**SHRD100\_RemoteId模块-详细设计**

|  |  |
| --- | --- |
| 拟 制 |  |
| 审 核 |  |
| 会签 |  |
| 批 准 |  |

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **修订版本** | **日期** | **作者** | **修改描述** | **备注** |
| V1.0 | 2023.08.03 | 滑国青 | 初始版本 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 引言 2](#_Toc141725567)

[1.1 目的 2](#_Toc141725568)

[1.2 范围 2](#_Toc141725569)

[1.3 缩略语定义 2](#_Toc141725570)

[1.4 参考资料 2](#_Toc141725571)

[2 需求概述 2](#_Toc141725572)

[2.1 功能需求 2](#_Toc141725573)

[2.1.1 需求1 2](#_Toc141725574)

[2.1.2 需求2 3](#_Toc141725575)

[2.2 接口需求 3](#_Toc141725576)

[2.3 性能需求 3](#_Toc141725577)

[2.4 边界需求 3](#_Toc141725578)

[3 软件设计说明 3](#_Toc141725579)

[3.1 系统示意图 3](#_Toc141725580)

[4 Issues解答 4](#_Toc141725581)

[4.1 问题1 4](#_Toc141725582)

[4.2 问题1 4](#_Toc141725583)

[5 附件 4](#_Toc141725584)

# 引言

## 目的

本文为“SHRD100\_RemoteId模块-详细设计”，主要用于定义软件功能，供项目组开发人员和软件维护人员阅读。

## 范围

本文档只限于塞防科技项目组研发、测试、产品以及项目相关人员作为内部信息对齐使用，未经公司批准以及书面授权不允许任何人以任何形式对本文档复制、传播、改动。

## 缩略语定义

|  |  |  |
| --- | --- | --- |
| **缩略语** | **全称** | **描述** |
| OTA | On The Air | 指固件升级功能 |

## 参考资料

|  |  |
| --- | --- |
| **名称** | **版本** |
| SHRD100\_系统设计说明书 |  |
| SHRD100嵌入式软件设计说明V1.2.docx |  |
|  |  |
|  |  |
|  |  |

# 需求概述

## 功能需求

### 需求1

### 需求2

## 接口需求

无

## 性能需求

无

|  |  |
| --- | --- |
| 序号 | 性能需求 |
| 1 |  |
| 2 |  |

## 边界需求

无

|  |  |
| --- | --- |
| 序号 | 边界需求 |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

# 软件设计说明

## 系统示意图

## 如何获取RemoteId使用wifi接口名

### 特征

当前，tracer产品上有两块wifi网卡，怎么知道哪块是rn440网卡，是我们需要的wifi接口呢？

通过特殊值，查找网卡信息

目前rn440网卡用于remoteId抓包，它是pci网卡，当驱动加载后，

存在/proc/net/rtl8852be/wlp1s0/pci\_aspm文件;

# find /proc/net/ | grep pci\_aspm

/proc/net/rtl8852be/wlp1s0/pci\_aspm

而rg450网卡是rtl8821cs,它是sdio网卡。

# find /proc/net/ | grep sdio\_card\_info

/proc/net/rtl8821cs/wlan0/sdio\_card\_info

通过上面特殊值，我们可以提取第4节，得到正确的网卡信息，这可以在系统启动后，使用脚本去作，然后写到文件中：

/tmp/ ifname\_remoteid

/tmp/ifname\_wifi

### appinit.d/S02check\_wifi.sh

IFNAME=`get\_wifi\_name.sh remoteid`

if [ $? -eq 0 ] ;then

echo -n ${IFNAME} > /tmp/ifname\_remoteid

fi

IFNAME=`get\_wifi\_name.sh wifi`

if [ $? -eq 0 ] ;then

echo -n ${IFNAME} > /tmp/ifname\_wifi

fi

### get\_wifi\_name.sh

usage()

{

echo "$0 {remoteid|wifi}"

}

type=$1

## find /proc/net/ | grep pci\_aspm

# /proc/net/rtl8852be/wlp1s0/pci\_aspm

## find /proc/net/ | grep sdio\_card\_info

# /proc/net/rtl8821cs/wlan0/sdio\_card\_info

if [ x${type} = x"remoteid" ];then

FILE=pci\_aspm

elif [ x${type} = x"wifi" ];then

FILE=sdio\_card\_info

else

usage

exit 2

fi

STR=`find /proc/net/ | grep ${FILE}`

if [ $? -eq 0 ] ;then

echo -n ${STR} | cut -d '/' -f5

exit 0

else

exit 1

fi

# RemoteID模拟发射程序

[GitHub - opendroneid/transmitter-linux: Example Open Drone ID Linux transmitter for Bluetooth and Wi-Fi.](https://github.com/opendroneid/transmitter-linux)

This program supports transmitting static drone ID data via Wi-Fi Beacon or Bluetooth on a desktop Linux PC or Raspberry Pi HW.

