

13.蓝牙透传实验例程说明

13.1 开发板简介：

PGX-Nano 开发板搭载了一个 esp32 模块，利用此模块，可使板卡以 WIFI、蓝牙的方式进行通信，应注意的是，若想通过指令对 esp32 模块进行配置从而进入 wifi、蓝牙通信模式，需先对 esp32 模块进行 AT 固件的烧录，具体烧录步骤与烧录所需工程请参考 ESP32 固件烧录指南文档与 esp32_at 文件夹的 FPGA 工程。

13.2 开发板 ESP32 模组使用简介

PGX-Nano 搭载一个 esp32 模组，型号为：ESP32-WROVER-IE；此型号模组集成集成 ESP32 系列 ESP32-D0WD-V3 芯片，其两个 CPU 核都可以被单独控制，并且支持 wifi 与蓝牙通信。模组具有丰富的外设接口，开发板选用串口与 esp32 进行数据的交互，ESP32 将交互的数据通过 wifi/蓝牙发送或者接收，从而使板卡进行 wifi/蓝牙通信。

开发板与 ESP32 模组相连的 6 个信号的功能如下：

信号	功能
FPGA_TX	ESP32 的 UART 通信发送接口，用于与开发板的数据交互，ESP32 指令配置回复接口
FPGA_RX	ESP32 的 UART 通信接收接口，用于与开发板的数据交互，ESP32 指令配置发送接口
BT_INT	Boot 信号，板卡复位结束，或者上电时，会检测此信号的电平，检测到高电平时，进入通信状态，检测到低电平时，进入 AT 固件下载状态。
BOOTH_EN	ESP32 模组使能信号，高电平时 ESP32 模组正常工作，低电平时，ESP32 模组处于复位状态。
BT_TX	ESP32 模组 AT 固件下载对应 UART 发送接口。
BT_RX	ESP32 模组 AT 固件下载对应 UART 接收接口。

使用开发板上 ESP32 模模组进行 WIFI/蓝牙通信的具体思路如下：

首先：

ESP32 模块在使用前需要先烧录 AT 固件，烧录完毕后，才能正常使用指令配置 ESP32 模块。

烧录 AT 固件时，使用 ESP32 上 BT_TX、BT_RX 两个串口信号进行 AT 指令的烧录。将 ESP32 模组的 BT_TX、BT_RX 信号与板卡的串口信号 FPGA_UART_TX、FPGA_UART_RX 相连，则可以使用板卡串口为 ESP32 模组下载固件。

ESP32 有两种状态，AT 固件下载状态，正常使用状态，以上电时或者复位结束时 BT_INT 信号的电平高低，判断进入那种状态，BT_INT 为高电平时，ESP32 进入正常使用状态，BT_INT 为低电平时，ESP32 进入 AT 固件下载状态。由此可知，在 ESP32 模组上电后，控制使能信号 BOOTH_EN、BT_INT 信号，即可使 ESP32 进入 AT 固件下载状态。

其次：

使用开发板上的 ESP32 模组进行 WIFI/蓝牙通信时，需要先对模组配置相应的指令，使其进入通信状态，进入通信状态后，开发板即可以开始进行 wifi/蓝牙通信。

首先，配置指令时，可以将 ESP32 用于数据通信 UART 信号 FPGA_TX、FPGA_RX 与开发板串口信号 FPGA_UART_TX、FPGA_UART_RX 连接，将板卡串口与 PC 端进行连接，通过 PC 端串口调试助手对 ESP32 模组进行指令的配置。

其次，配置指令后，ESP32 模组就可以完成通过 wifi/蓝牙发送来自与串口的数据以及接收 WIFI/蓝牙的数据通过串口发送给开发板的功能，若希望使用板卡与 ESP32 进行数据的交互，就需要使用判断条件，断开 FPGA_TX、FPGA_RX 与开发板串口信号 FPGA_UART_TX、FPGA_UART_RX 的连接，使用 FPGA 直接向 ESP32 模组的 FPGA_RX 发送数据，或者接收 ESP32 模组的 FPGA_TX 数据，

