

用户指南

——TruCardio 移植安卓 Java

版本 A.1

更新时间 2025 年 4 月 7 日

修订历史:

2025 年 1 月 30 日——zhangks——生成初版说明。

2025 年 4 月 7 日——yangmh——根据使用体验完善。

目录

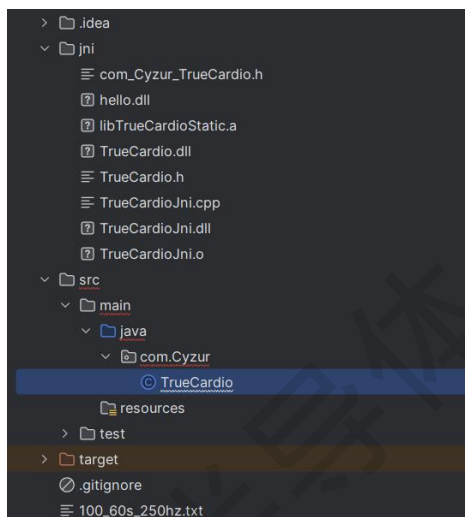
一、 概述	2
二、 工程架构	2
三、 移植指南	2
1. 打开项目与外部工具添加	2
2. 生成 JNI 头文件	6
3. 编写 C 文件并编译成 dll (或 so) 文件	7
4. 运行	8

一、概述

本文档主要是介绍 TruCardio 算法在 java 平台上的移植说明, 使用 jni 封装二次封装 TruCardio 接口。

二、工程架构

本工程是使用 idea 构造, 主要文件包括如下



(1) jni 文件夹

- 1) TruCardio.h 和 libTrueCardioStatic.a, 这是曦成半导体提供的 TruCardio 解析算法 C++ 版本接口和静态库 (动态库未用到)。
- 2) com_Cyzur_TrueCardio.h 和 TruCardioJni.cpp, 这是基于 jni 对 TruCardio 接口二次封装的文件。
- 3) TruCardioJni.dll, 这是对 TruCardioJni.cpp 打包生成的动态库。

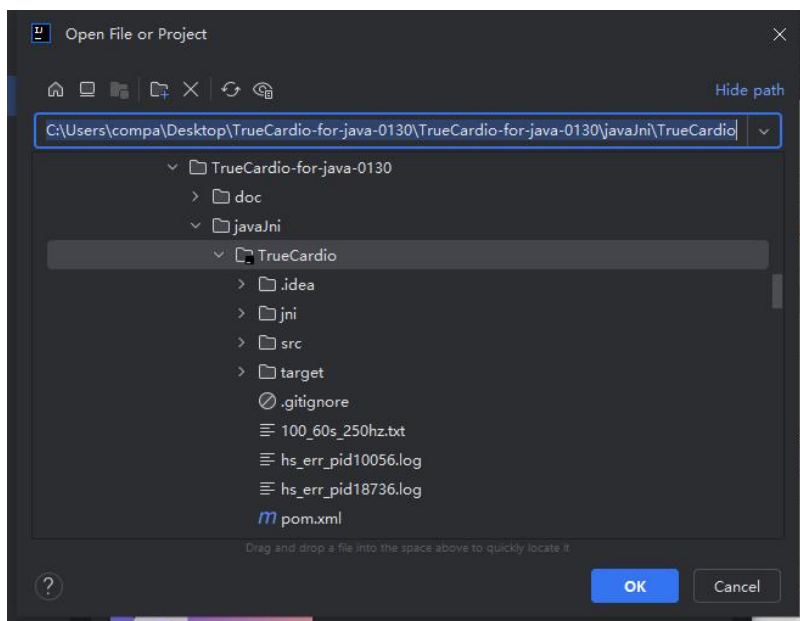
(2) Src 文件夹

src/main/java/com/Cyzur/TruCardio.java, 这是使用 TruCardioJni.dll 的参考例程。该例程读取本地 ECG 文件, 然后输出心电解析结果。