For further information about Open Drone ID and the related specifications, please see the documentation in [opendroneid-core-c](https://github.com/opendroneid/opendroneid-core-c).

The drone ID data is just static data, since the main purpose is to demonstrate how to setup the Bluetooth and Wi-Fi Beacon HW + SW to transmit valid drone ID data. This could easily be extended to simulate a bit more dynamic flight example.

## How to compile

The below instructions have been tested on Ubuntu 20.04 and on Raspbian 10 (Buster) / Rasbian 11 (Bullseye).

git clone git@github.com:opendroneid/transmitter-linux.git

cd transmitter-linux

git submodule update --init

The Bluez source code repository has been set to version 5.53, since this matched the version installed on Ubuntu 20.04. It might be required to check what version of Bluez is installed on your environment and adjust accordingly: Check any of btmon --version, btmgmt --version, bluetoothctl --version, hcitool or hciconfig.

Compile hostapd:

sudo apt install build-essential git libpcap-dev libsqlite3-dev binutils-dev bc pkg-config libssl-dev libiberty-dev libdbus-1-dev libnl-3-dev libnl-genl-3-dev libnl-route-3-dev

cd hostapd/hostapd

cp defconfig .config

make -j4

cd -

Compile gpsd (see [gpsd/build.adoc](https://gitlab.com/gpsd/gpsd/-/blob/master/build.adoc) for all build requirements):

sudo apt install scons

cd gpsd

sed -i 's/\(variantdir \*=\).\*$/\1 "gpsd-dev"/' SConstruct

scons minimal=yes shared=True gpsd=False gpsdclients=False socket\_export=yes

cd -

Compile the transmitter example application:

mkdir build && cd build

cmake ../.

make -j4

## Command line parameters

* b Enable Wi-Fi Beacon transmission
* l Enable Bluetooth 4 Legacy Advertising transmission using non-Extended Advertising API
* 4 Enable Bluetooth 4 Legacy Advertising transmission using Extended Advertising API
* 5 Enable Bluetooth 5 Long Range + Extended Advertising transmission
* p Use message packs instead of single messages
* g Use gpsd to dynamically update location messages after each loop of messages

## Starting Wi-Fi Beacon transmission

The Wi-Fi Beacon transmission only works properly when the PC is not connected to any Wi-Fi hotspots. If it is connected, it somehow prevents configuring the Wi-Fi HW to act as a local access point.

On Ubuntu, open the system settings -> Wi-Fi and for each of the networks that are stored on the machine, click the gear icon and then click forget network.

On the Raspberry Pi OS, try first to click the Wi-Fi icon in the top right corner. For each network that is already configured, right-click it and forget the network. (On Raspbian 11 Bullseye, left-click the network and it will ask if you want to forget it). It is also possible that there are connection settings stored in the file /etc/wpa\_supplicant/wpa\_supplicant.conf. Edit the file and delete/comment out all known Wi-Fi networks, then reboot.

Check that there is nothing preventing the usage of the Wi-Fi HW by running the tool rfkill. Any SW block should be possible to unblock via the same tool.

To start hostapd, a configuration file must be present with the following content. E.g. modify the beacon.conf that is provided in the root of the project folder. Make sure the second line matches the name of your WLAN device. Can be checked e.g. with ip link show or iw dev:

country\_code=DE

interface=wlan0

ssid=DroneIDTest

hw\_mode=g

channel=6

macaddr\_acl=0

auth\_algs=1

ignore\_broadcast\_ssid=0

wpa=2

wpa\_passphrase=thisisaverylongpassword

wpa\_key\_mgmt=WPA-PSK

wpa\_pairwise=TKIP

rsn\_pairwise=CCMP

ctrl\_interface=/var/run/hostapd

beacon\_int=200 # Does this work? Doesn't seem to have any effect?