在 WIFI/蓝牙透传状态下，在结束数据传输前，向 ESP32 模块发送不加换行灯信号的连续+++信号，即可退出透传状态。

13.3 实验目的：

使用板卡实现 UART 蓝牙透传。

13.4 实验要求：

根据实验步骤使 esp32 作为蓝牙服务器端进入蓝牙透传状态后，拨动拨码开关 SW0 开启板卡的蓝牙透传通信，按动按键 S2，板卡会通过蓝牙向蓝牙客户端发送一串 www.meyesemi.com 的字符，并且使用[蓝牙客户端](#)向板卡发送 0~9 的字符型数据，将会显示在板卡数码管上。（蓝牙客户端为手机，使用 APP：nRF Connect，提供的安装包在 tools 文件夹中）

13.5 实验原理：

透传原理：

透传即数据传输过程中，发送方发送数据的内容和长度与接收方接受到数据的内容和长度完全相同，不对数据进行任何处理。

13.6 实验源码：

顶层源码如下列所示（完整源码请前往实验例程查看）：

由于在使用 ESP32 模组在进行 WIFI/蓝牙透传前，需要先进行指令的配置，

因此设计一个拨动开关作为判断条件,在拨动开关为低电平时,板卡串口与 ESP32 模组数据传输对应串口相连,此时 PC 端串口调试助手输出的指令将同过板卡串口进入 ESP32 模组串口,从而实现指令的配置,在拨动开关为高电平时,由 FPGA 直接接管 ESP32 模组数据传输对应串口,按下按键 S2,板卡将发送一串字符,并且将 ESP32 模组数据传输串口传出的数据显示在数码管上。

传输结束,退出透传状态时,只需拨下拨动开关,根据程序设计,FPGA 会自动发送“+++”以结束透传状态。

```
module ble_test(  
    input wire    clk ,  
    input wire    rst_n ,  
    input wire    key ,  
    input wire    ctrl ,  
    input wire    fpga_uart_rx ,  
    input wire    fpga_tx ,  
    output reg     fpga_rx ,  
    output reg     fpga_uart_tx ,  
    output wire    en ,  
    output wire    boot ,  
    output wire [7:0]sel ,  
    output wire [15:0]seg  
);
```

```
wire tx ;  
wire [7:0]tx_data ;  
wire tx_pluse ;  
wire [7:0]rx_data ;  
wire rx_en ;  
wire tx_busy ;  
wire trsn_end ;  
wire rx_finish ;  
wire tx_end ;  
wire seg_flag ;  
wire [1:0]ctrl_flag ;
```

```
assign en = 1'b1 ;  
assign boot = 1'b1 ;
```

```
wire [1:0]key_flag ;  
assign key_flag[0] = ctrl ;  
assign key_flag[1] = key ;
```

//按键消抖, 对一个拨动开关、一个按键进行消抖处理

```
btn_deb#(  
    .BTN_WIDTH (4'd2),  
    .BTN_DELAY (26'hF423F)
```

```

)btn_deb
(
    .clk            (clk            ),
    .btn_in         (key_flag       ),
    .btn_deb        (ctrl_flag      )
);

```

//产生发送给 ESP32 数据传输对应串口的数据，退出透传时，需要发送“+++”，此模块检测拨动开关下降沿，检测到后自动发送“+++”结束透传状态

```

uart_data_gen uart_data_gen(
    .clk                (clk          ) ,
    .read_data          (rx_data      ) ,
    .ctrl_flag          (ctrl_flag) ,
    .tx_busy            (tx_busy      ) ,
    .write_max_num      (8'h14       ) ,
    .write_data         (tx_data      ) ,
    .write_en           (tx_pluse     ) ,
    .tran_end           (tran_end     )
);

```

//FPGA 产生发送给 ESP32 数据传输对应串口的 RX 端数据

```

uart_tx #(
    .BPS_NUM    (16'd434)
)uart_tx
(
    .clk                (clk          ) ,
    .tx_data            (tx_data      ) ,
    .tx_pluse           (tx_pluse) ,
    .tx_end             (tx_end       ) ,
    .uart_tx            (tx           ) ,
    .tx_busy            (tx_busy      )
);

