = 0x000E001C; //0000 0000 0000 1110 0000 0000 0001 1100
                NRF_GPIO->OUTSET = 0x00000008;
                printf("P0.03 = ON\r\n");
                break;
            case 2:
                NRF_GPIO->OUTCLR = 0x000E001C; //0000 0000 0000 1110 0000 0000 0001 1100
                NRF_GPIO->OUTSET = 0x00000010;
                printf("P0.04 = ON\r\n");
                break;
            case 3:
                NRF_GPIO->OUTCLR = 0x000E001C; //0000 0000 0000 1110 0000 0000 0001 1100
                NRF_GPIO->OUTSET = 0x00020000;
                printf("P0.17 = ON\r\n");
                break;
            case 4:
                NRF_GPIO->OUTCLR = 0x000E001C; //0000 0000 0000 1110 0000 0000 0001 1100
                NRF_GPIO->OUTSET = 0x00040000;
                printf("P0.18 = ON\r\n");
                break;
            case 5:
                NRF_GPIO->OUTCLR = 0x000E001C; //0000 0000 0000 1110 0000 0000 0001 1100
                NRF_GPIO->OUTSET = 0x00080000;
                printf("P0.19 = ON\r\n");
                break;
        }
    }
}
```



Embedded Studio Primer

Coding the BLE Peripheral

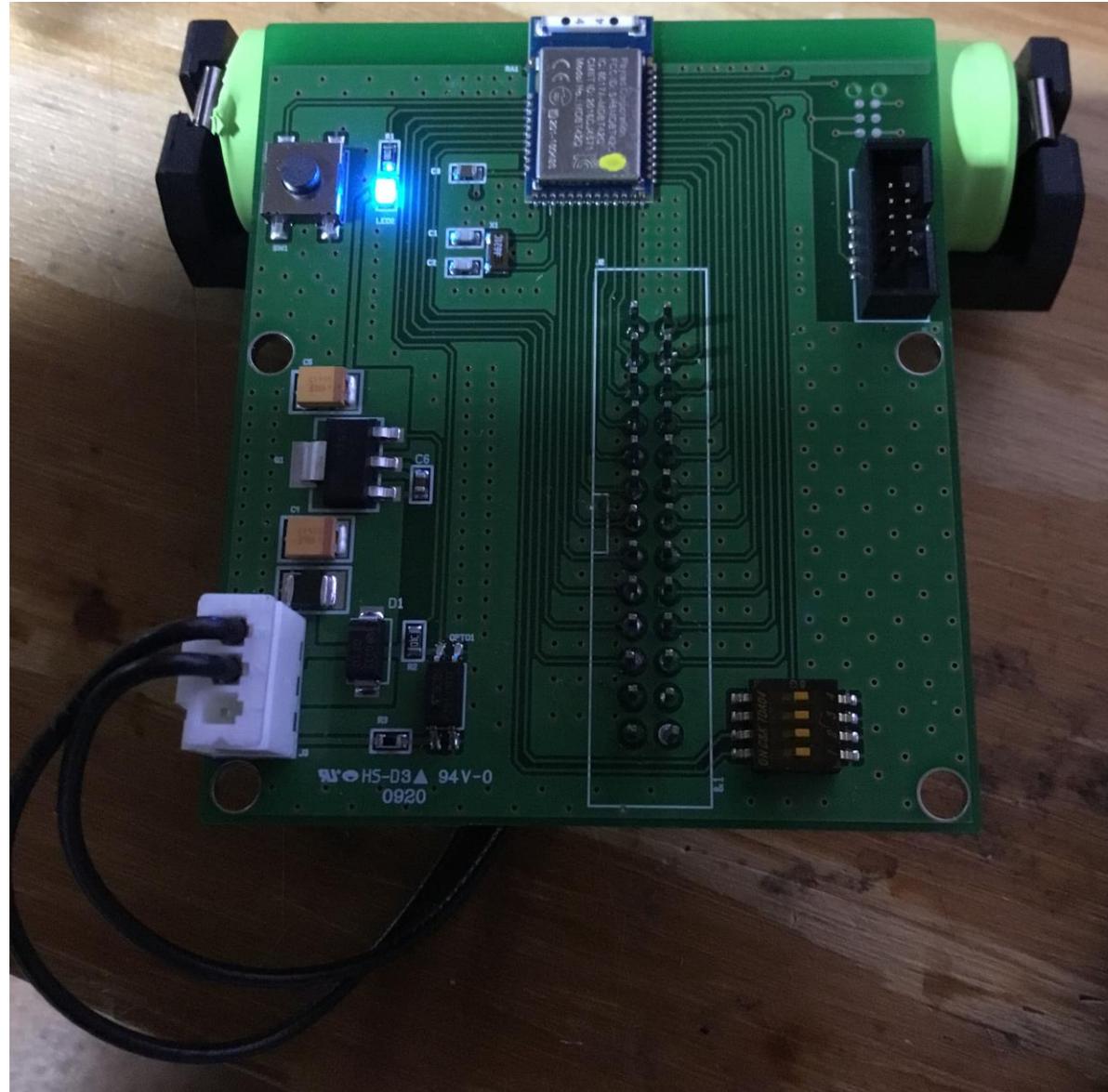
```
int main(void)
{
    bool erase_bonds;