#This is an empty information element. dd indicates hex. 01 is the length of the data and 00 the actual data:

vendor\_elements=dd0100

#This information element has one fixed Location message:

#vendor\_elements=dd1EFA0BBC0D00102038000058D6DF1D9055A308820DC10ACF072803D20F0100

Open a separate shell at the top of the project and start the hostapd daemon. This only works with sudo rights:

sudo hostapd/hostapd/hostapd beacon.conf

In the original shell, start the opendroneid transmitter. Again, this only works with sudo rights:

sudo ./transmit b p

This has been tested on a [CometLake Z490 desktop](https://rog.asus.com/motherboards/rog-strix/rog-strix-z490-i-gaming-model) with built-in Wi-Fi HW on the motherboard. For some reason, a fair amount of the messages being sent to hostapd are not received or at least not properly acknowledged by the lower SW layers. This is clearly visible when following the command line output. It is not clear why this is happening.

Another unclear issue is that setting the beacon interval in the beacon.conf file does not seem to have any effect.

When tested on Raspberry Pi 3B and 4B, the beacon transmit only partly worked. The pre-defined single location message that can be uncommented in the beacon.conf file works okay. The data that the transmit program tries to write to the hostap daemon works okay for single messages (sudo ./transmit b) but not for message packs (sudo ./transmit b p). There must be some size limitation in the Wi-Fi driver for vendor specific element data. Tested on Raspberry Pi 3B with Raspbian 11 Bullseye. Please note that the drone ID standards mandate message packs to be used for Wi-Fi Beacon transmissions.

## Starting Bluetooth transmission

The program must be run with sudo rights:

sudo ./transmit 5 p

For educational purposes, it is possible to follow the HCI command flow using the tool sudo btmon. This requires Bluez to be installed: sudo apt install bluez.

This has been tested on a [CometLake Z490 desktop](https://rog.asus.com/motherboards/rog-strix/rog-strix-z490-i-gaming-model) with built-in Bluetooth HW on the motherboard. On this HW, BT4 Legacy Advertisements using the Extended Advertising API (option 4) and BT5 Long Range + Extended Advertisements (option 5 p) would work just fine when run individually. However, when configuring both simultaneously, for some reason the HW/driver SW would only broadcast the BT4 signals.

Please note that when using Bluetooth transmission for drone ID in the USA, it will be mandatory to transmit both BT4 and BT5 [simultaneously](https://github.com/opendroneid/opendroneid-core-c#relevant-specifications). And although it is not mandated for Europe or Japan, it would be a good idea to do the same there, in order to maximize the compatibility with receivers and the range of the signals.

Using the BT4 Legacy Advertising non-Extended Advertising HCI commands (option l) for some reason didn't work.

When tested on a Raspberry Pi 3B and 4B, the Extended Advertising HCI interface commands are not supported (option 4 and option 5 p). Probably due to the HW/driver only supporting BT4 Legacy Advertising. The older BT4 Legacy non-Extended Advertising HCI commands (option l) worked okay.

A BT5 USB adapter/dongle of the brand ONVIAN and with the chipset RTL8761B has been tested on a PC with Ubuntu 20.04 and proven to be able to successfully transmit in Long Range mode.

## How to clean up

If the program is terminated abnormally, Beacon and Bluetooth broadcasts can remain running.

* To stop Beacon broadcast, stop the hostapd instance. It can be difficult to stop the transmit instance. After stopping hostapd, use sudo pkill transmit.
* To stop Bluetooth, use sudo btmgmt power off and then sudo btmgmt power on.

hostapd -i wlan0 -B beacon.conf

# 开源重要源码

## opendroneid

[Open Drone ID · GitHub](https://github.com/opendroneid/)

伪beacon发射模拟

[GitHub - opendroneid/transmitter-linux: Example Open Drone ID Linux transmitter for Bluetooth and Wi-Fi.](https://github.com/opendroneid/transmitter-linux)

Remoteid编解码库

[GitHub - opendroneid/opendroneid-core-c: Open Drone ID Core C Library](https://github.com/opendroneid/opendroneid-core-c)

[GitHub - opendroneid/specs: The Open Drone ID specification](https://github.com/opendroneid/specs)

[GitHub - opendroneid/wireshark-dissector: Dissector of Open Drone ID broadcast protocol for Wireshark](https://github.com/opendroneid/wireshark-dissector)