```

//FPGA 接收来自于 ESP32 数据传输对应串口 TX 端数据

```

uart_rx # (
    .BPS_NUM    (16'd434)
)uart_rx
(
    .clk                (clk          ) ,

    .uart_rx            (fpga_tx      ) ,

    .rx_data            (rx_data      ) ,

    .rx_en              (rx_en        ) ,

    .rx_finish          (rx_finish    )
);

```

```
);
```

```
reg [1:0]state ;
parameter idle = 2'd0 ;
parameter state_send = 2'd1 ;
parameter state_end = 2'd2 ;
//状态机判断波动开关状态，波动开关为关时，板卡串口与 ESP32 模组数据传输
串口相连，高电平时，由 FPGA 直接控制 ESP32 模组数据传输串口
```

```
always @(posedge clk ) begin
    case (state)
        idle : begin
            if (ctrl_flag[0])
                state <= state_send ;
            else
                state <= idle ;
        end
        state_send : begin
            if (tran_end)
                state <= state_end ;
            else
                state <= state_send ;
        end
        state_end : begin
            if (tx_end)
                state <= idle ;
            else
                state <= state_end ;
        end
        default: state <= idle ;
    endcase
end
```

```
always @(posedge clk ) begin
    if (state != idle )
        fpga_rx <= tx ;
    else
        fpga_rx <= fpga_uart_rx ;
end
```

```
always @(posedge clk ) begin
    if (!seg_flag)
        fpga_uart_tx <= fpga_tx ;
    else
```

```
fpga_uart_tx <=1'b1 ;
end
```

```
assign seg_flag = (state != idle )? 1'b1 : 1'b0 ;
```

//在拨动开关高电平时，将从 ESP32 模组 TX 端接受到的 0~9 字符型数据显示在数码管上

```
seg_dynamic seg_dynamic(
    .clk          (clk ) ,
    .rst_n        (rst_n) ,
    .rx_data      (rx_data ) ,
    .rx_en        (rx_en ) ,
    .seg_flag     (seg_flag ) ,
    .dig          (sel ) ,
    .seg          (seg)
);
```

```
Endmodule
```

13.7 实验流程:

1、第一次使用，请烧录 AT 固件（esp32 的 AT 固件烧录的位置为 ESP 模块的 flash 中，因此只需烧录一次即可），详情请参考《esp32 固件烧录指南.pdf》文档。

2、连接板卡串口，烧录程序后，使用串口对 esp32 进行指令的配置，与此同时，将手机与 ESP32 模块通过蓝牙建立连接，此时 sw0 应为未拨动状态。（串口调试助手设置波特率应为 115200）

4、指令配置完成后，拨动拨动开关 sw0 开启板卡的 UART 透传通信，按动按键 S2，板卡会通过蓝牙发送一串 www.meyesemi.com 的字符，手机可以使用 APP 检测到相应数据，并且使用手机 APP 通过蓝牙发送 0~9 的字符型数据（取消勾选加回车换行），将会显示在板卡数码管上。

5、实验结束，拨下拨码开关 sw0，退出蓝牙透传状态。

13.7.1 详情参考下列叙述:

进行 AT 指令配置,进入 wifi 透传状态:

串口调试助手需要勾选加回车换行。



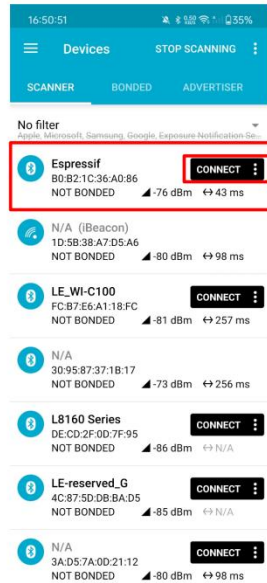
指令发送时前后不能有空格。

使用串口调试助手发送下列指令：

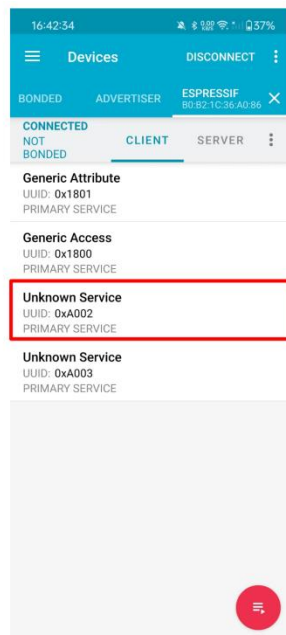
- (1) 指令: `AT+BLEINIT=2` // BLE 初始化, 将模块设置为 server
回复: OK
- (2) 指令: `AT+BLEGATTSSRVCRE` //GATTS 创建服务
回复: OK
- (3) 指令: `AT+BLEGATTSSRVSTART` //GATTS 开始服务
回复: OK
- (4) 指令:
`AT+BLEADVDATA="0201060A094573707265737369666030302A0"`
//设置广播设备名称 Espressif
回复: OK
- (5) 指令: `AT+BLEADVSTART` //开始广播
回复: OK
- (6) 指令: `AT+BLEGATTSCCHAR?` //查询 BLE 透传参数
回复: `+BLEGATTCCCHAR:"char",0,3,1,0xC300,2`
`+BLEGATTCCCHAR:"desc",0,3,1,1,0x2901`
`+BLEGATTCCCHAR:"char",0,3,2,0xC301,2`
`+BLEGATTCCCHAR:"desc",0,3,2,1,0x2901`
`+BLEGATTCCCHAR:"char",0,3,3,0xC302,8`
`+BLEGATTCCCHAR:"desc",0,3,3,1,0x2901`
`+BLEGATTCCCHAR:"char",0,3,4,0xC303,4`
`+BLEGATTCCCHAR:"desc",0,3,4,1,0x2901`
`+BLEGATTCCCHAR:"char",0,3,5,0xC304,8`
`+BLEGATTCCCHAR:"char",0,3,6,0xC305,16`
`+BLEGATTCCCHAR:"desc",0,3,6,1,0x2902`
`+BLEGATTCCCHAR:"char",0,3,7,0xC306,32`
`+BLEGATTCCCHAR:"desc",0,3,7,1,0x2902`
OK
- (7) 指令: `AT+BLESPPCFG=1,1,6,1,5` // 设置透传参数
回复: OK
(选择 1 号 service 的 6 号 characteristic 发送数据, 选择 1 号 service 的第 5 号 characteristic 接收数据, 可设置其他 characteristic 发送或者接收数据)

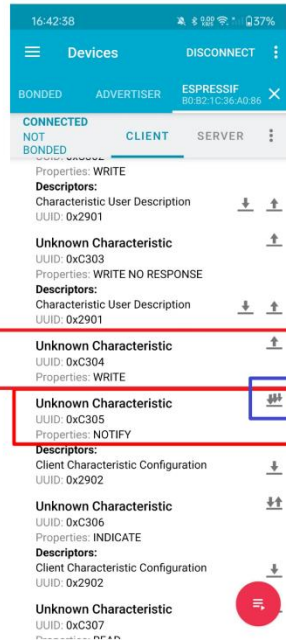
打开手机蓝牙调试助手 nRF Connect

选择 Espressif 选项点击 connect;



5 号 characteristic 对应 0xC304, 6 号 characteristic 对应 0x305, 找到相应位置, 开始 0x305 对应 characteristic 进行侦听





