

# 千寻差分数据 Linux SDK

## 开发指南

V1.3.0

千寻位置网络有限公司

2017 年 5 月·上海

# 法律声明

版权所有© 2016，千寻位置网络有限公司。保留一切法律权利。本文档包含的所有内容除特别声明之外，版权均属于千寻位置网络有限公司所有，受《中华人民共和国著作权法》及相关法律法规和中国加入的所有知识产权方面的国际条约的保护。未经本公司书面许可，任何单位和个人不得以任何方式(电子或机械,包括影印)或理由对该文档或其包含的任何产品、服务、信息、材料的任何部分进行使用、复制、修改、抄录、传播或与其它产品捆绑使用、销售，否则将视为侵权，本公司必依法追究其法律责任。本文档并不代表供应商或其代理的承诺，千寻位置网络有限公司可在不作任何申明的情况下对本文档内容进行修改。本文档中提到的其它公司及其产品的商标所有权属于该商标的所有者。

**千寻位置网络有限公司**

**联系邮箱：**[service@qxwz.com](mailto:service@qxwz.com)

**官方网站：**[www.qxwz.com](http://www.qxwz.com)

## 目 录

第一章 概述.....	6
1.1 产品简介.....	6
第二章 开发指南.....	7
2.1 开发准备.....	7
2.1.1 注册为开发者 .....	7
2.1.2 购买千寻 FindM 或 FindCM 服务.....	8
2.1.3 进入管理控制台-控制中心-千寻跬步 / 千寻知寸-服务实例，选择服务实例对应管理进行开发者配置工作 .....	8
2.1.4 进入管理控制台-控制中心-千寻跬步 / 千寻知寸-设备服务号，创建设备服务号.....	9
2.1.5 前往控制中心-千寻跬步 / 千寻知寸-应用列表获取 AppKey, AppSecret .....	9
2.1.6 集成 SDK，写入 AppKey, AppSecret, DeviceID ( 设备 SN ) , DeviceType ( 设备类型 ) .....	9
2.1.7 绑定与激活操作 ( 可选 ) .....	9
2.2 接入流程图 .....	10
2.2.1 使用流程.....	11
2.3 配置工程.....	11
2.3.1 so 和头文件引入.....	11
第三章 SDK 接口说明 .....	13

3.1 配置 RTCM.....	13
3.1.1 接口定义.....	13
3.1.2 示例代码.....	13
3.2 启动 RTCM.....	13
3.2.1 接口定义.....	13
3.2.2 示例代码.....	13
3.3 获取 RTCM 数据.....	14
3.3.1 接口定义.....	14
3.3.2 示例代码.....	14
3.4 发送 GGA.....	14
3.4.1 接口定义.....	14
3.4.2 示例代码.....	14
3.5 获取 SDK 返回状态 .....	15
3.6 获取用户账号信息.....	15
3.6.1 接口定义.....	15
3.6.2 示例代码.....	15
3.7 返回状态码.....	16
3.8 基本数据结构 .....	18
3.8.1 差分流数据.....	18
3.8.2 差分 SDK 配置.....	18
第四章 常见问题.....	错误! 未定义书签。
4.1 .....	错误! 未定义书签。



# 第一章 概述

## 1.1 产品简介

千寻位置网差分数据 Linux SDK 为基于嵌入式 Linux 平台的应用提供自动注册差分账号、自动获取差分数据的服务接口包，专注于为广大开发者提供高精度位置服务，本文简称为**差分数据 SDK**。

该 SDK 获取差分数据后，差分数据传输、芯片写入工作完全交由开发者完成，为开发者提供最大灵活性。用户在获取这些服务的同时，不用考虑数据存储，数据安全和繁琐的网络交互等细节。

千寻位置网差分数据 SDK 主要提供以下功能：

用户使用自己的高精度芯片差分算法，使用 SDK 可以获取差分数据，用户可以使用这些差分数据灌进相关定位芯片或算法。

## 第二章 开发指南

### 2.1 开发准备

#### 2.1.1 注册为开发者

开发者需要入驻千寻位置网，在千寻位置官网（qxwz.com）注册为千寻位置开发者，注册成功并通过个人/企业用户认证后，便可成为千寻位置网的开发者。详细信息请见千寻官网帮助文档。

