AT功能测试：

测试固件获取[[ingchips/AT\_UART: at uart (github.com)](https://github.com/ingchips/AT_UART)]

AT指令说明文档下载：[《桃芯串口透传指令与使用说明.docx》](https://github.com/ingchips/AT_UART/blob/master/9168_AT/doc/%E6%A1%83%E8%8A%AF%E4%B8%B2%E5%8F%A3%E9%80%8F%E4%BC%A0%E6%8C%87%E4%BB%A4%E4%B8%8E%E4%BD%BF%E7%94%A8%E8%AF%B4%E6%98%8E.docx)

* 1. 主从一体功能(至少 10 个链路，主从数量可配置，例如1主9 从，2主8 从);

测试工程：[[ingchips/AT\_UART: at uart (github.com)](https://github.com/ingchips/AT_UART)

2主8从：

准备1个设备作为测试设备A，另外准备8个设备作为设备A的从设备，准备两部手机作为设备A的主设备,A烧录AT的程序；

测试1：多链接下，A发送数据到从机

测试2：多链接下，A接收来自从机发送的数据

测试3：多链接下，A发送数据到手机

测试4：多链接下，A接收来自手机发送的数据

所有设备上电会默认开启广播，连接成功后自动关闭广播

设备A通过指令连接8个从设备

第一步：AT+SCAN扫描广播，5代表扫描持续5秒

|  |
| --- |
| [15:29:55.524]发→◇AT+SCAN=5  □  [15:30:00.528]收←◆  +SCAN:ON  OK  No: 0 Addr:F1F25D13D414 Type:1 Rssi:-67dBm  No: 1 Addr:F1F2247A4A5D Type:1 Rssi:-25dBm  No: 2 Addr:F1F21C223DA1 Type:1 Rssi:-38dBm  No: 3 Addr:F1F26E164811 Type:1 Rssi:-62dBm  No: 4 Addr:F1F26631FE99 Type:1 Rssi:-68dBm  No: 5 Addr:F1F212F38975 Type:1 Rssi:-82dBm  No: 6 Addr:F1F20A32B6B5 Type:1 Rssi:-70dBm  No: 7 Addr:F1F260CED041 Type:1 Rssi:-68dBm |

第二部：AT+CONN依次建立连接，参数对应扫描到的设备序号

注意：扫描设备序号和链接号意义不同

|  |
| --- |
| [15:32:02.821]发→◇AT+CONN=0  □  [15:32:03.106]收←◆  +BLECONN:0,F1:F2:5D:13:D4:14  [15:32:06.546]发→◇AT+CONN=1  □  [15:32:07.453]收←◆  +BLECONN:1,F1:F2:24:7A:4A:5D  。。。。。。  [15:32:21.016]发→◇AT+CONN=7  □  [15:32:21.175]收←◆  +BLECONN:7,F1:F2:60:CE:D0:41 |

默认连接间隔为350unit对应时间为437.5ms（unit \* 1.25），针对多连接情况下调整的间隔较大，可以通过指令AT+CONN\_PARAM修改指定链接的连接间隔，可根据实际情况设置（当某一链接有大量数据需要传输时，可以将该链接的连接间隔调小，以达到提高数据吞吐率的效果）。