## unix\_rid\_capture

开源remoteid的检测程序

[GitHub - sxjack/unix\_rid\_capture: A program that captures ASTM F3411 / ASD-STAN 4709-002 UAV direct remote identification signals transmitted over WiFi and Bluetooth.](https://github.com/sxjack/unix_rid_capture)

## Radiotap

wifi 抓包时的radiotap头的解析库

<https://github.com/boundary/wireshark/blob/master/epan/dissectors/packet-ieee80211-radiotap.c>

## wireless-tools交叉编译

wifi配置工具集，阅读源码，可以学习如何通过c程序进行wifi设置

[Wireless Tools-29 (linuxfromscratch.org)](https://www.linuxfromscratch.org/blfs/view/svn/basicnet/wireless_tools.html)

[**https://hewlettpackard.github.io/wireless-tools/wireless\_tools.29.tar.gz**](https://hewlettpackard.github.io/wireless-tools/wireless_tools.29.tar.gz)

[**https://www.linuxfromscratch.org/patches/blfs/svn/wireless\_tools-29-fix\_iwlist\_scanning-1.patch**](https://www.linuxfromscratch.org/patches/blfs/svn/wireless_tools-29-fix_iwlist_scanning-1.patch)

### Installation of Wireless Tools

First, apply a patch that fixes a problem when numerous networks are available:

**patch -Np1 -i ../wireless\_tools-29-fix\_iwlist\_scanning-1.patch**

To install Wireless Tools, use the following commands:

**make**

This package does not come with a test suite.

Now, as the *root* user:

**make PREFIX=/usr INSTALL\_MAN=/usr/share/man install**

# 如何增加hal接口

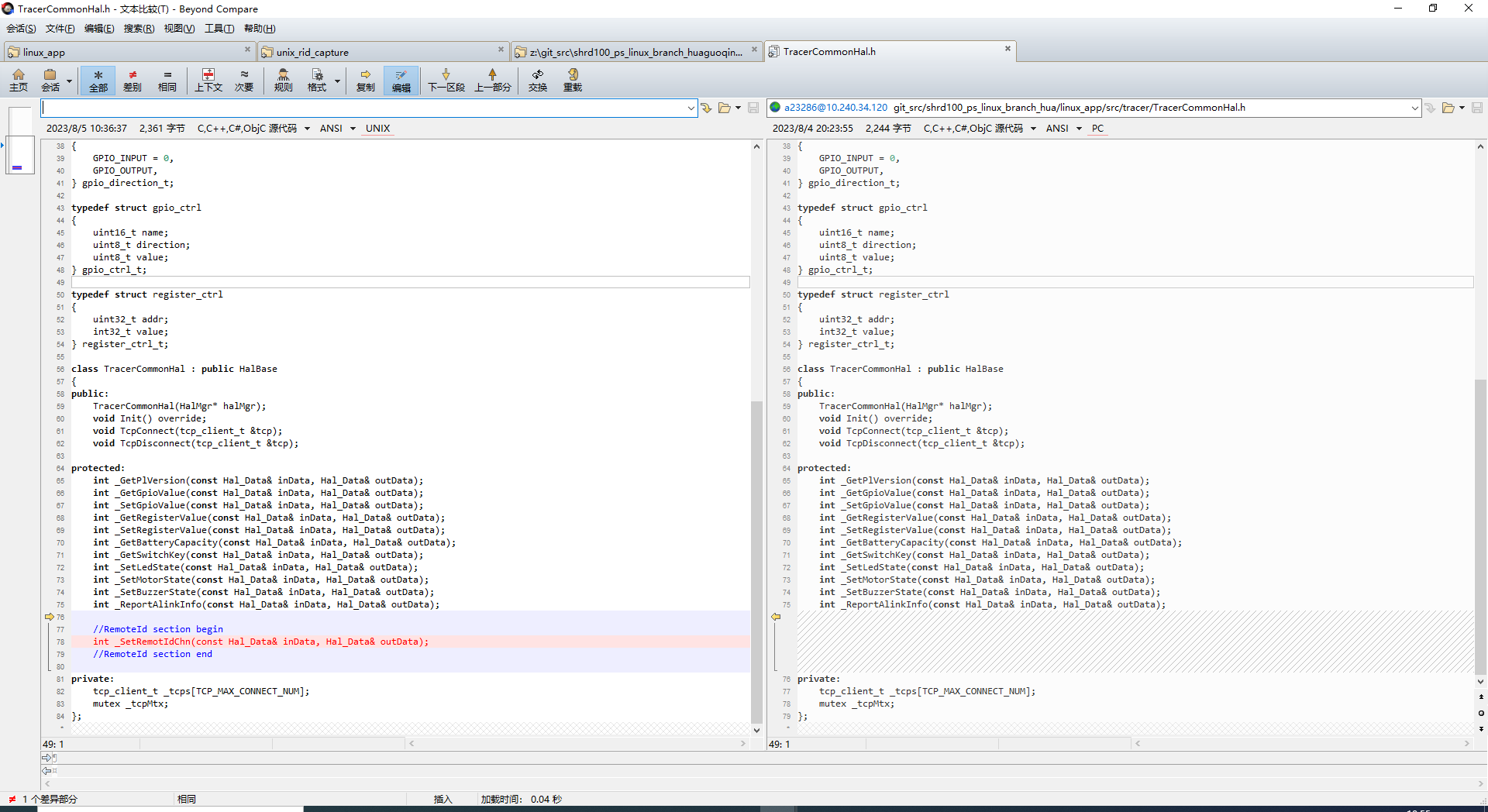
\linux\_app\src\eap\include\pub\eap\_sys\_drvcodes.h

