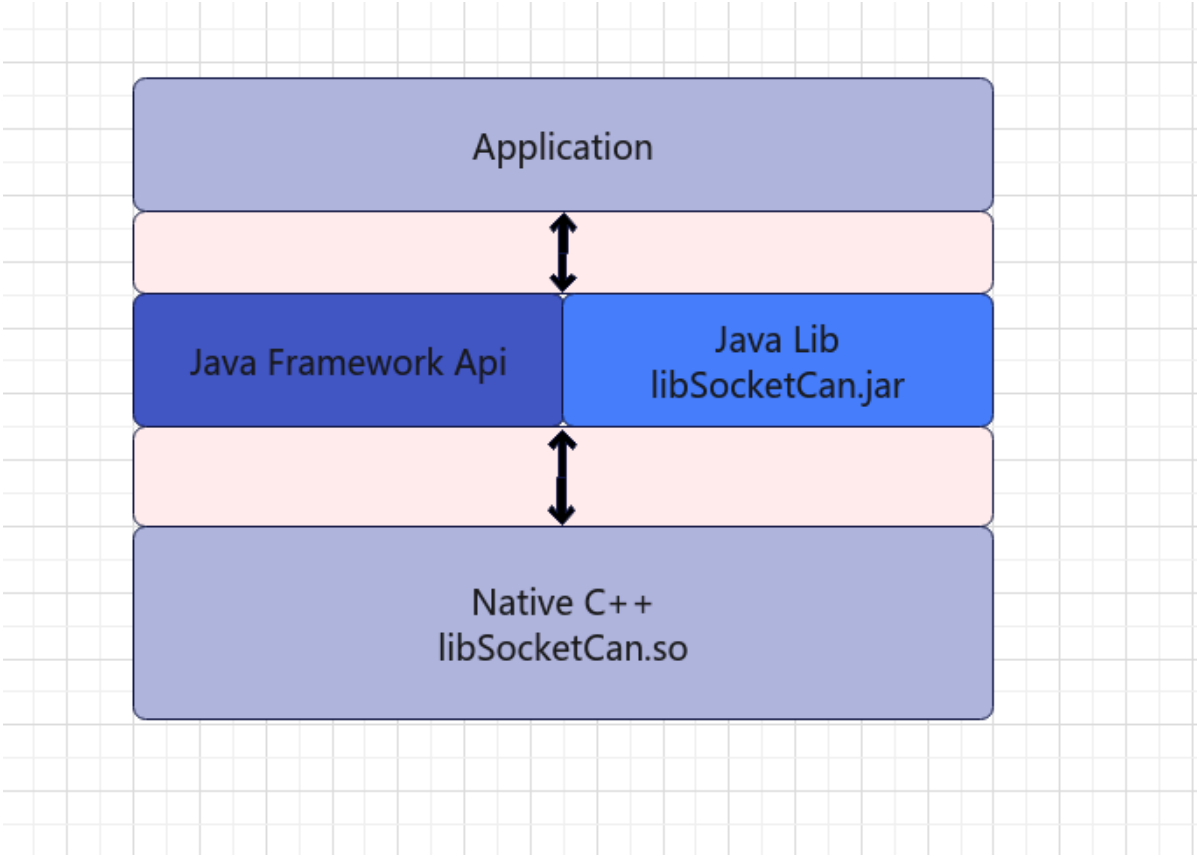
介绍

libSocketCan SDK 概述

libSocketCan SDK 是一组控制 imx8m tpc_1xx 安卓设备 CAN 总线的 API 开发库，包括(配置、打开、关闭、发送和接收 CAN 数据帧)，SocketCan 仅支持 can0 接口的控制

