

JM-WIFI-DTU

JSON 协议说明书

面向设备购买者、应用开发者与系统集成商

文档项目	内容
版权所有	临沂市景墨嘉铭科技有限公司
协议版本	JSON V2.0
适用通道	MQTT、BLE、调试串口
字符编码	UTF-8
消息格式	纯 JSON，不包含二进制帧
发布日期	2026-08-25
更新日期	2026-08-30
状态	C2 固件 7.0.0 代码同步版
适用产品	JM-WT24U(S) / JM-WT25(S)
文档版本	Ver 2.0

目录

1. 协议设计	4
1.1 为什么采用三种消息	4
1.2 请求格式	4
1.3 成功响应	4
1.4 失败响应	4
1.5 主动事件	5
2. 通用规则	5
2.1 传输边界	5
2.2 用户 MQTT 下发与响应流程	6
2.3 action 总表	7
3. 查询与配置接口	8
3.1 查询设备信息	8
3.2 设置 Wi-Fi	10
3.3 设置心跳	11
3.4 设置蓝牙名称	12
3.5 设置串口	12
3.6 设置 MQTT 连接	13
3.7 设置 MQTT 主题	14
3.8 设置 Wi-Fi 超时	15
3.9 设置工作模式	15
3.10 查询配置代号	16
3.11 设置配置代号	16
4. 设备控制接口	17

4.1 恢复出厂设置	17
4.2 重启设备	17
5. OTA 升级接口	18
5.1 开始 BLE OTA	18
5.2 取消 OTA	19
5.3 URL OTA	19
5.4 OTA 主动事件	20
6. 其他主动事件	21
7. 失败响应规范	22
8. 固件实现要求	错误！未定义书签。
8.1 消息识别伪代码	错误！未定义书签。
9. 消费者联调清单	错误！未定义书签。
附录 A：JSON Schema 摘要	24
附录 B Web Bluetooth 特征与透传方向	25

1. 协议设计

1.1 为什么采用三种消息

消息方向	唯一识别字段	结构	用途
消费者→设备	action	action + data	发起查询或设置
设备→消费者	success	success + message + data	返回上一条请求的处理结果
设备主动上报	event	event + data	心跳、OTA 进度、状态变化

同一对象不会同时出现 action、success 和 event。消费者发送一条请求后，应等待对应响应再发送下一条请求。这样无需命令号和请求编号，也不会产生字段重复。

1.2 请求格式

```
{
  "action": "set_uart",
  "data": {
    "baud": 9600,
    "dataBits": 8,
    "parity": "none",
    "stopBits": 1
  }
}
```

1.3 成功响应

```
{
  "success": true,
  "message": "uart_configured",
  "data": {}
}
```

1.4 失败响应

```
{ "success": false, "message": "invalid_baud_rate", "data": { "field": "baud", "reason":
"invalid_parameter" } }
```

1.5 主动事件

```
{
  "event": "ota_progress",
  "data": {
    "percent": 35,
    "receivedBytes": 311617,
    "totalBytes": 890336
  }
}
```

2. 通用规则

- 所有管理消息必须是一个完整 JSON 对象，字符编码为 UTF-8。MQTT Payload 承载完整对象；调试串口以 LF 结束；BLE FFE1 允许按 MTU 跨多次 Write 重组。用户 MQTT 中的非保留管理 Payload 按原始透传数据处理。
- 所有字段名区分大小写，统一使用小驼峰命名。
- data 必须是对象；无参数或无返回数据时使用空对象 {}。
- success 只有 true 和 false。
- 失败时 data 包含 reason，并在可定位到字段时包含 field。
- 消费者必须按“一次请求、一次响应”顺序通信；设备处理完成前不得并发发送管理请求。

2.1 传输边界

通道	消息边界	建议上限	说明
MQTT	一个 Payload 承载一条完整 JSON 或一段透传数据	JSON 3072 bytes	监管 MQTT 每个 Payload 必须是一条完整 JSON；用户 MQTT 仅识别两个 jmwifidtu_ 专用 action，其余 Payload 原样透传
串口	一次写入一条完整 JSON	JSON 4096 bytes	每条紧凑 JSON 整体写入
BLE	JSON 3072 bytes	FFE1 可跨多个 Write	分片间 CR/LF 会在对

