NFC芯片类型(调试过的):

PN547, PN548, PN553, PN557,

PN60T, PN67T, PN80T, PN81T,

SN100T

[NFC不同的卡](nfc卡差异.xlsx)

软件方面

nq-nci.c

默认被所有项目使用, 一些老的项目可能用pn553.c.

nfc使用了i2c的驱动框架, 在NFC进行硬件检查时, 会通过i2c分别向芯片发送复位信号{0x20, 0x00, 0x01, 0x00} 和初始化信号{0x20, 0x01, 0x00}, 之后通过i2c\_master\_recv(client, nci\_init\_rsp, sizeof(nci\_init\_rsp)) 将初始化后的硬件信息存入nci\_init\_rsp, 硬件信息的后4字节分别是chiptype(芯片类型) rom\_version(硬件rom版本) fw\_major(最低版本) fw\_minor(最高版本).

vendor/nxp/pn\*/device-nfc\_for\_QC.mk

nxp默认使用该文件, 该mk文件将通过宏控将conf文件, so固件, service包等复制到对应的目录, 如果不需要大批移植代码, 只需要验证conf文件即可.

device/qcom/sepolicy/legacy/vendor/common/hal\_nfc.te

nfc hal层的权限文件, 如果需要在hal层读取文件节点, 需要在此文件添加权限. 存在权限文件不同名的可能.

hardware/nxp/nfc/halimpl/utils/phNxpConfig.cpp

nfc兼容的核心文件. 当要兼容的nfc是共代码的, 则可以在该文件读取驱动创建的节点区分配置文件.

硬件方面

以MC557为例: [MC557\_SCH](MC557_MB_SCH_V1.00(20210810)-1530.pdf)

nfc日志分析

以PN557 android11(SLB761X) 的为例, [NFC\_LOG](nfclog.txt)