使用泉盛UV-K5/K6/K1定制固件

DTrac APP 新增了对泉盛UV-K5/K6/K1的支持，现在可以通过APP操控UV-K5/K6/K1实时同步卫星转发器参数，方便业余卫星通联操作。

实现功能

仅需使用APP连接对讲机，抛弃繁琐一个手机全搞定，业余卫星通联操作更智能，实现以下功能：

- 自动获取位置；
- 自动更新星历；
- 成熟算法预测；
- 实时同步卫星多普勒频率；
- 实时同步通联模式参数；
- 实时同步发射亚音参数；
- 卫星入境时自动开启监听，离境后自动关闭监听；

注：由于泉盛UV-K5/K6/K1硬件限制，除FM外的模式仅对接收信道有效。

硬件改造

泉盛UV-K5/K6/K1系列电台虽然自带串口，可以基于外接网络转串口的方式来实现对接，但总是不太方便，最好的方法是在电台内部加上经典蓝牙透传串口组件，一劳永逸。

推荐使用3.3V的双模蓝牙透传模块，方便日后拓展，如EBYTE/亿佰特EWM104-BT41SP或度云的doBT-M01双模蓝牙模块，我个人更推荐使用EBYTE/亿佰特EWM104-BT41SP模块，它可以通

过BLE在线修改模块设置，详见数据手册。操作步骤：

- 设置为从机模式；
- 自定义蓝牙模块名称如：UV-K6；
- 修改波特率为38400；
- 拆机（自行搜索网上教程）；
- 然后根据蓝牙模块的引脚定义参考下图位置使用合适的漆包线连接
  - 蓝牙 → 电台
  - TXD 接 RXD
  - RXD 接 TXD
  - 3.3V 接 3V3
  - GND 接 GND
- 将模块绝缘处理，放在合适位置即可。

其中，ble部分UUID定义如下：

- Service UUID: 0000FFF0-0000-1000-8000-00805F9B34FB
- Notify feature：0000FFF1-0000-1000-8000-00805F9B34FB
- Write feature：0000FFF2-0000-1000-8000-00805F9B34FB

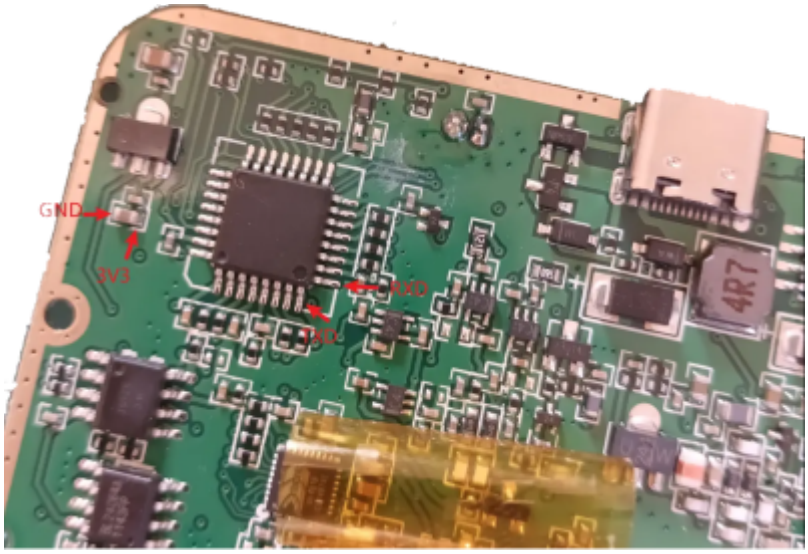
前期使用度云双模蓝牙模块（doBT-M01（从机版本））的玩家，需要依次执行以下设置命令激活SPP和BLE：

- AT+CR00\r\n
- AT+CT03\r\n
- AT+BDUV-K6\r\n
- AT+B501\r\n
- AT+BMUV-K6_BLE\r\n
- AT+B401\r\n
- AT+U0FFF0\r\n
- AT+U1FFF3\r\n
- AT+U2FFF1\r\n
- AT+U3FFF2\r\n