通道	消息边界	建议上限	说明
		接收一条 JSON；设备按字符串/转义感知的括号深度判断最外层对象闭合，末尾 LF 可选	象未闭合时忽略；完整对象闭合后立即提交；5 秒未完成或超过 4096 字节时清空

使用提示： BLE 请求可通过一次或多次 FFE1 Write 提交。接收器在最外层 JSON 对象闭合时只提交一次，兼容仅最后一片带 LF 或每片附带 CR/LF。FFE2 响应/事件按协商 MTU 分片 Notify，并在完整 JSON 后额外发送一个 LF 通知；网页端应重组后再解析。

2.2 用户 MQTT 下发与响应流程

用户 MQTT 的订阅主题同时承载设备管理和串口透传。。只有 action 等于 jmwifidtu_factory_reset 或 jmwifidtu_reboot 时才进入管理流程；其他任何 Payload 均不作为管理命令处理。

- 用户 MQTT 只开放恢复出厂和清除 Wi-Fi 后重启两项管理动作。
- 监管 MQTT 是独立管理通道，继续使用 factory_reset、reboot 等标准 action。

Action	说明	Payload/处理	结果
/	其他 Payload	不执行 JSON 管理协议	字节保持不变并写入透传串口
jmwifidtu_factory_reset	用户 MQTT 恢复出厂 (专用)	rebooting	设备恢复出厂
jmwifidtu_reboot	用户 MQTT 清除 Wi-Fi 并重启 (专用)	rebooting	设备清除 WiFi 并重启

用户 MQTT 恢复出厂请求

```
{
  "action": "jmwifidtu_factory_reset",
  "data": {"confirm": true}
```

```
}

```

用户 MQTT 清除 Wi-Fi 并重启请求

```
{
  "action": "jmwifidtu_reboot",
  "data": {}
}

```

2.3 action 总表

action	用途	响应数据
get_device_info	查询设备信息	完整设备配置
set_wifi	设置 Wi-Fi	配置保存状态；连接完成后主动事件
set_ble_heartbeat	设置 BLE 心跳	空对象
set_ble_name	设置蓝牙名称	空对象
set_uart	设置串口参数	空对象
set_mqtt	设置 MQTT 连接参数	空对象
factory_reset	恢复出厂设置	rebooting
reboot	重启设备	rebooting
set_mqtt_topics	设置 MQTT 主题	空对象
set_wifi_timeout	设置 Wi-Fi 超时	空对象
start_ble_ota	开始 BLE OTA	ready
cancel_ota	取消 OTA	空对象
set_work_mode	设置工作模式	空对象
start_url_ota	URL OTA	ready
get_config_code	独立查询配置代号	返回 16 位大写十六进制代号
set_config_code	独立设置配置代号	空对象

3. 查询与配置接口

3.1 查询设备信息

读取设备身份、网络、串口、MQTT 和工作模式。

请求

```
{"action": "get_device_info", "data": {}}
```

成功响应

```
{ "success": true, "message": "device_info_retrieved", "data": { "mac": "6825DDA123F6",
"firmwareVersion": "7.0.0", "productName": "JM-WT24US", "wifi":
{"ssid": "Test", "password": "ok123456", "connected": true, "timeoutMinutes": 5, "clearOnTimeout": false},
"ble": { "name": "JM-WT24US-Ver6", "heartbeatMinutes": 1}, "uart":
{"baud": 115200, "dataBits": 8, "parity": "none", "stopBits": 1}, "mqtt":
{"server": "49.4.50.86", "port": 1883, "username": "admin", "password": "mqtt123", "tls": false, "subscribeTopic": "sqdj/sub/6825DDA123F6", "publishTopic": "sqdj/pub/6825DDA123F6", "heartbeatTopic": "sqdj/ping/6825DDA123F6", "heartbeatSeconds": 60}, "workMode": "wifi_only" } }
```

字段	类型/范围	必选	说明
data	object	是	字段必须存在；当前固件忽略对象内的额外字段，建议发送{}
data.mac	12 位字符串	响应	设备 MAC
data.firmwareVersion	string	响应	可读固件版本
data.productName	string	响应	产品型号
data.wifi	object	响应	Wi-Fi 状态与配置（详见下列子字段）
data.wifi.ssid	string	响应	当前保存的 Wi-Fi 名称