（<https://www.qxwz.com/help-document-settled-platform.html#link-0>）

### 2.1.2 购买千寻 FindM 或 FindCM 服务

### 2.1.3 进入管理控制台-控制中心-千寻跬步 / 千寻知寸-服务实例，选择服务实例对应管理进行开发者配置工作

开发者配置

\* 服务实例接入方式②:

SDK

▼

\* 设备服务号绑定方式②:

自动绑定

▼

\* 设备服务号激活方式②:

自动激活

▼

确定

1. 选择服务实例接入方式为 SDK；
2. 选择设备服务号绑定方式：自动绑定或手动绑定；  
若选择手动绑定，则需对需要连接千寻服务的设备进行绑定操作；
3. 选择设备服务号激活方式：自动激活或手动激活或终端激活方式；  
若选择手动激活，则需在千寻官网管理控制台中进行激活后使用；  
若选择终端激活方式，则需在终端调用千寻激活接口后进行激活后使用。



2.1.4 进入管理控制台-控制中心-千寻跬步 / 千寻知寸-设备服务号，创建设备服务号

创建设备服务号

\* 服务实例:

S000000F8CI/Findm\_20170527\_HgS... ▾

创建即激活:

☒

设备服务号可以创建后立刻激活，提供服务并开始计费。您也可以选择不激活，之后需要使用时再激活。

\* 创建数量:

1

可创建数量为该服务实例购买的最大数量，不可更改

取消

确定

2.1.5 前往控制中心-千寻跬步 / 千寻知寸-应用列表获取 AppKey, AppSecret

服务实例ID	应用标识 ( Appkey )	应用密钥 ( AppSecrete )	应用名称	应用行业	应用细分	操作
114	500096	<a href="#">显示</a> <a href="#">重置</a>				<a href="#">登记应用信息</a>

2.1.6 集成 SDK，写入 AppKey, AppSecret, DeviceID ( 设备 SN )，DeviceType ( 设备类型 )

2.1.7 绑定与激活操作 ( 可选 )

1. 若服务实例配置为需要绑定设备，则需要在控制中心-千寻跬步 / 千寻知寸-设备服务号进行绑定工作，写入 SDK 的 DeviceID 和 DeviceType

绑定设备服务号

\* 设备服务号: D00000000AI

\* 设备SN (DeviceID) :

请输入设备SN

\* 设备类型 (DeviceType) :

请输入设备类型

取消

确定

2. 若服务实例配置为需要激活，前往控制中心-千寻跬步 / 千寻知寸-设备服务号对当前设备进行激活操作。

当前状态: 全部状态

服务实例: 全部服务实例

激活勾选项

续费勾选项

设备服务号:

查询

导出全部

<input type="checkbox"/>	设备服务号	设备服务号密钥	服务实例ID	差分账号	当前状态	设备ID	设备类型	过期时间	操作
<input type="checkbox"/>	D00000000AI	<a href="#">显示</a> <a href="#">重置</a>	114		即将过期	123	234	2017-05-28 23:59:59	<a href="#">解绑</a> <a href="#">续费</a>