两部手机使用蓝牙应用（测试时使用nRF Connect）连接测试设备

设备被连接时上报信息：

|  |
| --- |
| [15:36:59.715]收←◆  +BLECONN:8,44:FD:37:1C:14:D5  [15:43:55.852]收←◆  +BLECONN:9,75:0F:65:8E:B0:D9 |

注意：连接设备A前需要先通过AT+MODE=B开启设备广播，由于被作为从设备连接成功后，设备会自动关闭广播，所以两部手机连接需要分别先通过指令开启两次广播

通过指令查看当前所有链接，一共10个链接，0~7范围内为从机链接，8和9号是主机链接（两部手机）

|  |
| --- |
| [15:47:22.130]发→◇AT+LINK?  □  [15:47:22.135]收←◆  +BLECONN:0,F1:F2:5D:13:D4:14  +BLECONN:1,F1:F2:24:7A:4A:5D  +BLECONN:2,F1:F2:1C:22:3D:A1  +BLECONN:3,F1:F2:6E:16:48:11  +BLECONN:4,F1:F2:66:31:FE:99  +BLECONN:5,F1:F2:12:F3:89:75  +BLECONN:6,F1:F2:0A:32:B6:B5  +BLECONN:7,F1:F2:60:CE:D0:41  +BLECONN:8,44:FD:37:1C:14:D5  +BLECONN:9,75:0F:65:8E:B0:D9 |

使用AT指令发送数据

内置INGChips Console Service有Generic Input和Generic Output特征，分别用于主向从/从向主传输数据，使用Generic Output特征需要先使能Notify，由于程序默认在连接时就自动查找Generic Output特征并使能了Notify，所以可以直接从机发送数据到主机。

Generic Input特征的UUID为6E400002-B5A3-F393-E0A9-E50E24DCCA9E

Generic Output特征的UUID为6E400003-B5A3-F393-E0A9-E50E24DCCA9E

通过AT+BLEGATTC查看1号从机GATT Server Profile，并确定Generic Input/Output特征的value handle号：（特征的value handle在后续传输数据的AT指令中作为指令参数用到）

|  |
| --- |
| AT+BLEGATTC=1  □  [15:56:28.644]收←◆  +BLEGATTCPRIMSRV:1,1,6,6E400001-B5A3-F393-E0A9-E50E24DCCA9E  +BLEGATTCCHAR:1,5,6,6,4,6E400002-B5A3-F393-E0A9-E50E24DCCA9E  +BLEGATTCCHAR:1,2,4,3,18,6E400003-B5A3-F393-E0A9-E50E24DCCA9E  +BLEGATTCDESC:1,4,2902  +BLEGATTCPRIMSRV:1,7,13,3345C2F0-6F36-45C5-8541-92F56728D5F3  +BLEGATTCCHAR:1,12,13,13,6,3345C2F2-6F36-45C5-8541-92F56728D5F3  +BLEGATTCCHAR:1,10,11,11,4,3345C2F3-6F36-45C5-8541-92F56728D5F3  +BLEGATTCCHAR:1,8,9,9,2,3345C2F1-6F36-45C5-8541-92F56728D5F3  +BLEGATTCC:1,0 |

其中+BLEGATTCCHAR开头的列出了特征信息，格式为+BLEGATTCCHAR:link\_id,start\_handle,end\_handle,value\_handle,properties,UUID，可以通过UUID号确定Generic Input/Output特征的value handle分别是6和3

* + 1. 测试1：多链接下，发送数据到从机

使用链接1号，Generic Input，主向从发送字符数据”1234”（对应十六进制编码：31 32 33 34）

|  |
| --- |
| 主：指令输入方  [16:08:24.252]发→◇AT+BLEGATTCWR=1,6,31323334  □  [16:08:24.258]收←◆  OK |
| 从：  [16:08:24.367]收←◆1234 |

测试通过

* + 1. 测试2：多链接下，接收来自从机发送的数据

使用链接1号，Generic Output，从向主发送字符数据”1234”（对应十六进制编码：31 32 33 34）

先使能Notify，由于程序在连接从机时自动使能了Generic Output特征的Notify所以，不需要额外的指令来配置使能，直接发送就行。

发送

|  |
| --- |
| 从：指令输入方  [16:12:25.392]发→◇AT+BLEGATTSWR=8,3,0,31323334  □  [16:12:25.396]收←◆notify:0 3  OK |
| 主：  [16:12:25.432]收←◆1234 |