libSocketCan SDK 为应用程序提供 API 模块，用于控制 CAN 进行各种操作，该 SDK 的软件栈如下图所示



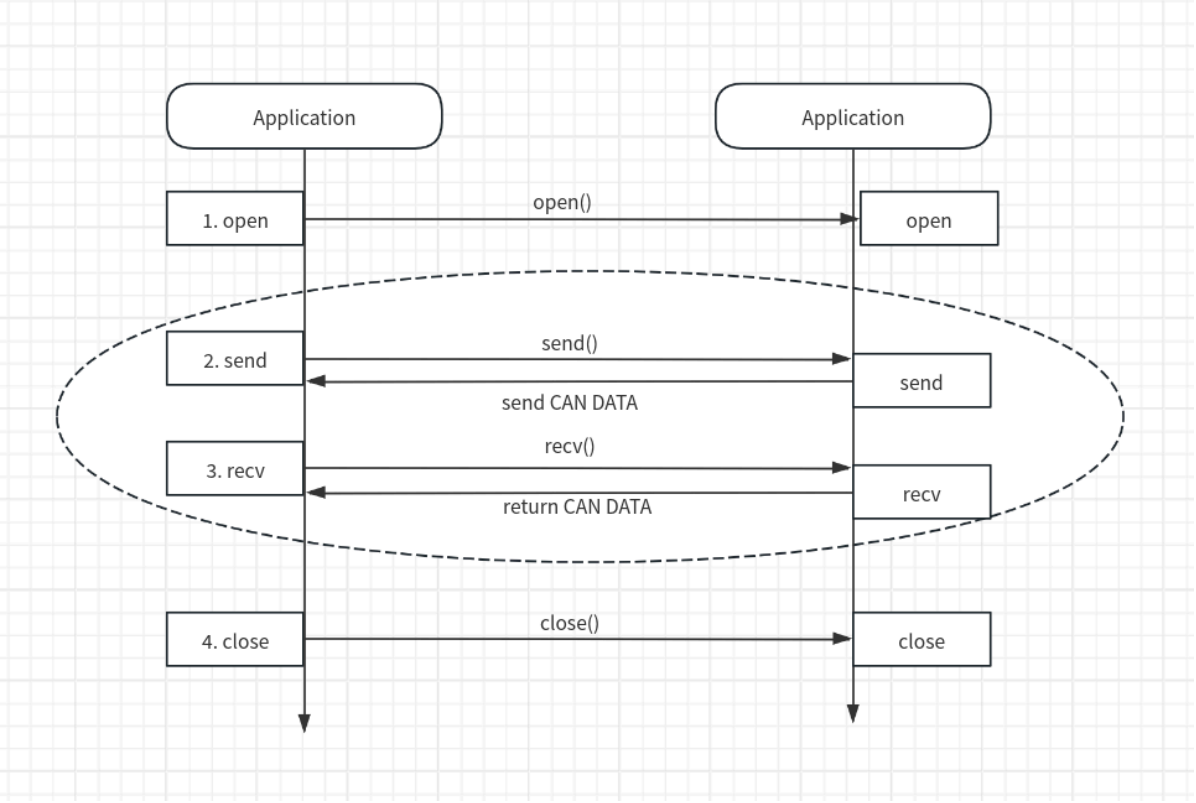
libSocketCan API 主要包括两个部分

- libSocketCan.jar Java 方法库
- libSocketCan.so 本地方法库

用法

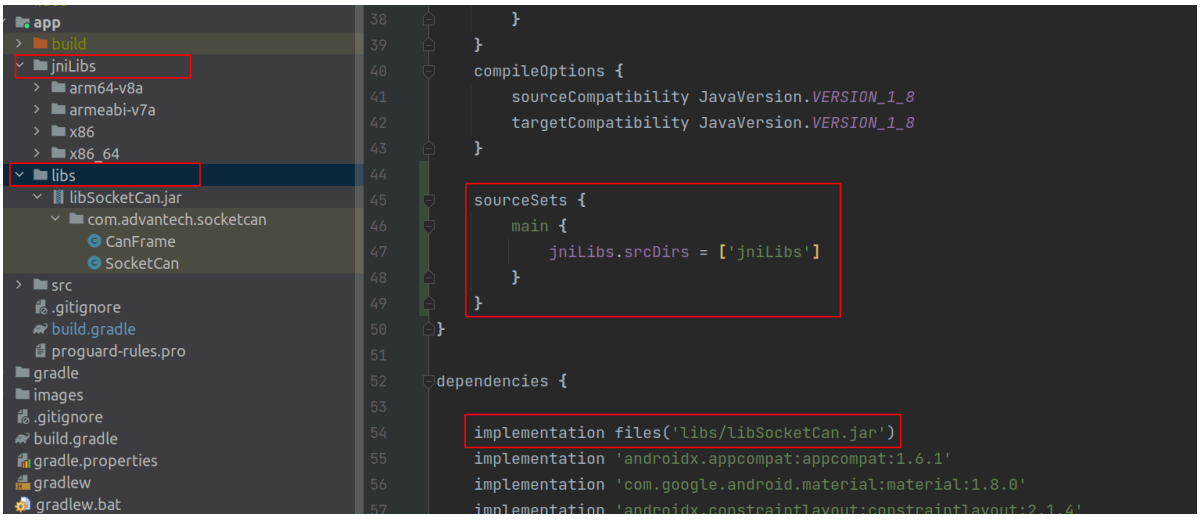
SDK API 调用流程

我们导入成功 libSocketCan SDK 之后，就可以开始开发我们的 app，具体的使用流程如下图所示



在 Android Studio 中导入 SDK

要使用 application 访问 libSocketCan 的功能，您必须将 libSocketCan.jar 和 libSocketCan.so 导入到您的 AndroidStudio 工程中。对应的目录及配置如下图所示。



API 接口说明

接口

接口名	方法成员
OnFrameDataReceivedListener	*can data 数据回调* **void onDataReceived(CanFrame canFrame)**

类

类名	变量成员	方法成员
CanFrame	*can id* **public int canId** *can data 是一个 byte 数组* **public byte[] data** *can data length 数据长度* **public int canDlc** *扩展帧标识* **private boolean isExtended** *远程帧标识* **private boolean isRemote**	*构造方法* **public CanFrame(int canId, byte[] data, int canDlc)** *构造方法* **public CanFrame(int canId, byte[] data, int canDlc, boolean isExtended, boolean isRemote)**
ErrorCodeEnum	*数据发送失败* **ERR_SEND_FAIL** *参数解析错误* **ERR_INVALID_ARGUMENT** *can data 数据长度匹配错误* **ERR_MISSING_CAN_DLC** *没有找到目标类* **ERR_NOT_FIND_CLASS** *数据解析错误* **ERR_PARSING_DATA** *打开或关闭 socket fd 错误* **ERR_FD_STATUS** *bind socket failed* **ERR_BIND_SOCKET** *端口没有打开* **ERR_SOCKET_NOT_OPEN** *错误码* **private final int errorCode** *对应错误码的描述* **private final String value**	*根据错误码获取描述* **public static String getValue(int errorCode)**
Mask	*掩码 id* **private int filterId** *掩码滤波* **private int mask**	none

类名	变量成员	方法成员