亿佰特EWM104-BT41SP双模蓝牙模块SPP和BLE默认是激活的，只需要依次执行以下设置命令：

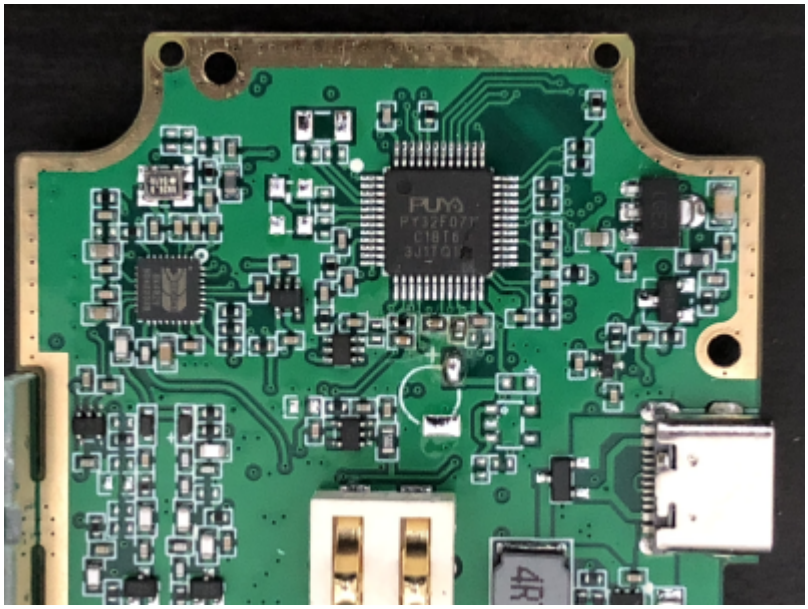
- AT+BAUD=4
- AT+SPPNAME=UV-K6
- AT+BLENAME=UV-K6_BLE

UV-K5/K6(DP32G030)

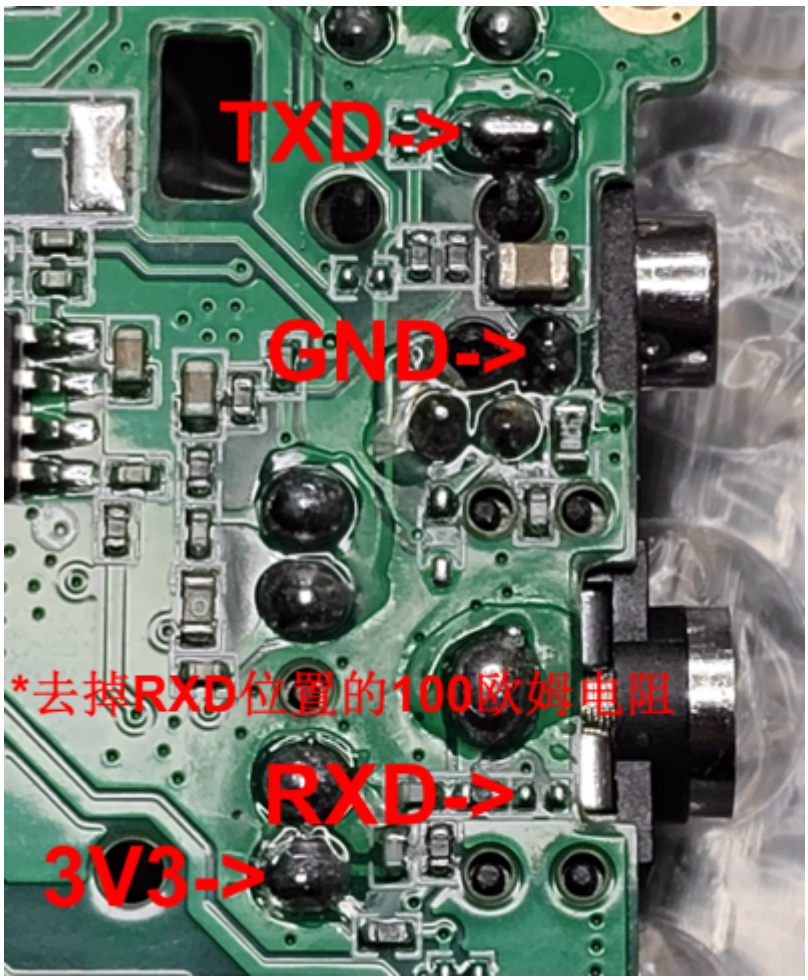


UV-K5/6的TXD在MCU的8脚，RXD在MCU的9脚。

UV-K1/K5v3(PY32F071)