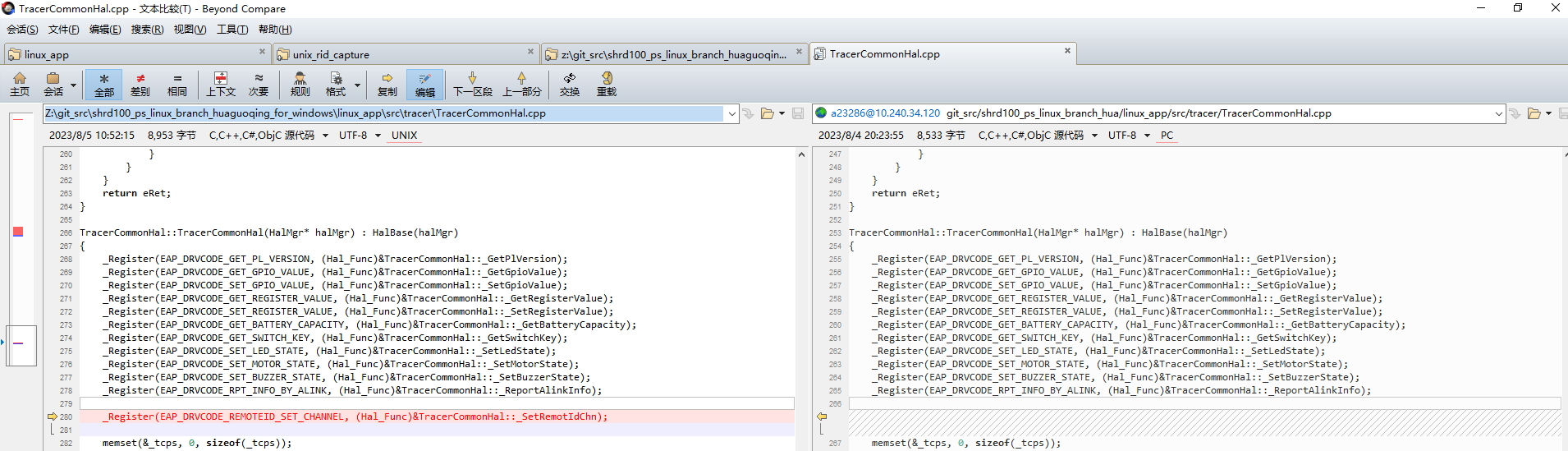
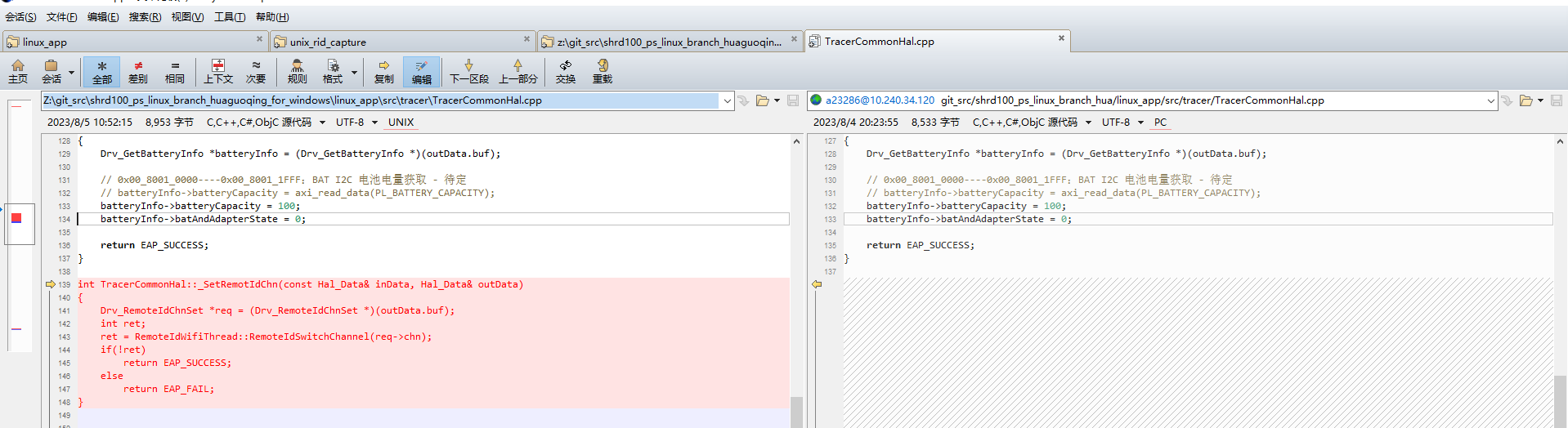
增加命令类型宏，及输入参数的数据结构