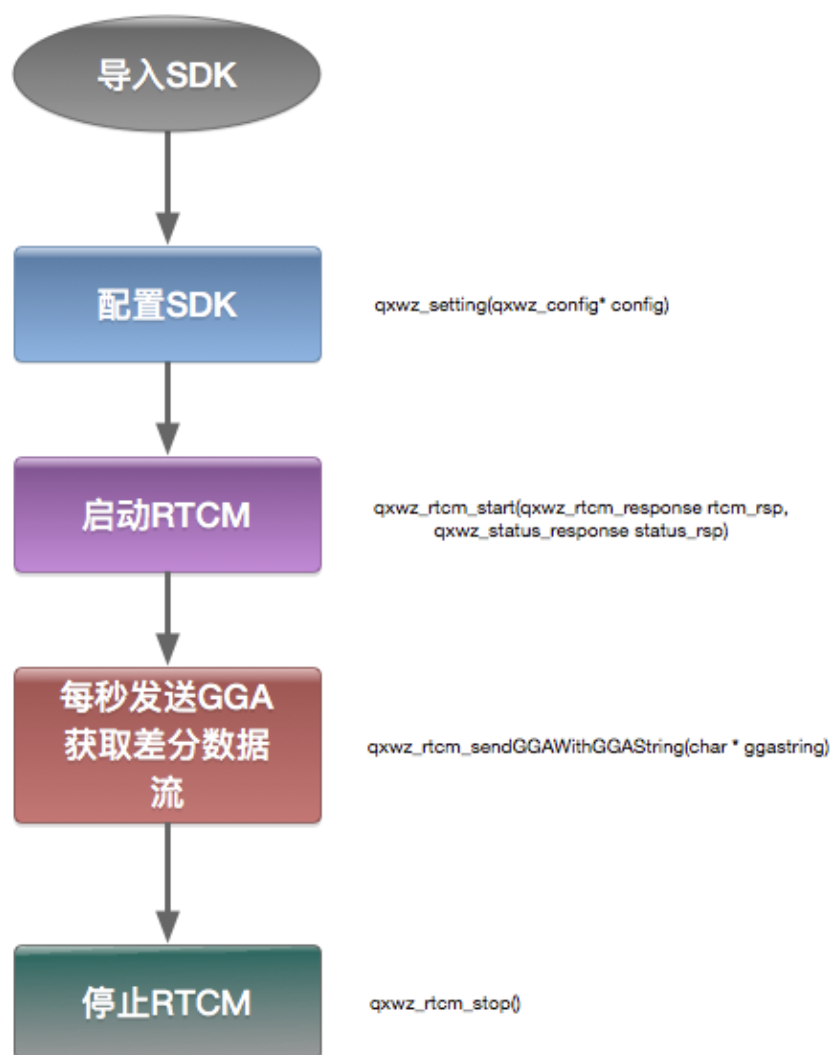
## 2.2 接入流程图

说明：目前 Linux 版差分 SDK 是通过 arm-none-linux-gnueabi-gcc 这个编译器编译出来的。如有特殊平台需要，请联系服务邮箱 [service@wz-inc.com](mailto:service@wz-inc.com)

### 2.2.1 使用流程

1, 工程引入 SDK 包, 调用对应的 API 接口完成操作。

2, 由于 API 接口有先后依赖关系, 列出相关 API 的调用流程 (包括获取差分账号)。如下:



## 2.3 配置工程

### 2.3.1 so 和头文件引入

1, 准备千寻位置网提供的 so 包. 例如 "librtcm\_v0.0.1.so"

2, 将千寻位置网提供的 SDK 拷贝到项目目录下, 拷贝完成的工程目录如图所示:

```
-rwxrwxr-x  1 carry carry 149040 Jun 17 00:11 librtcm_arm_v1.0.0.so
-rwxrwxr-x  1 carry carry   9300 Jun 17 01:02 main
-rw-rw-r--  1 carry carry    937 Jun 17 00:12 main.c
-rw-r--r--  1 carry carry   1555 Jun 17 00:13 qxwz_rtcn.h
carry@ubuntu:~/c$ gcc -g -o main main.c -lrtcm -L. -Wl,-rpath=. -std=c99
```

## 第三章 SDK 接口说明

### 3.1 配置 RTCM

#### 3.1.1 接口定义

```
/**
 * 配置SDK
 */

void qxwz_setting(qxwz_config* config);
```

#### 3.1.2 示例代码

```
qxwz_config config;
config.appkey=" " ;      //申请的appKey
config.appSecret=" " ;   //申请的secret
config.deviceId="00:11:67:44:41:6A "; //设备的唯一ID，建议使用设备的序列号
config.deviceType=" Test device";    //设备类型
qxwz_setting(&config);
```

### 3.2 启动 RTCM

#### 3.2.1 接口定义

```
/**
 * 启动rtcm服务
 */

void qxwz_rtc_start(qxwz_rtc_response rtc_rsp, qxwz_status_response status_rsp);
```

#### 3.2.2 示例代码

```
void qxwz_rtc_response_callback(qxwz_rtc data){
    printf("QXWZ_RTCM_DATA:%s\n",data.buffer);
}
```

```
void qxwz_status_response_callback(qxwz_rtc_status_code code){
    printf("QXWZ_RTCM_STATUS:%d\n",code);
}

qxwz_rtc_start(qxwz_rtc_response_callback,qxwz_status_response_callback);
```

## 3.3 获取 RTCM 数据

### 3.3.1 接口定义

```
/**
 * RTCM差分数据回调函数
 * @param qxwz_rtc_response
 */

typedef void (*qxwz_rtc_response)(qxwz_rtc data);
```