字段	类型/范围	必选	说明
data.wifi.password	string	响应	当前保存的 Wi-Fi 密码
data.wifi.connected	boolean	响应	当前 Wi-Fi 是否已连接并取得 IP
data.wifi.timeoutMinutes	1 ~ 65535 整数	响应	Wi-Fi 断网重连超时时间，单位分钟
data.wifi.clearOnTimeout	boolean	响应	Wi-Fi 超时后是否清除已保存的 Wi-Fi 配置
data.ble	object	响应	BLE 配置（详见下列子字段）
data.ble.name	string	响应	当前蓝牙广播名称
data.ble.heartbeatMinutes	1 ~ 65535 整数	响应	BLE 与监管 MQTT 心跳周期，单位分钟
data.uart	object	响应	串口配置（详见下列子字段）
data.uart.baud	支持波特率整数	响应	当前串口波特率
data.uart.dataBits	5/6/7/8	响应	当前串口数据位
data.uart.parity	none/even/odd	响应	当前串口校验方式
data.uart.stopBits	1/2	响应	当前串口停止位
data.mqtt	object	响应	MQTT 配置（详见下列子字段）
data.mqtt.server	string	响应	当前 MQTT 服务器域名或 IP 地址
data.mqtt.port	1 ~ 65535 整数	响应	当前 MQTT 服务端口
data.mqtt.username	string	响应	当前 MQTT 用户名，未配置时为空字符串
data.mqtt.password	string	响应	当前保存的 MQTT 密码，未配置时为空字符串
data.mqtt.tls	boolean	响应	当前是否启用 TLS

字段	类型/范围	必选	说明
data.mqtt.subscribeTopic	string	响应	当前设备接收主题
data.mqtt.publishTopic	string	响应	当前设备发布主题
data.mqtt.heartbeatTopic	string 或 null	响应	当前心跳主题; null 表示关闭心跳
data.mqtt.heartbeatSeconds	1 ~ 65535 整数	响应	当前 MQTT 心跳周期, 单位秒
data.workMode	枚举字符串	响应	当前工作模式

3.2 设置 Wi-Fi

配置设备连接的 Wi-Fi 名称和密码。

请求

```
{
  "action": "set_wifi",
  "data": {"ssid": "Test", "password": "ok123456"}
}
```

成功响应（仅表示配置已保存；新连接完成后另发 wifi_status_changed 事件）

```
{
  "success": true,
  "message": "wifi_configured",
  "data": {"connected": false, "ssid": "Test"}
}
```

失败响应示例

```
{
  "success": false,
```

```
"message": "invalid_wifi_password_length",
"data": {"field": "password", "reason": "invalid_length"}
}
```

字段	类型/范围	必选	说明
data.ssid	UTF-8, 1 ~ 32 字节	是	Wi-Fi 名称
data.password	UTF-8, 8 ~ 63 字节	是	Wi-Fi 密码
data.ssid	string	响应	已保存的目标 Wi-Fi 名称
data.connected	boolean	响应	响应时目标 SSID 和密码是否已经处于有效连接; 新配置通常为 false, 连接成功以随后事件为准

使用提示: 无论成功或失败均通过 wifi_status_changed 事件回传结果。

3.3 设置心跳

设置 BLE 与监管 MQTT 的设备心跳周期; 调试串口没有定时心跳任务, 不受本参数控制。

请求

```
{"action": "set_ble_heartbeat", "data": {"intervalMinutes": 3}}
```

成功响应

```
{"success": true, "message": "ble_heartbeat_configured", "data": {}}
```

字段	类型/范围	必选	说明
data.intervalMinutes	整数; <=0 按 1 处理; 最大 65535	是	BLE 与监管 MQTT 心跳间隔, 单位分钟

使用提示: 非数字、非整数或大于 65535 时返回 invalid_heartbeat_interval。

3.4 设置蓝牙名称

修改设备广播显示的蓝牙名称；当前固件仅接受 ASCII 字符。

请求

```
{"action": "set_ble_name", "data": {"name": "SLAVE"}}
```

成功响应

```
{"success": true, "message": "ble_name_configured", "data": {}}
```

字段	类型/范围	必选	说明
data.name	ASCII, 1 ~ 19 字符	是	蓝牙广播名称

使用提示：重启后生效。移动端若仍显示旧名称，应清除蓝牙缓存后重新扫描。

3.5 设置串口

设置 DTU 透传串口的通信参数。

请求

```
{
  "action": "set_uart",
  "data": {"baud": 9600, "dataBits": 8, "parity": "none", "stopBits": 1}
}
```

