

# Rockchip Introduction USB SQ Tool

文件标识：RK-SM-YF-195

发布版本：V1.6.0

日期：2024-04-16

文件密级：绝密 秘密 内部资料 公开

## 免责声明

本文档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2023 瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园A区18号

网址：[www.rock-chips.com](http://www.rock-chips.com)

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：[fae@rock-chips.com](mailto:fae@rock-chips.com)

## 前言

### 概述

由于 Rockchip 平台的 USB PHY 手册没有对外发布，且芯片的 TRM 文档也缺少 USB PHY 寄存器的详细说明，因此，当开发者遇到与 USB 信号质量相关的问题时，缺少有效的软件方法或工具来调试 USB 信号问题。为了解决该问题，Rockchip 发布了 USB SQ Tool 工具来简化 USB PHY 信号质量的调试工作，该工具支持 Rockchip 大部分芯片，可用于 USB PHY 信号一致性测试命令的查询、信号质量调整命令的查询，以及根据调整命令自动生成对应的 PHY 驱动代码。

### 产品版本

芯片名称	内核版本
RK1808、RK2118、RK3036、RK312X、RK3228、RK3288、RK3308、RK3308B-S、RK3326、RK3326S、RK3328、RK3399、RK3528、RK3562、RK3566、RK3568、RK3576、RK3588、RK3588S、RV1103、RV1106、RV1109、RV1126、PX30、PX30S	Linux-4.4、Linux-4.19、Linux-5.10、Linux6.1、RT-Thread

### 读者对象

本文档（本指南）主要适用于以下工程师：

硬件工程师

软件工程师

技术支持工程师

### 修订记录

日期	版本	作者	修改说明
2021-09-16	V0.0.1	郑见炜	初始版本
2022-01-10	V1.0.0	郑见炜	增加 RK3588、RK3588S、RK3326S、PX30S、RK3308、RK3228、RK3328 平台说明
2022-04-18	V1.1.0	郑见炜	增加 RV1103、RV1106 平台说明
2022-05-06	V1.2.0	吴良峰	增加 RV1103、RV1106 Tuning 说明，更新 RK3588 USB3.1 Tuning 说明，修订格式
2022-06-25	V1.3.0	郑见炜	增加 RV1109、RV1126、RK3036、RK312X Tuning 说明
2023-01-15	V1.4.0	郑见炜	增加 RK3528、RK3562 平台说明，增加 RK3566、RK3568 USB3.1 Tuning 说明
2023-07-18	V1.5.0	郑见炜	增加 RK1808 平台说明，增加 RK3399 USB3.1 Tuning 说明
2024-04-16	V1.6.0	郑见炜	增加 RK3576、RK2118 平台说明

# 目录

## Rockchip Introduction USB SQ Tool

1. Rockchip USB PHY 信号调整说明
2. Rockchip USB SQ TOOL 界面说明
  - 2.1 Rockchip USB SQ TOOL 主界面说明
  - 2.2 RK1808 USB2.0 Tuning 界面说明
  - 2.3 RK3399 USB2.0 Tuning 界面说明
  - 2.4 RK3568 USB2.0 Tuning 界面说明
  - 2.5 RK3588 USB2.0 Tuning 界面说明
  - 2.6 RV1106 USB2.0 Tuning 界面说明
  - 2.7 RV1109 USB2.0 Tuning 界面说明
  - 2.8 RK1808 USB3.1 Tuning 界面说明
  - 2.9 RK3399 USB3.1 Tuning 界面说明
  - 2.10 RK3528 USB3.1 Tuning 界面说明
  - 2.11 RK3588 USB3.1 Tuning 界面说明
3. Rockchip USB PHY Tuning 代码生成说明
  - 3.1 RK1808 USB2.0 PHY
  - 3.2 RK2118 USB2.0 PHY
  - 3.3 RK3036/RK312X USB2.0 PHY
  - 3.4 RK3228 USB2.0 PHY
  - 3.5 RK3288 USB2.0 PHY
  - 3.6 RK3308B\_S/RK3308 USB2.0 PHY
  - 3.7 RK3326S/PX30S/RK3326/PX30/RK3328 USB2.0 PHY
  - 3.8 RK3399 USB2.0 PHY
  - 3.9 RK3528 USB2.0 PHY
  - 3.10 RK3562 USB2.0 PHY
  - 3.11 RK3568/RK3566 USB2.0 PHY
  - 3.12 RK3576 USB2.0 PHY
  - 3.13 RK3588 USB2.0 PHY
  - 3.14 RV1103/RV1106 USB2.0 PHY
  - 3.15 RV1109/RV1126 USB2.0 PHY
  - 3.16 RK1808 USB3.1 PHY
  - 3.17 RK3399 USB3.1 PHY
  - 3.18 RK3528/RK3562/RK3566/RK3568 USB3.1 PHY
  - 3.19 RK3576 USB3.1 PHY
  - 3.20 RK3588 USB3.1 PHY