测试通过

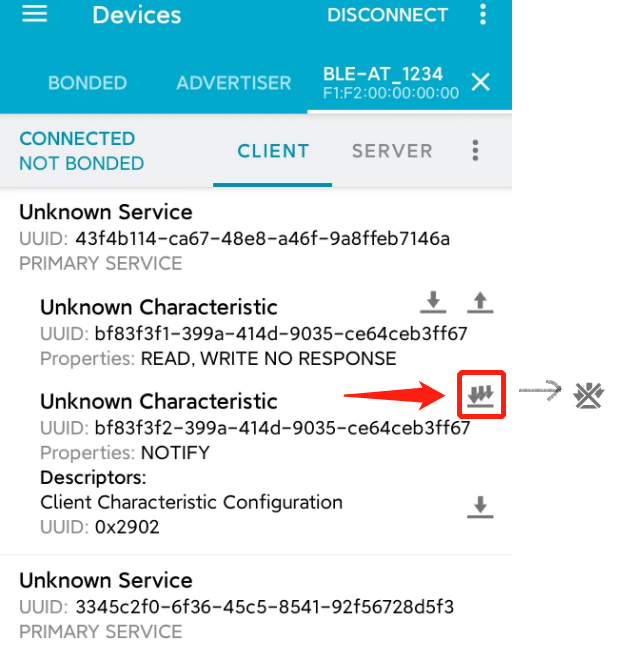
使用手机BLE应用发送数据

* + 1. 测试3：多链接下，发送数据到主机

使用链接8号，向主机发送字符数据”1234”（对应十六进制编码：31 32 33 34）

先使能Notify

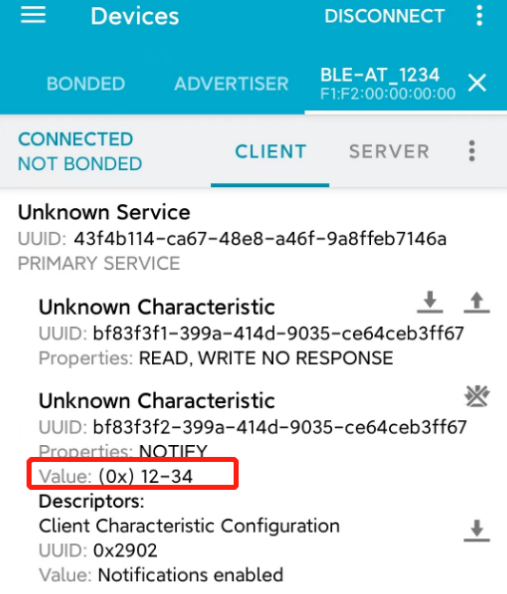
点击下图红框圈出的按钮，点击之后图标变化成右侧的样子



发送数据到主机：

|  |
| --- |
| [16:14:25.814]发→◇AT+BLEGATTSWR=8,3,0,31323334  □  [16:14:25.820]收←◆notify:8 3  OK |

同时通过nRFConnect查看到收到测试设备发来的数据

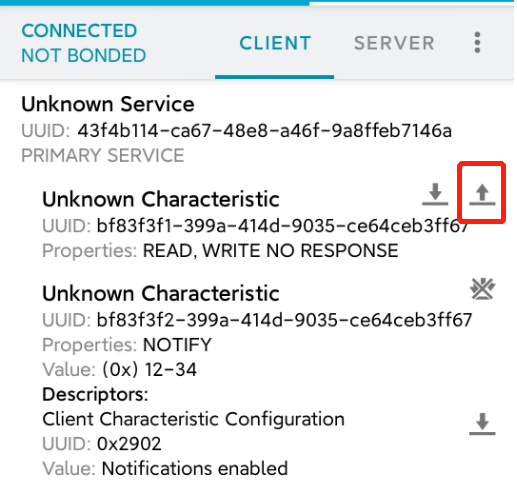
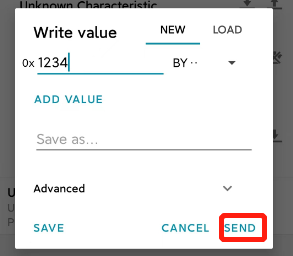