三、移植指南

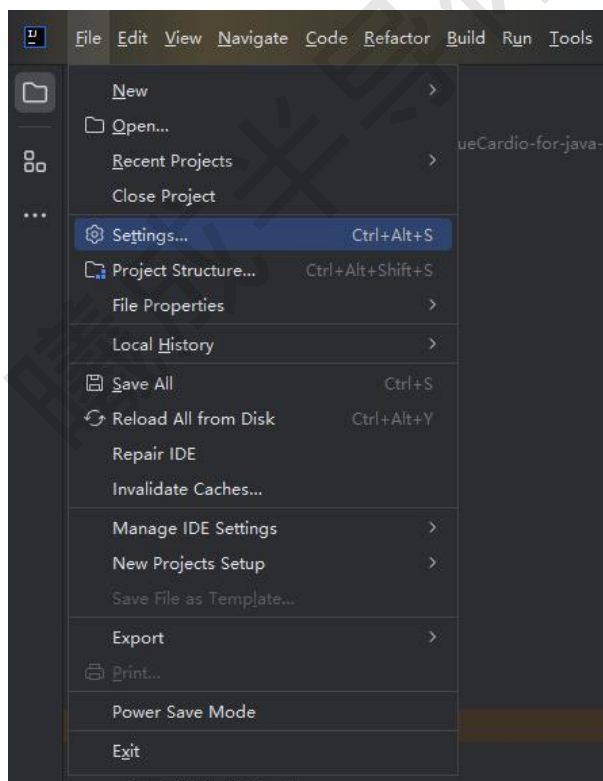
1. 打开项目与外部工具添加

- (1) 在 idea 中打开 JavaJni 下的 TrueCardio 文件夹

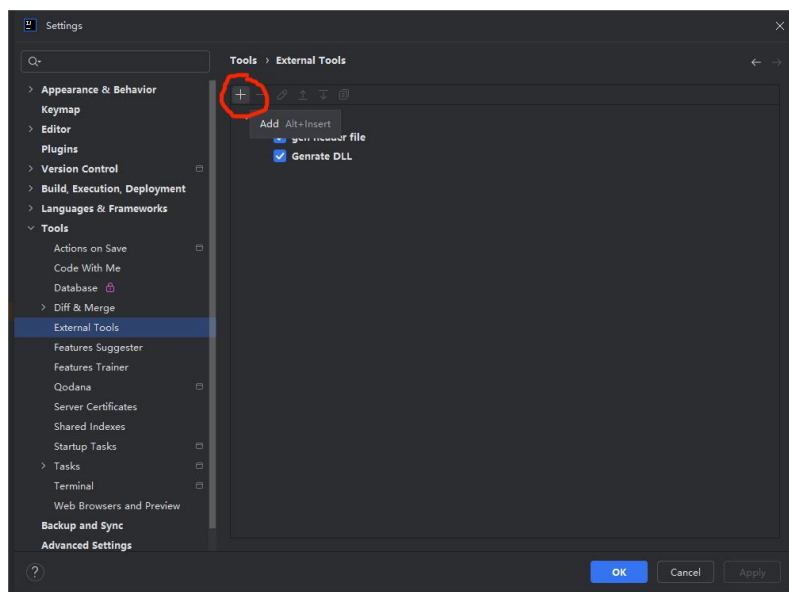


(2) 添加外部工具 Gen Header File

在 idea 中点击 File-Settings



找到 Tools-External Tools,点击左上角+号



依次设置 Name、Program、Arguments、Working directory，参数分别为：

Name: Gen Header File

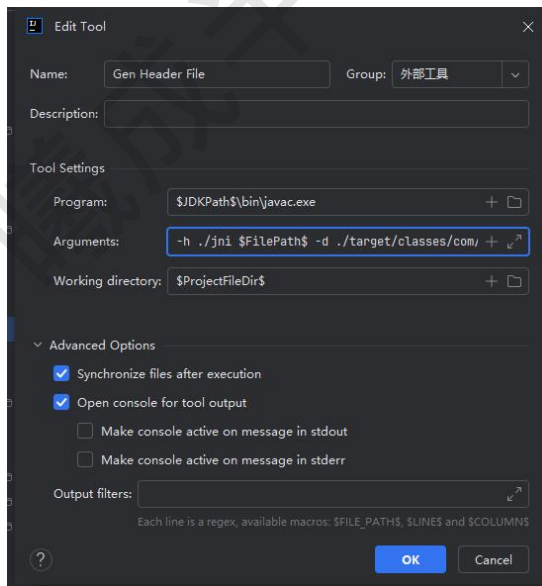
Program: `$JDKPath$\bin\javac.exe`

(注意，如果未在 idea 中自动安装 JDK，需设置

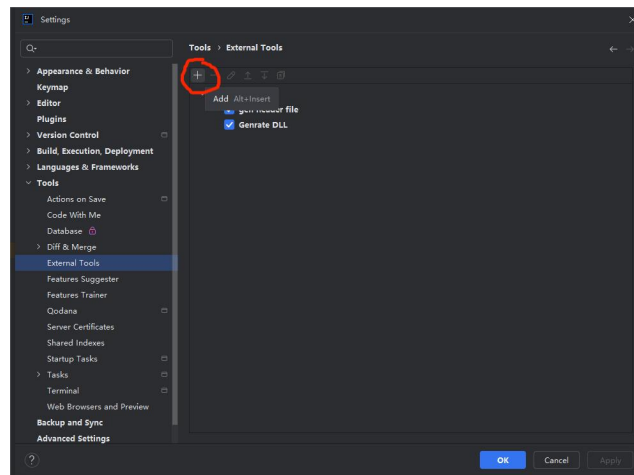
为实际 JDK 路径或把 JDK 路径添加到 idea 项目中，JDK 版本需大于 17)