(8) 指令: AT+BLESPP

//开启 BLE 透传

回复: OK

完成 AT 指令配置过程如下:

```

[10:03:48.860]发→◇AT+BLEINIT=2
□
[10:03:48.864]收←◆AT+BLEINIT=2

[10:03:50.000]收←◆
OK

[10:03:55.908]发→◇AT+BLEGATTSSRVCRE
□
[10:03:55.913]收←◆AT+BLEGATTSSRVCRE

[10:03:55.972]收←◆
OK

[10:03:57.500]发→◇AT+BLEGATTSSRVSTART
□
[10:03:57.505]收←◆AT+BLEGATTSSRVSTART

OK

[10:03:58.700]发→◇AT+BLEADVDATA="0201060A09457370726573736966030302A0"
□
[10:03:58.711]收←◆AT+BLEADVDATA="0201060A09457370726573736966030302A0"

OK

[10:04:00.148]发→◇AT+BLEADVSTART
□
[10:04:00.152]收←◆AT+BLEADVSTART

OK

[10:04:01.300]发→◇AT+BLEGATTCHAR?
□
[10:04:01.305]收←◆AT+BLEGATTCHAR?

[10:04:01.338]收←◆+BLEGATTCHAR: "char", 1, 1, 0xC300, 0x02
+BLEGATTCHAR: "desc", 1, 1, 1, 0x2901
+BLEGATTCHAR: "char", 1, 2, 0xC301, 0x02
+BLEGATTCHAR: "desc", 1, 2, 1, 0x2901
+BLEGATTCHAR: "char", 1, 3, 0xC302, 0x08
+BLEGATTCHAR: "desc", 1, 3, 1, 0x2901
+BLEGATTCHAR: "char", 1, 4, 0xC303, 0x04
+BLEGATTCHAR: "desc", 1, 4, 1, 0x2901
+BLEGATTCHAR: "char", 1, 5, 0xC304, 0x08
+BLEGATTCHAR: "char", 1, 6, 0xC305, 0x10
+BLEGATTCHAR: "desc", 1, 6, 1, 0x2902
+BLEGATTCHAR: "char", 1, 7, 0xC306, 0x20
+BLEGATTCHAR: "desc", 1, 7, 1, 0x2902
+BLEGATTCHAR: "char", 1, 8, 0xC307, 0x02
+BLEGATTCHAR: "desc", 1, 8, 1, 0x2901
+BLEGATTCHAR: "char", 2, 1, 0xC400, 0x02
+BLEGATTCHAR: "desc", 2, 1, 1, 0x2901
+BLEGATTCHAR: "char", 2, 2, 0xC401, 0x02
+BLEGATTCHAR: "desc", 2, 2, 1, 0x2901

OK

[10:04:04.396]发→◇AT+BLESPPCFG=1, 1, 6, 1, 5
□
[10:04:04.402]收←◆AT+BLESPPCFG=1, 1, 6, 1, 5

OK

```

当手机与 esp32 连接后，esp32 模块会通过串口反馈一些信息

```

[10:04:28.050]收←◆+BLECONNPARAM:0,0,0,6,0,500
[10:04:29.753]收←◆+BLECONNPARAM:0,0,0,24,0,500

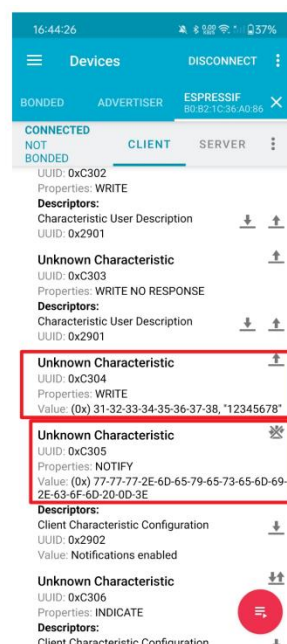
```

最后开启透传

```
[10:04:50.900]发→◇AT+BLESPP
[10:04:50.904]收←◆AT+BLESPP
OK
```

13.8 实验现象

指令配置完成并且 ESP32 模块与手机建立蓝牙连接后，拨动拨动开关 sw0 开启板卡的蓝牙透传通信，按动按键 S2，板卡会通过蓝牙发送一串 `www.meyesemi.com` 的字符，通过手机 APP 可以侦测到相应字符对应 ASCII 码数据，并且使用手机发送 0~9 的字符型数据（取消勾选加回车换行），将会显示在板卡数码管上。



提示：

- 1、更多 at 指令，可前往 [esp32 模块厂商乐鑫科技官网](#) 获取。