测试通过

* + 1. 测试4：多链接下，接收来自主机发送的数据

使用nRFConnect发送数据：

点击左图中红框圈中的按钮，弹出发送数据编辑界面（右图所示），输入数据0x1234，最后点击右下角的SEND按钮发送

同时测试设备端打印接收数据：

|  |
| --- |
| [16:15:07.007]收←◆1234 |

测试通过

* 1. 支持两种低功耗模式，通过 BLE 主机唤醒或者外部引脚唤醒;
     1. Shutdown模式

进入关机模式：

|  |
| --- |
| [19:02:24.862]发→◇AT+SHUTDOWN  □ |

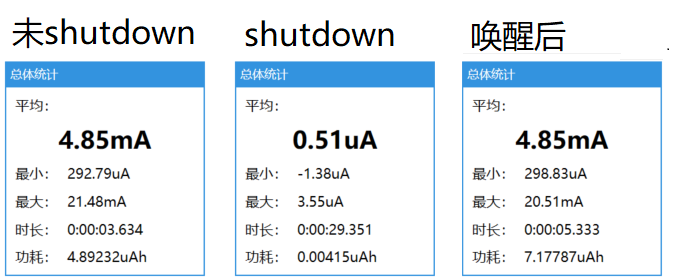
发送任意指令验证是否进入关机模式：无响应数据表示进入关机模式成功

|  |
| --- |
| [19:03:12.185]发→◇AT+MODE?  □ |

退出关机模式（唤醒）：拉高GPIO0 收到响应数据MAIN\_OK表示唤醒成功

|  |
| --- |
| [19:06:29.804]收←◆MAIN\_OK |

功耗变化：



* + 1. 低功耗模式

进入低功耗模式：

|  |
| --- |
| [17:20:00.689]发→◇AT+SLEEP=S  □  [17:20:00.737]收←◆+SLEEP:S  OK |

发送任意指令验证是否进入低功耗模式：无响应数据表示进入低功耗模式成功

|  |
| --- |
| [19:09:09.452]发→◇AT+MODE?  □ |

退出低功耗模式：

1. 拉高GPIO0进行唤醒
2. 唤醒状态下输入AT+SLEEP:E退出低功耗模式

|  |
| --- |
| [17:23:38.559]发→◇AT+SLEEP=E  □  [17:23:38.599]收←◆+SLEEP:E  OK |

不拉高唤醒引脚的话，GPIO掉电，将无法处理AT指令输入

功耗变化：



注意：由于程序默认会开启广播，上图结果是先通过AT+MODE=I关闭广播后测出的功耗，在开启广播的情况下平均功耗会更高

在开启200ms间隔的广播的情况下的平均功耗：

* 1. 支持 OTA 升级

准备升级的固件v2.bin

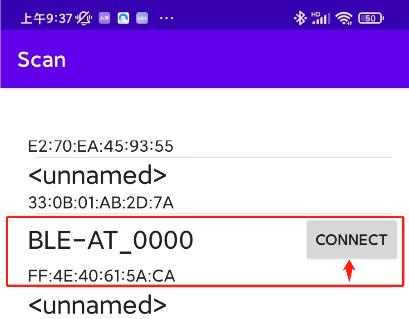
安装Android版的FOTA应用

[<https://github.com/ingchips/fota_for_android/releases/download/v1.2/fota.apk>]