Arguments: `-h ./jni $FilePath$ -d ./target/classes/com/Cyzur/`

Working directory: `$ProjectFileDir$`



(3) 添加外部工具 Genrate DLL



依次设置 Name、Program、Arguments、Working directory，参数分别为：

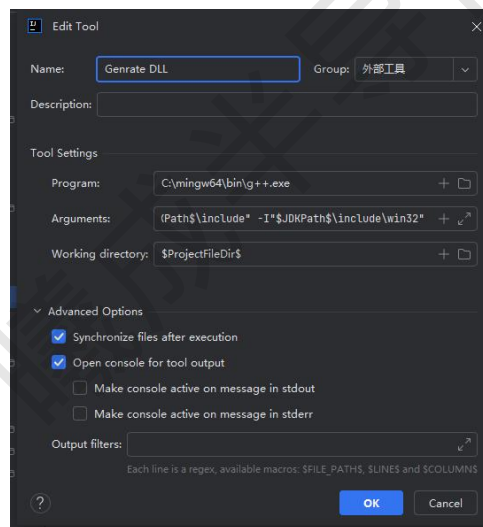
Name: Generate DLL

Program: C:\mingw64\bin\g++.exe (注意，此处需替换为实际的 g++ 路径)

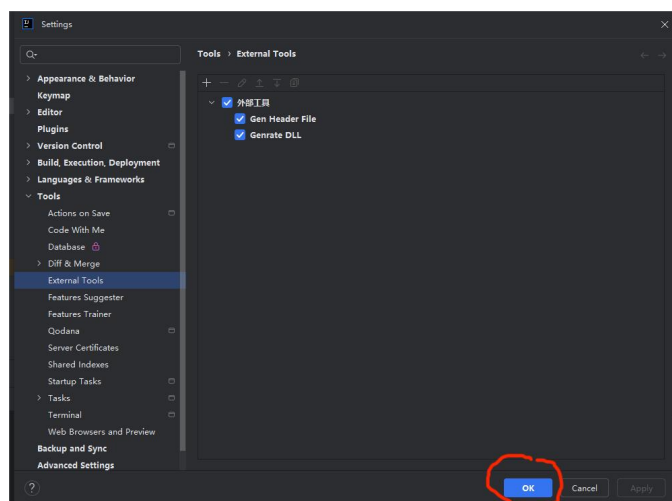
Arguments: -Wl,--add-stdcall-alias -shared -o "\$FileDir\$\TrueCardioJni.dll"

\$FilePath\$ jni/libTrueCardioStatic.a -I"\$JDKPath\$\include" -I"\$JDKPath\$\include\win32"