成功响应

```
{"success": true, "message": "uart_configured", "data": {}}
```

字段	类型/范围	必选	说明
data.baud	支持值见下方	是	波特率

字段	类型/范围	必选	说明
data.dataBits	5/6/7/8	是	数据位
data.parity	none/even/odd	是	校验方式
data.stopBits	1/2	是	停止位

使用提示：支持波特率：9600、19200、38400、57600、115200、230400、460800、921600。

3.6 设置 MQTT 连接

设置 MQTT 服务器和认证信息。

请求

```
{
  "action": "set_mqtt",
  "data": {
    "server": "49.4.50.86", "port": 1883,
    "username": "admin", "password": "8888888888",
    "tls": false
  }
}
```

成功响应

```
{"success": true, "message": "mqtt_configured", "data": {}}
```

字段	类型/范围	必选	说明
data.server	string	是	域名或 IP 地址
data.port	1 ~ 65535 整数	是	服务端口
data.username	string	否	用户名
data.password	string	否	密码

字段	类型/范围	必选	说明
data.tls	boolean	否	是否启用 TLS，默认 false

使用提示：username 或 password 缺失时写为空字符串；tls 缺失时写为 false。

3.7 设置 MQTT 主题

设置订阅、发布和心跳主题。

请求

```
{
  "action": "set_mqtt_topics",
  "data": {
    "subscribeTopic": "sqdj/sub/6825DDA123F6",
    "publishTopic": "sqdj/pub/6825DDA123F6",
    "heartbeatTopic": "sqdj/ping/6825DDA123F6",
    "heartbeatSeconds": 60
  }
}
```

成功响应

```
{"success": true, "message": "mqtt_topics_configured", "data": {}}
```

字段	类型/范围	必选	说明
data.subscribeTopic	string	否	设备接收主题
data.publishTopic	string	否	设备发布主题
data.heartbeatTopic	string 或 null	否	null 表示关闭心跳
data.heartbeatSeconds	1 ~ 65535 整数	条件必选	heartbeatTopic 为非 null 字符串时必须填；单独提供时仅更新周期

使用提示：主题不能为空、最长 512 字节且不能以 “/” 结尾。字段缺失表示保持原值；heartbeatTopic 为 null 表示关闭心跳。设备先在旧发布主题返回成功响应，再切换主题。

3.8 设置 Wi-Fi 超时

设置断网重连超时以及是否清除 Wi-Fi 配置。

请求

```
{"action": "set_wifi_timeout", "data": {"timeoutMinutes": 3, "clearWifi": false}}
```

成功响应

```
{"success": true, "message": "wifi_timeout_configured", "data": {}}
```

字段	类型/范围	必选	说明
data.timeoutMinutes	1 ~ 65535 整数	是	超时时间，单位分钟
data.clearWifi	boolean	是	超时后是否清除 Wi-Fi 配置

使用提示：默认超时 5 分钟。

3.9 设置工作模式

设置 Wi-Fi 与 BLE 透传模式。

请求

```
{"action": "set_work_mode", "data": {"mode": "wifi_ble"}}
```

成功响应

```
{"success": true, "message": "work_mode_configured", "data": {}}
```

字段	类型/范围	必选	说明
data.mode	wifi_ble/wifi_only/ble_onl	是	wifi_ble 双透传；wifi_only

字段	类型/范围	必选	说明
	y		仅 Wi-Fi; ble_only 仅 BLE

使用提示: 建议重启, 透明路由按新模式完整生效以重启后为准。

3.10 查询配置代号

仅用于 BLE、调试串口和监管 MQTT; 用户自定义 MQTT 不开放本动作。

请求

```
{"action":"get_config_code","data":{}}
```

成功响应

```
{"success":true,"message":"config_code_retrieved","data":{"configCode":"8F31A2C4D9017B6E"}}
```

字段	类型/范围	必选	说明
data	object	是	请求必须为空对象
data.configCode	16 位大写十六进制字符串	响应	设备当前配置代号

3.11 设置配置代号

独立设置设备配置代号。仅用于 BLE、调试串口和监管 MQTT; 用户自定义 MQTT 不开放本动作。

请求

```
{"action":"set_config_code","data":{"configCode":"8F31A2C4D9017B6E"}}
```

成功响应

```
{"success":true,"message":"config_code_configured","data":{"configCode":"8F31A2C4D9017B6E"}}
```



```
}}
```