输入AT指令开启设备广播

|  |
| --- |
| [09:36:38.452]发→◇AT+MODE=B  □  [09:36:38.456]收←◆OK |

FOTA应用搜索设备并连接



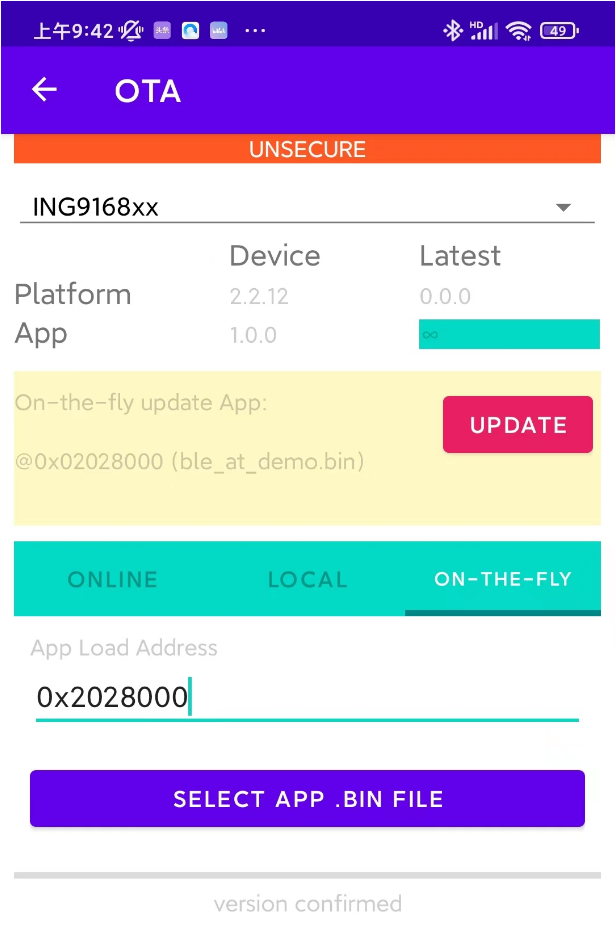
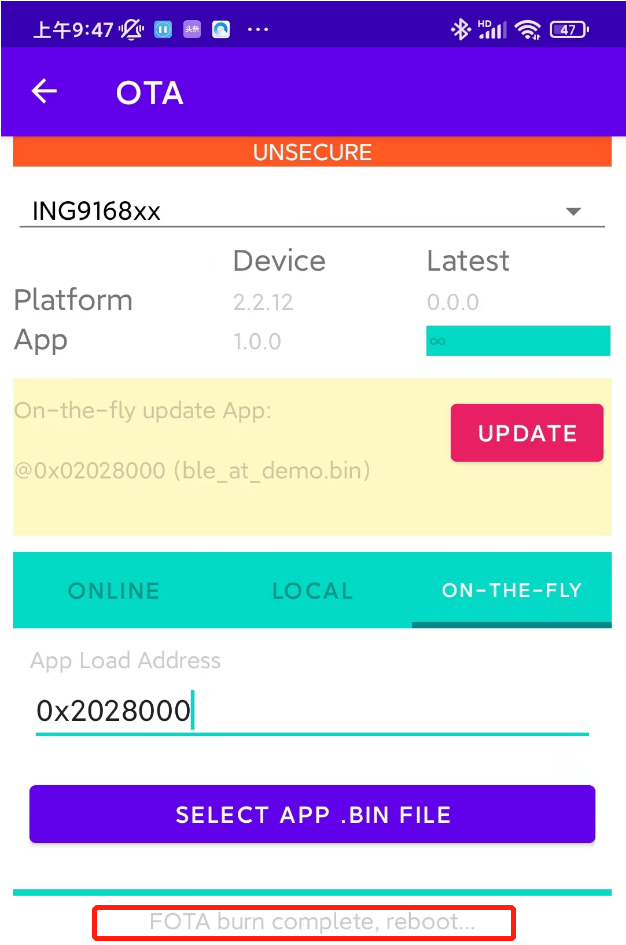
顶部下拉框选择系列：ING9168xx