Working directory: \$ProjectFileDir\$



点击 OK 以保存我们设置好的外部工具



2. 生成 JNI 头文件

(1) TruCardio.java 的主要结构如下: 包含读取本地文件函数 readIntArrayFromFile, 加载动态库, native 关键字将函数 TruCardio()声明为本地函数, main 函数应用。

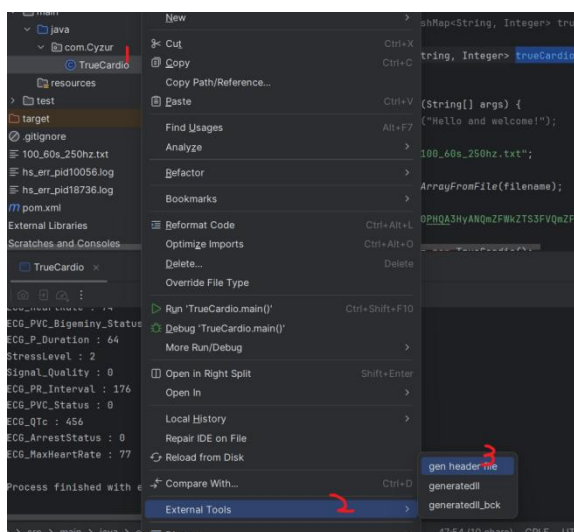
```
1 public static int[] readIntArrayFromFile(String filename) {...}

2 // 修改为你的动态库路径, 或使用 LoadLibrary.
3 static {
4     System.load( filename: "E:\\9.TrueCardio\\javaJni\\TrueCardio\\jni\\TrueCardioJni.dll");
5 }

6 // private native HashMap<String, Integer> trueCardio(int[] ecgData, String cnlicense);
7 1 usage
8 public native HashMap<String, Integer> trueCardio(int[] ecgData, String cnlicense);

9
10 public static void main(String[] args) {
11     System.out.printf("Hello and welcome!");
12
13     String filename = "100_60s_250hz.txt";
14
15     int[] ecg = readIntArrayFromFile(filename);
```

(2) 然后在 TruCardio.java 上右右键选择 external tool-->Gen Header File。生成./jni/com_Cyzur_TruCardio.h



3. 编写 C 文件并编译成 dll（或 so）文件

（1）TruCardioJni.cpp 主要包含三部分内容：

1) 将 java 传入的参数改写为 C++形式的参数

```
JNIEXPORT jobject JNICALL Java_com_Cyzur_TrueCardio_trueCardio(JNIEnv *env, jobject obj, jint len, jint *nativeEcgData, const char *nativeLicense) {
    // 将 Java int 数组转为 C++ 向量
    jsize len = env->GetArrayLength(ecgData);
    jint *nativeEcgData = env->GetIntArrayElements(ecgData, nullptr);
    std::vector<int32_t> cppEcgData(nativeEcgData, nativeEcgData + len);

    // 将 Java 字符串转为 C++ 字符串
    const char *nativeLicense = env->GetStringUTFChars(cnLicense, nullptr);
```

2) 调用 C++ 库函数

```
// 调用 C++ 库函数
std::unordered_map<std::string, int> cppmapResult = TrueCardio(cppEcgData, nativeLicense);
```