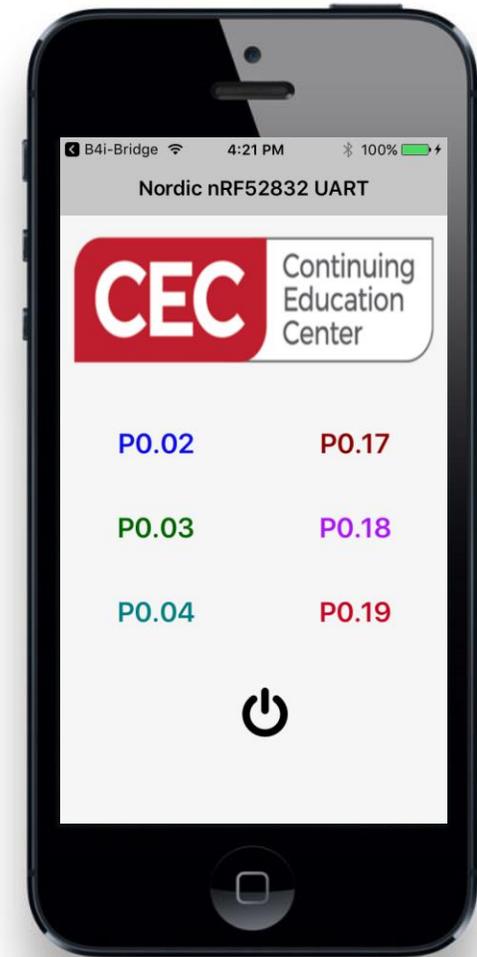
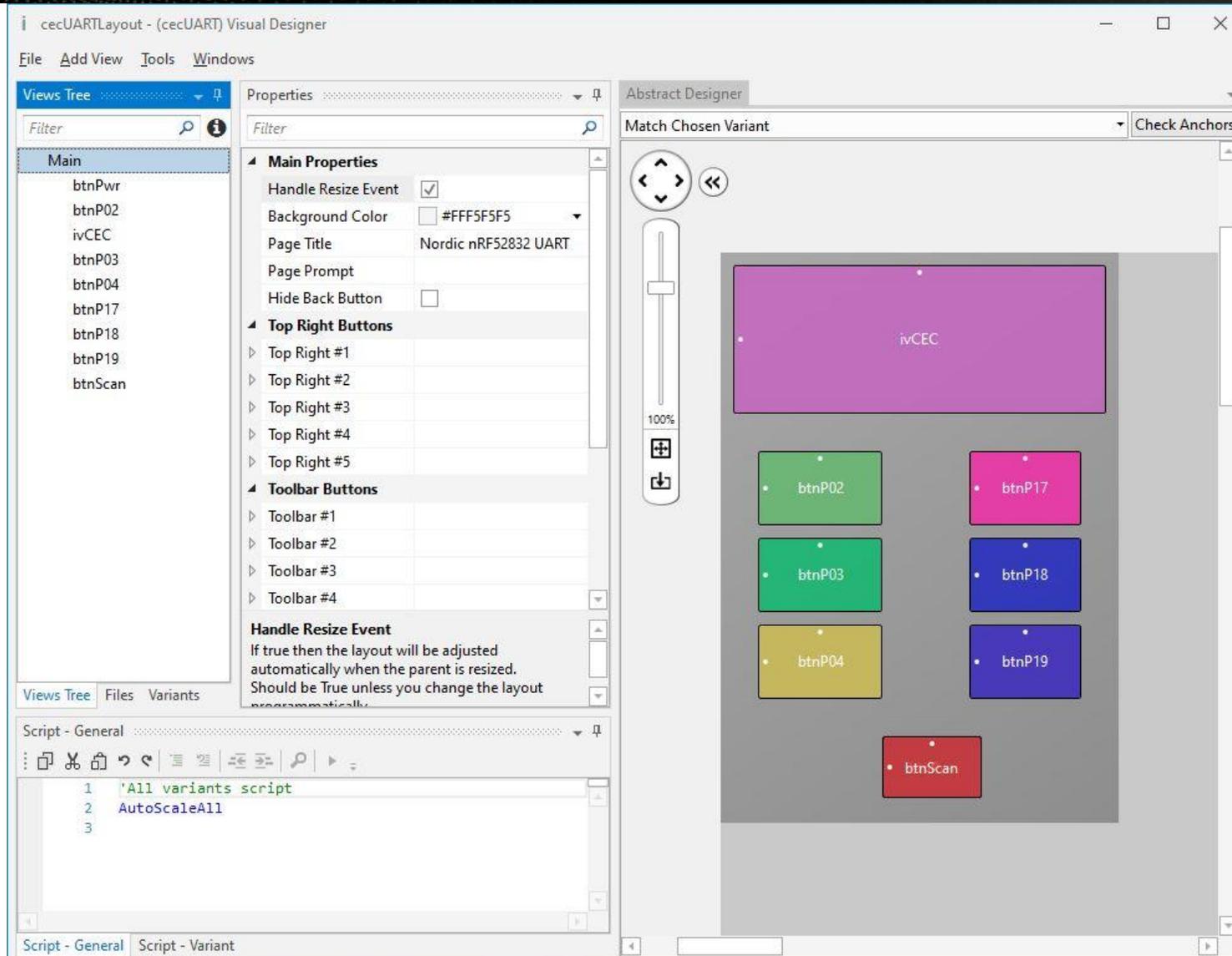
    // Initialize.
    uart_init();
    log_init();
    timers_init();
    buttons_leds_init(&erase_bonds);
    power_management_init();
    ble_stack_init();
    gap_params_init();
    gatt_init();
    services_init();
    advertising_init();
    conn_params_init();