绿色选项卡选择升级模式：ON-THE-FLY

App Load Address填写值：0x2028000

点击SELECT APP .BIN FILE按钮选择固件二进制文件

文件选择完后出现UPDATE按钮，点击UPDATE开始执行FOTA

->

执行完成后验证FOTA是否成功，我这里简单起见将复位时的日志多加了一句作为版本2，版本1复位时打印一个OK，版本2复位时打印两个OK，测试复位时打印两个OK，FOTA升级成功

|  |
| --- |
| [09:44:58.377]收←◆OK  OK |

* 1. 支持 AT 指令，读取/设置模块常规参数、进入低功耗模式等，可配置以下关键参数:发射功率、串口波特率、广播间隔、连接间隔、UUID 等;
     1. 读取/设置设备地址：

|  |
| --- |
| 读取：  [16:23:08.821]发→◇AT+MAC?  □  [16:23:08.827]收←◆  +MAC:F1F27563404F  OK |
| 设置：  [16:23:39.462]发→◇AT+MAC=F1F27563404E  □  [16:23:39.469]收←◆  +MAC:F1F27563404E  OK  读取验证是否设置成功：  [16:23:52.623]发→◇AT+MAC?  □  [16:23:52.628]收←◆  +MAC:F1F27563404E  OK |

* + 1. 读取/设置广播数据：

读取：

|  |
| --- |
| [18:55:18.138]发→◇AT+BLEADVDATA  □  [18:55:18.141]收←◆+BLEADVDATA:"0201060C09424C452D41545F30303030" |

设置：

准备adv data：

|  |
| --- |
| // 0x01 - «Flags»  2, 0x01,  0x06,  // 0x09 - «Complete Local Name»  12, 0x09,  0x42, 0x4C, 0x45, 0x2D, 0x41, 0x54, 0x5F, 0x31,  0x32, 0x33, 0x34,  // Total size = 16 bytes |

设置：

|  |
| --- |
| [18:56:54.680]发→◇AT+BLEADVDATA="0201060C09424C452D41545F31323334"  □  [18:56:54.685]收←◆OK |

读取验证是否设置成功：

|  |
| --- |
| [18:57:33.586]发→◇AT+BLEADVDATA  □  [18:57:33.590]收←◆+BLEADVDATA:"0201060C09424C452D41545F31323334" |

* + 1. 设置UART波特率：

|  |
| --- |
| [16:24:30.409]发→◇AT+UART=921600,8,0,1  □  [16:24:30.416]收←◆  +UART:921600,8,0,1  OK |

串口助手软件切换到921600输入任意AT指令验证波特率是否设置成功：有响应数据表示波特率设置成功

|  |
| --- |
| [16:25:31.275]发→◇AT+MODE?  □  [16:25:31.277]收←◆  +MODE:I  OK |

* + 1. 设置广播间隔/发射功率：

例如设置广播间隔100ms，因为设置广播间隔API传参的单位是625us所以需要先进行换算100ms则需要设置值160

查看广播间隔/发射功率：结果中256即为当前广播间隔，3即为发射功率(单位dbm)

|  |
| --- |
| [11:19:12.587]发→◇AT+BLEADVPARAM  □  [11:19:12.589]收←◆+BLEADVPARAM:256,256,19,1,7,0,1,00:00:00:00:00:00,3  OK |

设置广播间隔/发射功率：

|  |
| --- |
| [11:06:50.303]发→◇AT+BLEADVPARAM=160,160,19,1,7,0,1,"00:00:00:00:00:00",2  □  [11:06:50.308]收←◆OK |

* + 1. 设置连接间隔：

例如设置连接间隔100ms，因为设置连接间隔API传参的单位是1.25ms所以需要先进行换算100ms则需要设置值80

设置连接间隔：

|  |
| --- |
| [16:27:40.743]发→◇AT+CONN\_PARAM=1,80,0,40  □  [16:27:40.748]收←◆  +CONN\_PARAM:1,80,0,40  OK |

等待上报信息

|  |
| --- |
| [16:27:50.311]收←◆  +CONN\_PARAM: interval 100.00 ms  OK |

* + 1. 设置UUID：

可以通过SDK内置的工具生成GATT服务并同时设置UUID

具体使用见：[4 核心工具 | INGCHIPS SDK 开发者用户手册](https://ingchips.github.io/user_guide_cn/core-tools.html)

除了使用工具生成的静态数据外，还可以使用SDK内置的API来动态生成GATT服务端数据。API使用可参考SDK Example：Thermometer with FOTA