注：UV-K1接线方法，请参照相关电路图，关键是找到RXD、TXD、3V3、GND的位置，一般根据K头走线很容易查找，以下内容仅供参考。



UV-K1需要去除原RXD位置与3.5mm插座PTT功能极的100欧姆电阻，防止开机误触进入刷机模式，TXD在MCU的30脚，RXD在MCU的31脚。

由于UV-K1较紧凑，蓝牙芯片建议选择小尺寸的，如EBYTE/亿佰特EWM104-BT41SP双模蓝牙模块，绝缘处理后恰好可以放在对讲机的顶部空隙位置。

定制固件

定制的固件在F4HWN、BA7IQE等最新开源固件上轻微修改，完整保留了原固件功能。相对于泉盛官方原版固件，由于内存映射完全改变，你会丢失所有内存通道和设置。注： 固件使用Apache-2.0协议

固件主要迭代功能，请参阅原固件更新日志。

UV-K5/K6(DP32G030)

固件修改自

https://github.com/armel/uv-k5-firmware-custom

参考文献

https://deepwiki.com/armel/uv-k5-firmware-custom

UV-K1/K5v3(PY32F071)

固件修改自

https://github.com/armel/uv-k1-k5v3-firmware-custom

参考文献

https://deepwiki.com/armel/uv-k1-k5v3-firmware-custom

下载地址

警告：固件仅供学习和业余无线电交流，由于泉盛UV-K系列对讲机版本迭代很多，下载使用时请认准对应的版本，如果造成对讲机变砖，概不负责！

UV-K5/K6(DP32G030)

注：仅适用于MCU为DP32G030的UV-K5和UV-K6使用。

固件下载地址：

UV-K5/K6(DP32G030) for DTrac v1.0.1(F4HWN v4.3)

固件修改自

https://github.com/armel/uv-k5-firmware-custom

参考文献

https://deepwiki.com/armel/uv-k5-firmware-custom

UV-K1/K5v3(PY32F071)

注：仅适用于MCU为PY32F071的UV-K1和UV-K5/K6使用。

固件下载地址：

UV-K1/K5V3(PY32F071) for DTrac v1.0.5(F4HWN v5.3.1)

固件修改自

https://github.com/armel/uv-k1-k5v3-firmware-custom

参考文献

https://deepwiki.com/armel/uv-k1-k5v3-firmware-custom

刷机方法

刷机方法简介：在线刷机、离线线刷和蓝牙无线刷机的方法都是一样的，长按PTT键开机，选择正确的端口，连接电台，选择固件，点击更新即可。

原版固件升级详细操作步骤可参考官方升级固件程序和说明文档，升级变砖时可以先刷回原版试试。

UV-K5/K6官方升级固件程序和说明

其它版本硬件，强烈推荐在线刷机方式。

备份校准文件

养成良好的习惯，刷机前先备份校准数据，更新固件后立即创建一个新的校准文件，然后在安装不同固件之前（比如回到原厂固件时）先恢复它，刷机后再恢复校准数据。

开机用普通模式通过刷机网站在线备份，参阅网站友好提示。

用你对讲机的序列号来重命名校准文件，序列号印在设备背面（电池下方）标签上，这样可以避免你有多台设备时校准文件混淆。

GL!

UV-K5/K6(DP32G030)

在线刷机

https://www.dtrac.cn/uvtools/

https://armel.github.io/uvtools/

UV-K1/K5v3(PY32F071)

在线刷机

https://www.dtrac.cn/uvtools2/

https://armel.github.io/uvtools2/

恢复校准文件

开机用普通模式通过刷机网站在线恢复校准，参阅网站友好提示。

对讲机设置

将RxMode设置为MAIN ONLY，并使用VFO频率待机模式。

APP设置

本例，电台类型选择“QUANSHENG UV-K6”，选择已配对的电台蓝牙适配器，如上述自定义的“UV-K6”，然后用蓝牙方式连接对讲机。

当然，你也可以通过使用电脑的虚拟串口软件将电台自带的串口转换为网络透传服务，然后用网络接口的方式来与DTrac APP对接，可参考以下内容 使用虚拟串口 。

演示视频

UV-K6

dtrac_quansheng_uvk6.mp4

UV-K1

https://v.douyin.com/rpza7seZ6w8/

隐藏校准菜单

进入固件隐藏校准菜单的方法如下：开机时按住PTT+侧键 1，然后松开所有键，进入隐藏菜单，固件刷新后建议执行菜单75/75的RESET ALL，复位一次。

进入后，找到与电池校准相关的选项，如“BatCal”，即可进行电压校准等操作。

相关协议

Protocol for UV-K5/K6/K1

The programming protocol used by this software has been reverse engineered by observing communications between the radio and the original programming software. It is not a variation of the typical Baofeng-like protocol.