### 3.3.2 示例代码

```
// 示范: 根据差分账号获取差分数据流
void qxwz_rtc_response_callback(qxwz_rtc data){
    printf("QXWZ_RTCM_DATA:%s\n",data.buffer);
}
```

## 3.4 发送 GGA

### 3.4.1 接口定义

```
/**
 * 向ntrip服务器发送GGA字符串用来获取rtcm数据
 * @param ggastring
 */

void qxwz_rtc_sendGGAWithGGAString(char* ggastring);
```

### 3.4.2 示例代码

```
qxwz_rtc_sendGGAWithGGAString("$GPGGA,000001,3112.518576,N,12127.901251,E,1,8,1,0,M,-32,M,3.0*4B\r\n");
```

## 3.5 获取 SDK 返回状态

```
/**
 * RTCM服务状态码回调函数
 * @param qxwz_status_response
 */

typedef void (*qxwz_status_response)(qxwz_rtc_status code);
```

## 3.6 获取用户账号信息

### 3.6.1 接口定义

```
/**
 *
 * 查询用户账号信息
 *
 * @return qxwz_account_info*
 *
 */

qxwz_account_info* getqxwzAccount(void);
```

当用户账号还有 10 天将到期时，SDK 会通过状态码的方式通知用户账号即将到期，状态码为 2011。

当用户账号已过期时，SDK 会通知用户账号已过期，状态码为 2010。详见下一小节

### 3.6.2 示例代码

```
/**
 * SDK 账号信息
 */
typedef struct {
    char *appkey;
    char *deviceId;
    char *deviceType;
    char *serviceType;
    time_t expire_time; /*自1970年1月1日的秒数*/
```

```
char *NtripUserName;  
char *NtripPassword;  
} qxwz_account_info;
```

```
qxwz_account_info *p_account_info = NULL;  
void get_qxwz_sdk_account_info(void)  
{  
    p_account_info = getqxwzAccount();  
    if(p_account_info->appkey != NULL) {  
        printf("appkey=%s\n",p_account_info->appkey);  
    }  
    if(p_account_info->deviceID != NULL) {  
        printf("deviceID=%s\n",p_account_info->deviceID);  
    }  
    if(p_account_info->deviceType != NULL) {  
        printf("deviceType=%s\n",p_account_info->deviceType);  
    }  
  
    if(p_account_info->NtripUserName != NULL) {  
        printf("NtripUserName=%s\n",p_account_info->NtripUserName);  
    }  
    if(p_account_info->NtripPassword != NULL) {  
        printf("NtripPassword=%s\n",p_account_info->NtripPassword);  
    }  
    printf("expire_time=%d\n",p_account_info->expire_time);  
}
```

## 3.7 返回状态码

状态码	状态说明
1000	ntrip 连接到服务器
1001	ntrip 断开服务器
1002	appKey 验证失败
1003	appKey 验证成功



1004	网络异常
1005	NTRIP 用户达到上限
1006	NTRIP 用户不存在
1007	NTRIP 用户认证成功
1011	发送的 GGA 不合法
1012	GGA 发送超时
1013	正在连接 ntrip 服务器
1014	正在接收 ntrip 服务器数据
1015	非法 APP KEY
1016	非法 APP SECRET
1017	非法 Device type
1018	非法 Device id
1019	无法获取差分用户
1020	SDK 内部错误
1021	Ntrip 播发数据正常
1022	Ntrip 认证失败
2001	缺少参数
2002	账号不存在
2003	账号重复
2004	错误密码
2005	账号不可用
2006	没有有效的账号

2007	POPUser 不存在
2008	服务端内部错误
2010	账号已过期，需续费
2011	账号即将过期

## 3.8 基本数据结构

### 3.8.1 差分流数据

```

/**
 * qxwz_rtcM 数据，包括数据 ( buffer ) ，和数据长度(length)
 */

typedef struct{
    char* buffer;
    size_t length;
}qxwz_rtcM;

```

### 3.8.2 差分 SDK 配置

```

/**
 * 启动SDK配置信息
 * deviceId 设备ID号
 * deviceType 设备类型
 */

typedef struct{
    char * appkey;
    char * appSecret;
    char * deviceId;
    char * deviceType;
}qxwz_rtcM_setting;

```