SocketCan	**private Context mContext** **private boolean isOpened** **private int mSocketFd** **private int mPort** **private String mSpeed** **private Map<Integer, Mask> maskFilters** **private OnFrameDataReceivedListener mDataReceivedListener**	*构造函数，用于构造 SokcetCan 实例* **public SocketCan(Context context, OnFrameDataReceivedListener listener)** *打开 can 接口* **public synchronized int open(int port, String speed)** 关闭 can0 接口 **public synchronized int close()** 发送数据帧 **public int send(CanFrame frame)** 接收数据 **public CanFrame recv()**

函数方法

SocketCan

- Syntax

```
public SocketCan(Context context, OnFrameDataReceivedListener listener)
```

- Description

构造 SocketCan 实例

- Parameters
  - context：上下文
  - listener：用于监听 can 数据的回调接口，如果传入的 listener 不为 null，则须实现回调方法接收数据，否则需主动起一个线程调用 recv 方法接收数据
- Returns

none

- Remarks

none

open

- Syntax

```
public synchronized int open(int port, String speed)
```

- Description

打开 can 总线

- Parameters
  - port：can 接口序号，比如 0 或 1

    - speed：用于配置 CAN 接口的速率，一般情况下可将速率设置为 100 000，250 000
- Returns

返回值为负值代表打开失败，反之成功

- Remarks

目前只支持 can0 接口

close

- Syntax

```
public synchronized int close()
```

- Description

关闭 can0

- Parameters

none

- Returns

返回值为 0 表示关闭成功，否则失败

- Remarks

none

send

- Syntax

```
public int send(CanFrame frame)
```

- Description

发送 can 数据

- Parameters

CanFrame 结构体，发送数据时需要构造 CanFrame 结构，传入 canid、data、len 等

- Returns

返回值为负数代表失败，发送成功会返回一个发送数据 data 的长度

- Remarks

仅支持单帧发送

recv

- Syntax

```
public CanFrame recv()
```

- Description

接收 can 数据

- Parameters

none

- Returns

返回值一个 CanFrame

- Remarks

调用该函数需要其一个读线程，一直监听数据