The format of the datagram sent to the radio is:

0xAB 0xCD len 0x00 <data bytes> <2 bytes CRC> 0xDC 0xBA

The length is the length of the data bytes.

The data is protected by a typical CRC-16 xmodem algorithm. The data bytes and the CRC are obfuscated by xor-in it with an 8-byte sequence.

Fortunately the EEPROM data contains a lot of 0xFF and 0x00 bytes, so the XOR sequence is easy to find by observing the traffic.

The datagram sent from the radio is the same, but the CRC field is set to 0xFFFF. This shows that the CRC is not for data integrity, but for further obfuscation (same as the XOR).

I intend to publish a further description of the protocol, and the EEPROM contents, meanwhile the sources can be used as documentation.

计算机与无线电台之间的指令

- ID 功能描述
- 0x0514 会话初始化，关于版本信息和状态的回复
- 0x051B EEPROM读出
- 0x051D 致EEPROM
- 0x0527 RSSI值读数
- 0x0529 ADC读数，电池信息回复

- 0x052D 检查锁状态
- 0x052F 会话初始化、回复、详细版本信息和状态
- 0x0601 BK4819 寄存器读出
- 0x0602 写入BK4819寄存器
- 0x0801 按键模拟
- 0x0803 将LCD屏幕内存转储到PC
- 0x0808 扫描
- 0x0850 写入多个寄存器
- 0x0851 读取多个寄存器
- 0x0860 配置GPIO引脚
- 0x0861 GPIO 引脚状态读数
- 0x0870 启用全控制模式
- 0x0888 扫描
- 0x9999 for DTrac app CTCSS_CODE
- 0x8888 for DTrac app DownFrequency
- 0x7777 for DTrac app UpFrequency
- 0x6666 for DTrac app Mode
- 0x5555 for DTrac app MonitorStatus

计算机与无线电之间通信的实际数据

以下是您在全盛码头选择[F-0]调频收音机时的序列数据。

AB CD 06 00 17 64 16 E6 2E 91 E8 EA DC BA

1. 分组发起 数据包以0xAB 0xCD开头。
2. 数据长度 接下来的两个字节表示实际的有效载荷数据长度。
3. 有效载荷 接下来的7字节是经过异或处理的载荷数据中的主要信息部分，包括要发送和接收的命令。
4. CRC校验和 是一种CRC校验和，用于确保以下1字节有效载荷数据的完整性。
5. 数据包终止 最后一个0xDC 0xBA表示数据包的结束。

发送和接收的数据在以下字节列中进行了异或加密：

通过异或处理的数据可以通过异或解锁，每个字节对应的字节{ 0x16, 0x6c, 0x14, 0xe6, 0x2e, 0x91, 0x0d, 0x40, 0x21, 0x35, 0xd5, 0x40, 0x13, 0x03, 0xe9, 0x80 }

AB CD 06 00 17 64 16 E6 2E 91 E8 EA DC BA

- 0x17 XOR 0x16 → 0x01
- 0x64 XOR 0x6C → 0x08
- 0x16 XOR 0x14 → 0x02
- 0xE6 XOR 0xE6 → 0x00
- 0x2E XOR 0x2E → 0x00
- 0x91 XOR 0x91 → 0x00
- 0xE8 XOR 0x0D → 0xE5

实际数据 → AB CD 06 00 01 08 02 00 00 00 E5 EA DC BA

Data for DTrac APP

for DTrac app CTCSS_CODE

typedef struct {

```
Header_t Header;
uint8_t CTCSS_CODE;
```

} CMD_9999_t;

for DTrac app DownFrequency

typedef struct {

```
Header_t Header;
uint32_t DownFrequency;
```

} CMD_8888_t;

for DTrac app UpFrequency

typedef struct {

```
Header_t Header;
uint32_t UpFrequency;
```

} CMD_7777_t;

for DTrac app Mode

typedef struct {

```
Header_t Header;
char Mode;
```

} CMD_6666_t;

for DTrac app MonitorStatus

typedef struct {

```
Header_t Header;
char MonitorStatus;
```

} CMD_5555_t;

其他

http://www.dtrac.cn/k5viewer/

https://armel.github.io/k5viewer/

https://spm81.github.io/Multi-UVTools/#home