字段	类型/范围	必选	说明
data.configCode	16 位大写十六进制字符串	是	仅允许字符 0-9、A-F

4. 设备控制接口

4.1 恢复出厂设置

清除用户配置并恢复设备出厂状态。监管 MQTT、BLE 和调试串口使用 factory_reset；用户 MQTT 使用 jmwifidtu_factory_reset。

请求

```
{"action": "factory_reset", "data": {"confirm": true}}
```

成功响应

```
{"success": true, "message": "factory_reset_accepted", "data": {"rebooting": true}}
```

字段	类型/范围	必选	说明
data.confirm	必须为 true	是	防止误操作
data.rebooting	boolean	响应	设备是否正在重启

使用提示：建议消费者应用显示二次确认。设备发送成功响应后延迟 100 ~ 300ms 再重启。

4.2 重启设备

重启设备。标准 reboot 可通过 clearWifi 选择是否清除 Wi-Fi；用户 MQTT 使用 jmwifidtu_reboot。

请求

```
{"action": "reboot", "data": {"clearWifi": false}}
```

成功响应

```
{"success": true, "message": "reboot_accepted", "data": {"rebooting": true}}
```

字段	类型/范围	必选	说明
data.clearWifi	boolean	否	是否清除 Wi-Fi, 默认 false
data.rebooting	boolean	响应	设备是否正在重启

5. OTA 升级接口

5.1 开始 BLE OTA

使设备进入 BLE 固件接收状态。

请求

```
{  
  "action": "start_ble_ota",  
  "data": {"firmwareSize": 890336, "md5": "3f8a7b2c119e4d06551a3c88f200e7dd"}  
}
```

成功响应

```
{"success": true, "message": "ble_ota_ready", "data": {"ready": true}}
```

失败响应示例

```
{"success": false, "message": "ota_busy", "data": {"reason": "busy"}}
```

字段	类型/范围	必选	说明
data.firmwareSize	正整数	是	固件总字节数
data.md5	32 位十六进制字符串	是	固件 MD5
data.ready	boolean	响应	是否可以开始传输

使用提示：固件二进制写入 FFE3，不放入 JSON。每个 Write 格式为[seq 低字节][seq 高字节][固件数据]，seq 从 0 开始按小端 uint16 逐包递增；当前建议 MTU 为 256、单包固件数据最多 251 字节。乱序包会被丢弃并等待发送端重试；连续 60 秒无数据自动取消；累计达到 firmwareSize 后自动校验 MD5 并提交。FFE3 当前仅支持 Write。

5.2 取消 OTA

取消尚未提交到 Flash 的升级任务。

请求

```
{"action": "cancel_ota", "data": {}}
```

成功响应

```
{"success": true, "message": "ota_cancelled", "data": {}}
```

字段	类型/范围	必选	说明
data	object	是	字段必须存在；当前固件忽略对象内的额外字段，建议发送 {}

5.3 URL OTA

从 HTTP 地址下载固件并升级。

请求

```
{
  "action": "start_url_ota",
```

```
"data": {  
  "url": "http://example.com/fw/JM-WT24-v5.bin",  
  "md5": "3f8a7b2c119e4d06551a3c88f200e7dd"  
}
```

成功响应

```
{"success": true, "message": "url_ota_started", "data": {"ready": true}}
```

字段	类型/范围	必选	说明
data.url	有效 HTTP URL	是	固件下载地址
data.md5	32 位十六进制字符串	建议	固件完整性校验
data.ready	boolean	响应	任务是否启动

5.4 OTA 主动事件

其中 BLE 升级有进度反馈，url 升级无进度反馈

进度事件

```
{  
  "event": "ota_progress",  
  "data": {"percent": 35, "receivedBytes": 311617, "totalBytes": 890336}  
}
```

完成事件

```
{  
  "event": "ota_completed",  
  "data": {"success": true, "message": "ota_completed_rebooting"}  
}
```

```
}

```

失败事件

```
{
  "event": "ota_completed",
  "data": {"success":false,"message":"md5_mismatch","reason":"md5_mismatch"}
}

```

event	data 字段	说明
ota_progress	percent、receivedBytes、totalBytes	升级进度，percent 范围 0~100
ota_completed	success、message、reason（失败时）	升级最终结果；成功后设备重启