    // Start execution.
    printf("\r\nUART started.\r\n");

    advertising_start();

    // Enter main loop.
    for (;;)
    {
        idle_state_handle();
    }
}
```





Sub Process_Globals

'These global variables will be declared once when the application starts.

'Public variables can be accessed from all modules.

Public App As Application

Public NavControl As NavigationController

Private Page1 As Page

Private manager As BleManager

Private ConnectedName As String

Private btnP02 As Button

Private btnP03 As Button

Private btnP04 As Button

Private btnP17 As Button

Private btnP18 As Button

Private btnP19 As Button

Private btnPwr As Button

Private btnScan As Button

Private ivCEC As ImageView

Dim nus_sid As String="6E400001-B5A3-F393-E0A9-E50E24DCCA9E"

Dim nus_rxc As String="6E400002-B5A3-F393-E0A9-E50E24DCCA9E"

'Dim nus_txc As String="6E400003-B5A3-F393-E0A9-E50E24DCCA9E"

Dim ledData(1) As Byte

End Sub



```
Sub btnScan_Click
manager.Scan(Null)
```

```
End Sub
```

```
Sub Manager_DeviceFound (Name As String, Id As String, AdvertisingData As Map, RSSI As Double)
```

```
Log("Found: " & Name & ", " & Id & ", RSSI = " & RSSI & ", " & AdvertisingData)
```

```
ConnectedName = Name
```

```
If ConnectedName = "Nordic_UART" Then
```

```
manager.StopScan
```

```
manager.Connect(Id)
```

```
End If
```

```
End Sub
```



```

➤ Found: , DF8A49B6-8167-4B45-89CA-C7D27A81D9FB, RSSI = -50, (read only map) {
➤ kCBAAdvDataIsConnectable = 0;
➤ kCBAAdvDataManufacturerData = <06000109 2002abf2 9dcee181 8a3bfd17 501d7ade 3855f8b0 810d
➤ }
➤ Found: , 42453C1C-66DB-4D96-9190-2D6DC6A48AFD, RSSI = -52, (read only map) {
➤ kCBAAdvDataIsConnectable = 1;
➤ kCBAAdvDataTxPowerLevel = 7;
➤ }
➤ Found: Nordic_UART, 92DBCCE4-E335-4E98-A9D1-8EFA52DF72DF, RSSI = -49, (read only map) {
➤ kCBAAdvDataIsConnectable = 1;
➤ kCBAAdvDataLocalName = "Nordic_UART";
➤ }

```