\linux\_app\src\tracer\TracerCommonHal.h

\linux\_app\src\tracer\TracerCommonHal.cpp

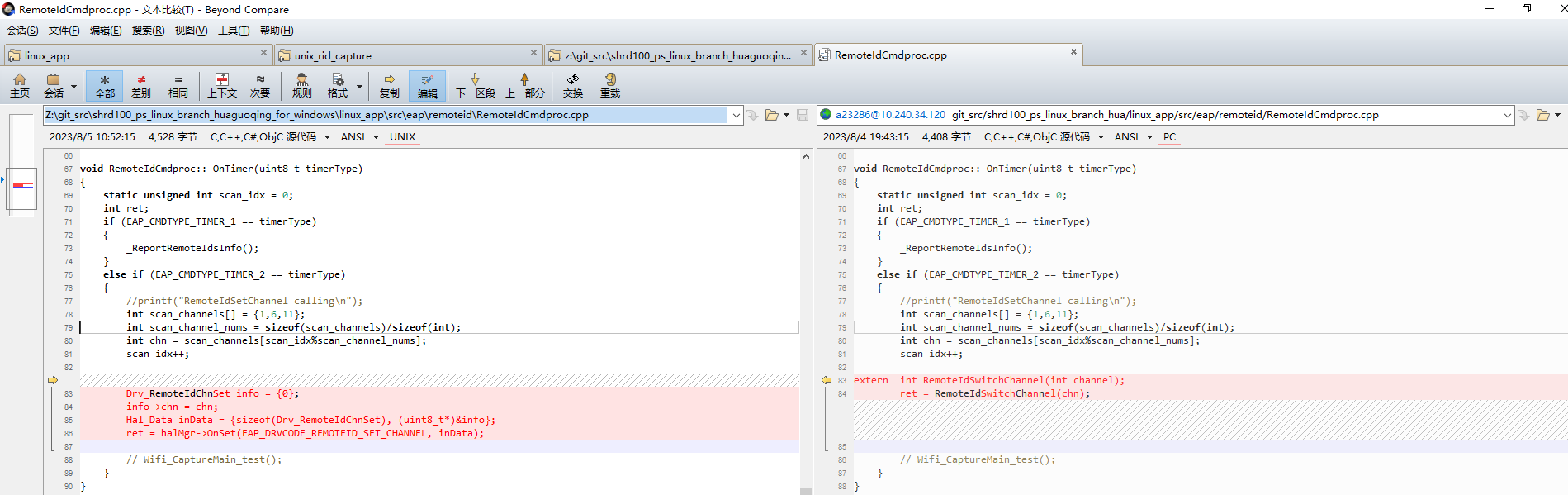
增加命令类型的执行函数,并绑定执行函数与命令类型（注册命令）





\linux\_app\src\eap\remoteid\RemoteIdCmdproc.cpp

收到消息后，调用hal的对应命令的方法



# Issues解答

## 问题1

## 问题1

# 附件

无。