6. 其他主动事件

设备心跳（监管 MQTT 发布到 jm/sup/ping/<MAC>；BLE/串口按设定周期；连接 WiFi 后附带 BSSID 和公网 IP）

```
{
  "event": "device_heartbeat",
  "data": {"mac":"6825DDA123F6","uptimeSeconds":3600,"rssi":-58,"bssid":"AA:BB:CC:DD:EE:FF","ip":"58.217.12.34"}
}

```

Wi-Fi 状态变化（set_wifi 下发后，成功/失败均回传 SSID 和密码；连接成功时附带 BSSID 和公网 IP）

```
{
  "event": "wifi_status_changed",
  "data":
  {"connected":true,"ssid":"Test","ip":"58.217.12.34","password":"ok123456","bssid":"AA:BB:CC:DD:EE:FF"}
}

```

```
EE:FF"}  
}
```

连接失败或断开时, ip 和 bssid 为空:

```
{  
  "event": "wifi_status_changed",  
  "data": {"connected":false,"ssid":"Test","ip":"","password":"ok123456","bssid":""}  
}
```

event	触发时机	关键字段
device_heartbeat	监管 MQTT 连接后立即一次, 之后达到设定周期; BLE 达到设定周期	mac、uptimeSeconds、 rssi、 bssid、 ip
wifi_status_changed	目标 Wi-Fi 取得 IP 后向 set_wifi 原通道回传; 监管 MQTT 同时接收连接、断开或切换事件; set_wifi 下发后成功/失败均包含 ssid 和 password	connected、 ssid、 ip、 password、 bssid
mqtt_status_changed	MQTT 连接或断开	connected、 server
ble_status_changed	BLE 连接或断开	connected

7. 失败响应规范

协议不提供数字错误码。失败原因通过可读 message 说明; 程序如需稳定判断, 可读取 data.reason 字符串。reason 不是旧协议结果码, 也不与 success 重复: success 用于判断成功或失败, reason 仅在失败时说明失败类别。

reason	适用场景	message 示例
invalid_json	JSON 语法错误	invalid_json
missing_field	缺少必选字段	missing_required_field
invalid_parameter	类型错误或值不支持	invalid_parameter

reason	适用场景	message 示例
invalid_length	字符串长度不合法	invalid_length
unsupported_action	action 不存在	unsupported_action
busy	设备正在执行互斥任务	ota_busy
md5_mismatch	固件校验失败	md5_mismatch
internal_error	无法进一步分类	internal_error

标准失败响应

```
{  
  "success": false,  
  "message": "missing_required_field",  
  "data": {"field": "baud", "reason": "missing_field"}  
}
```

附录 A : JSON Schema 摘要

请求:

```
{  
  "action": "<操作名称>",  
  "data": { ... }  
}
```

响应:

```
{  
  "success": true | false,  
  "message": "<english_snake_case_status>",  
  "data": { ... }  
}
```

主动事件:

```
{  
  "event": "<事件名称>",  
  "data": { ... }  
}
```

最终约束: 协议只采用本附录规定的三种对象结构, 不混入旧协议字段或二进制帧字段。若未来必须支持并发请求, 应单独发布新的协议大版本。

附录 B Web Bluetooth 特征与透传方向

以下 TX / RX 命名均以 DTU 设备视角为准。JSON 配置指令与透明数据使用相互独立的特征，网页端不得将 FFE4 与 FFE5 的方向反向理解。

UUID (16 位)	名称 / 用途	属性	设备视角方向	网页端行为
FFE0	主服务	Primary Service	—	发现服务后再获取下列特征
FFE1	RX (JSON 配置接收)	Write	网页 → 设备	网页可单次或分片写入 JSON V2.0 请求；设备按完整对象重组
FFE2	TX (JSON 配置发送)	Read / Notify	设备 → 网页	网页订阅并接收 JSON V2.0 响应/事件
FFE3	OTA	Write	网页 → 设备	蓝牙本地升级数据通道
FFE4	TX_DATA (透明发送)	Notify	设备 → 网页	网页订阅通知，接收透明串口数据
FFE5	RX_DATA (透明接收)	Write	网页 → 设备	网页写入透明数据，由设备接收

对接要求：配置助手使用 FFE1 / FFE2；BLE OTA 数据使用 FFE3；透明数据使用 FFE4 / FFE5。当前 C2 固件已创建并启用以上全部特征。FFE1、FFE3 和 FFE5 使用 Write；FFE2 使用 Read / Notify；FFE4 使用 Notify。