```
Discovering services
```

```
Services discovery completed.
```

```
➤ DataAvailable Fired
```

```
➤ P0.02 clicked
```

```
➤ P0.03 clicked
```

```
➤ P0.04 clicked
```

```
➤ P0.17 clicked
```

```
➤ P0.18 clicked
```

```
➤ P0.19 clicked
```

```
Sub Manager_Connected (services As List)  
manager.ReadData(nus_sid)  
SetState(True)  
End Sub
```

```
Sub SetState (connected As Boolean)  
btnScan.Enabled = Not(connected)  
btnScan.Visible = Not(connected)  
btnPwr.Visible = connected  
btnPwr.Enabled = connected  
btnP02.Enabled = connected  
btnP03.Enabled = connected  
btnP04.Enabled = connected  
btnP17.Enabled = connected  
btnP18.Enabled = connected  
btnP19.Enabled = connected  
End Sub
```



```
Sub btnP19_Click  
ledData(0) = 5  
manager.WriteData(nus_sid,nus_rxc,ledData)  
Log("P0.19 clicked")  
End Sub
```

```
Sub btnP18_Click  
ledData(0) = 4  
manager.WriteData(nus_sid,nus_rxc,ledData)  
Log("P0.18 clicked")  
End Sub
```

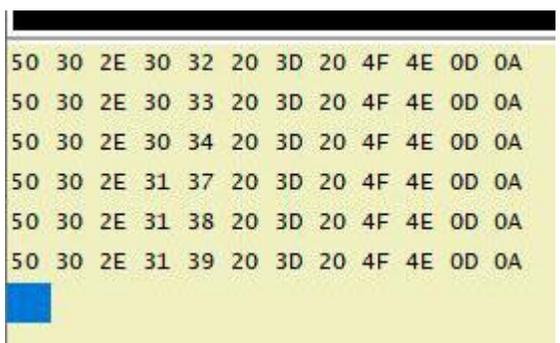
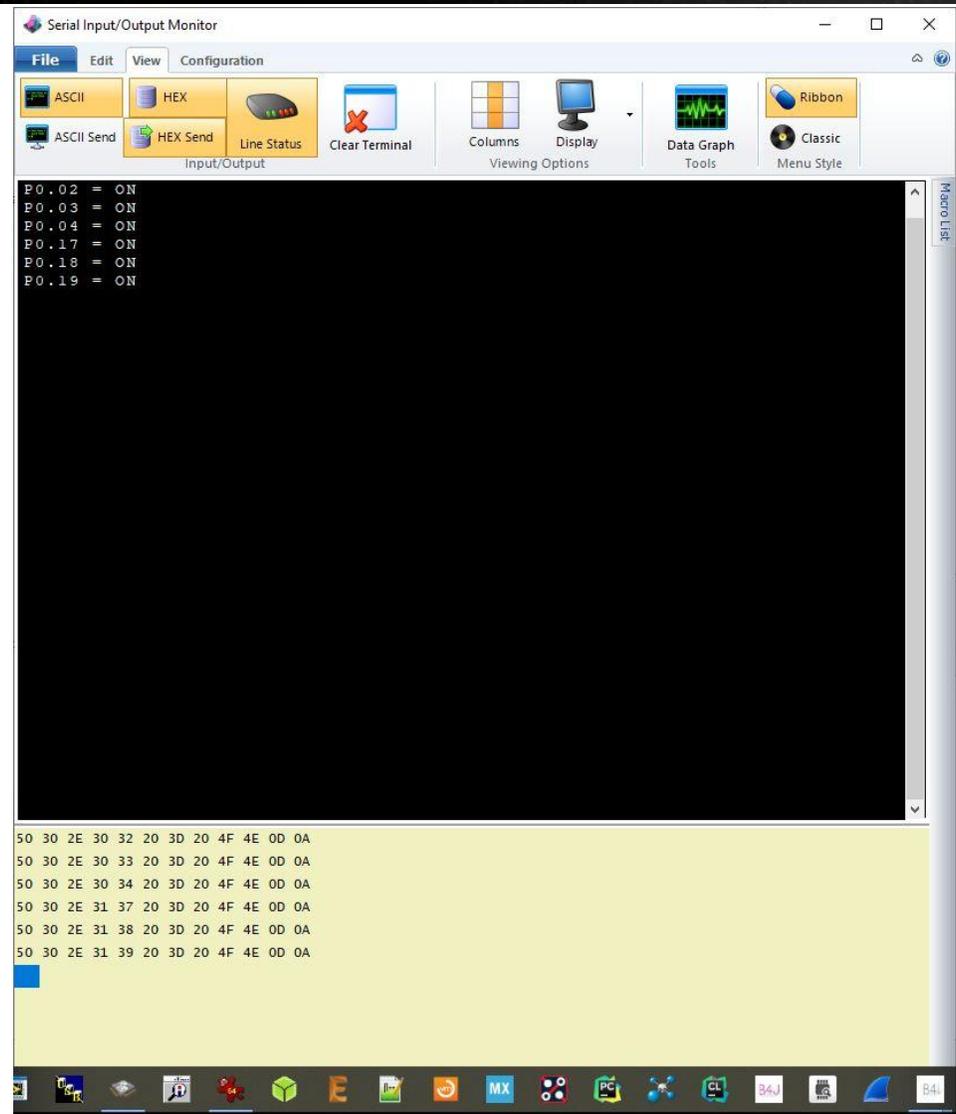
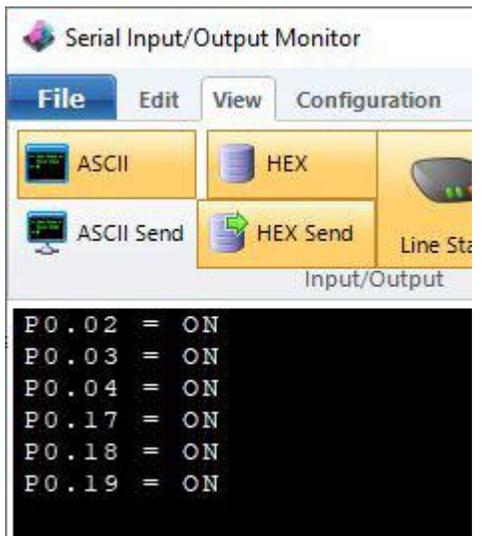
```
Sub btnP17_Click  
ledData(0) = 3  
manager.WriteData(nus_sid,nus_rxc,ledData)  
Log("P0.17 clicked")  
End Sub
```