3) 将 C++形式的返回值改写为 java 形式的返回值

```
// 创建 java.util.HashMap 对象
jclass hashMapClass = env->FindClass("java/util/HashMap");
jmethodID hashMapInit = env->GetMethodID(hashMapClass, "<init>", "()V");
jclass integerClass = env->FindClass("java/lang/Integer");
jmethodID integerConstructorID = env->GetMethodID(integerClass, "<init>", "(I)V");
jmethodID putMethodID = env->GetMethodID(hashMapClass, "put", "(Ljava/lang/Object;Ljava/lang/Object;)V");
jobject hashMap = env->NewObject(hashMapClass, hashMapInit);

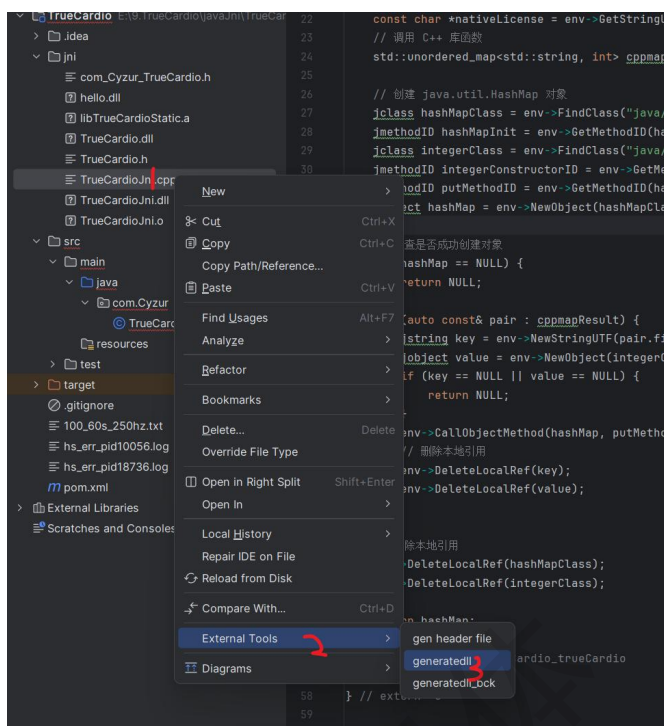
// 检查是否成功创建对象
if (hashMap == NULL) {
    return NULL;
}

for (auto const& pair : cppmapResult) {
    jstring key = env->NewStringUTF(pair.first.c_str());
    jobject value = env->NewObject(integerClass, integerConstructorID, pair.second);
    if (key == NULL || value == NULL) {
        return NULL;
    }
    env->CallObjectMethod(hashMap, putMethodID, key, value);
    // 删除本地引用
    env->DeleteLocalRef(key);
    env->DeleteLocalRef(value);
}

// 删除本地引用
env->DeleteLocalRef(hashMapClass);
env->DeleteLocalRef(integerClass);

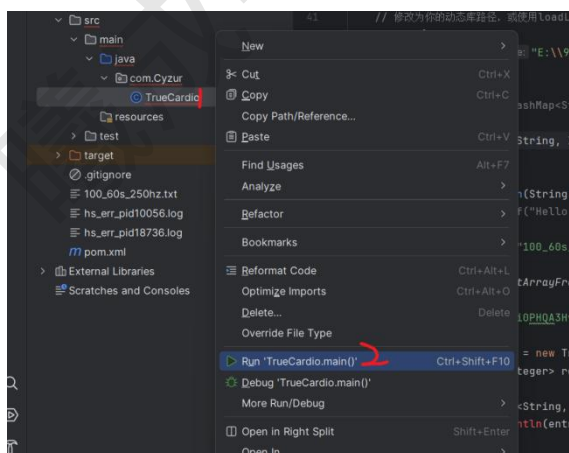
return hashMap;
} // Java_com_Cyzur_TrueCardio_trueCardio
```

（2）然后在 jni/TrueCardioJni.cpp 上右右键选择 external tool-->Generate DLL。生成./jni/TruCardioJni.dll 文件。



4. 运行

(1) 然后在 src/main/java/com/Cyzur/TrueCardio.java 上右右键选择 Run。



(2) 结果如下：

```
HRVrMSSD : 29
RPEAKS_NUM : 73
ECG_AF_Status : 0
ECG_MinHeartRate : 68
ECG_QRS_Complax : 104
HRVSDNN : 25
ECG_PVC-Trigeminy_Status : 0
ECG_RSP : 14
ECG_RRmax : 812
ECG_HeartRate : 74
ECG_PVC-Bigeminy_Status : 0
ECG_P_Duration : 64
StressLevel : 2
Signal_Quality : 0
ECG_PR_Interval : 176
ECG_PVC_Status : 0
ECG_QTc : 456
ECG_ArrestStatus : 0
ECG_MaxHeartRate : 77
```

此处读取了文件夹中 100_60s_250hz.txt 中的数据作为输入进行分析