```
Sub btnP04_Click  
ledData(0) = 2  
manager.WriteData(nus_sid,nus_rxc,ledData)  
Log("P0.04 clicked")  
End Sub
```

```
Sub btnP03_Click  
ledData(0) = 1  
manager.WriteData(nus_sid,nus_rxc,ledData)  
Log("P0.03 clicked")  
End Sub
```

```
Sub btnP02_Click  
ledData(0) = 0  
manager.WriteData(nus_sid,nus_rxc,ledData)  
Log("P0.02 clicked")  
End Sub
```

```
➤ Found: , DF8A49B6-8167-4B45-89CA-C7D27A81D9FB, RSSI = -50, (read only map) {  
➤   kCBAAdvDatalsConnectable = 0;  
➤   kCBAAdvDataManufacturerData = <06000109 2002abf2 9dcee181 8a3bfd17 501d7ade 3855f8b0 810d  
➤ }  
➤ Found: , 42453C1C-66DB-4D96-9190-2D6DC6A48AFD, RSSI = -52, (read only map) {  
➤   kCBAAdvDatalsConnectable = 1;  
➤   kCBAAdvDataTxPowerLevel = 7;  
➤ }  
➤ Found: Nordic_UART, 92DBCCE4-E335-4E98-A9D1-8EFA52DF72DF, RSSI = -49, (read only map) {  
➤   kCBAAdvDatalsConnectable = 1;  
➤   kCBAAdvDataLocalName = "Nordic_UART";  
➤ }  
Discovering services  
Services discovery completed.  
➤ DataAvailable Fired  
➤ P0.02 clicked  
➤ P0.03 clicked  
➤ P0.04 clicked  
➤ P0.17 clicked  
➤ P0.18 clicked  
➤ P0.19 clicked
```



Thank you for attending

Please consider the resources below:

- <https://www.raytac.com>
- <https://www.nordicsemi.com/Products/Low-power-short-range-wireless>
- <https://www.b4x.com/b4i.html>



DesignNews

Thank You

Sponsored by