# 1. Rockchip USB PHY 信号调整说明

---

当 USB 模块在实际应用场景中，遇到信号质量相关的问题时，可以通过优化硬件电路或者软件 Tuning USB PHY 寄存器来解决信号问题。无论是硬件或者软件优化的方法，建议都先测试 USB 眼图，根据眼图测试报告来分析和调试 USB PHY 信号。本文档说明的 Rockchip USB SQ Tool 是软件优化方法，可以有效改善如下常见的 USB 问题：

- USB 眼图指标测试失败问题；
- USB 枚举失败的问题（如：信号质量问题或者 PHY 供电压差问题导致的枚举失败）；
- USB 连接外设会自动发生异常断开；
- USB 连接外设拔掉无法检测到断开事件；

注意：由于 USB 信号相关问题都跟硬件环境关系非常大，比如：USB 线缆质量较差、线缆长度太长、USB 走线或线缆阻抗太大等，所以遇到 USB 信号相关问题时，应该优先考虑优化硬件环境。只有在硬件环境无法优化的情况下，再考虑进行软件信号调整。

## 2. Rockchip USB SQ TOOL 界面说明

---

### 2.1 Rockchip USB SQ TOOL 主界面说明

目前 USB SQ Tool 在 USB2.0 的 Tuning 支持的平台有：RK1808、RK2118、RK3036、RK312X、RK3228、RK3288、RK3308、RK3308B-S、RK3326、RK3326S、RK3328（RK3328 目前只支持 USB2.0 OTG 以及 USB2.0 HOST 口的 Tuning，USB3.0 的 U2PHY 暂不支持）、RK3399、RK3528、RK3562、RK3566、RK3568、RK3576、RK3588、RK3588S、RV1103、RV1106、RV1109、RV1126、PX30、PX30S。USB3.1 Tuning 支持的平台有：RK1808、RK3399、RK3528、RK3562、RK3566、RK3568、RK3576、RK3588。其他平台待开发验证后会陆续推出。该工具目前仅适用于 Windows 平台。

Rockchip USB SQ Tool 工具下载路径为：<https://redmine.rock-chips.com/documents/113>

双击打开 Rockchip USB SQ Tool.exe 文件，首先可以看到的是平台选择界面，如图 1 所示，点击平台选择界面中的按键，即可进入对应的平台调整界面。

## 请选择平台：



图1 USB SQ TOOL Main Window

有些平台的 PHY Tuning 界面是一致的，所以只以其中一个平台为例进行说明。USB2.0 PHY Tuning 界面示例见表 1，USB3.1 PHY Tuning 界面示例见表 2。

表1 USB2.0 PHY Tuning 界面示例

平台	USB2.0 PHY Tuning 界面示例
RK1808	<a href="#">RK1808 USB2.0 Tuning 界面说明</a>
RK3036、RK312X、RK3228、RK3308、RK3326、RK3328、RK3399、PX30	<a href="#">RK3399 USB2.0 Tuning 界面说明</a>
RK3308B-S、RK3326S、RK3528、RK3562、RK3566、RK3568、PX30S	<a href="#">RK3568 USB2.0 Tuning 界面说明</a>
RK3576、RK3588、RK3588S	<a href="#">RK3588 USB2.0 Tuning 界面说明</a>
RV1103、RV1106、RK2118	<a href="#">RV1106 USB2.0 Tuning 界面说明</a>
RV1109、RV1126	<a href="#">RV1109 USB2.0 Tuning 界面说明</a>

表2 USB3.1 PHY Tuning 界面示例

平台	USB3.1 PHY Tuning 界面示例
RK1808	<a href="#">RK1808 USB3.1 Tuning 界面说明</a>
RK3399	<a href="#">RK3399 USB3.1 Tuning 界面说明</a>
RK3528、RK3562、RK3566、RK3568	<a href="#">RK3528 USB3.1 Tuning 界面说明</a>
RK3576、RK3588、RK3588S	<a href="#">RK3588 USB3.1 Tuning 界面说明</a>

注意：PHY Tuning 请按照工具页面中的顺序进行调整，只有在前面的调整步骤没有改善或者无法满足要求的情况下，再开启下一个步骤的调整（有些参数的调整可能对结果影响很小，那么可以忽略跳过这些步骤）。USB信号的调整具有一定的风险，不可盲目的调整，最好能测试眼图时对着眼图去适当的调整，调整到能满足需求即可，而不是调整越大越好。用户需要自己把控调整风险。

## 2.2 RK1808 USB2.0 Tuning 界面说明

RK1808 USB2.0 Tuning 界面如下图所示，所有的 Tuning 项与 RK3568 平台的 USB2.0 Tuning 项相同，不同之处在于 RK1808 平台的USB2.0 PHY 寄存器有高 16 位的“write\_enable bit”，所以无需读取寄存器的初始值。RK1808 USB2.0 信号调整参数的说明详见后续 [RK3568 USB2.0 Tuning 界面说明](#) 章节的参数说明。

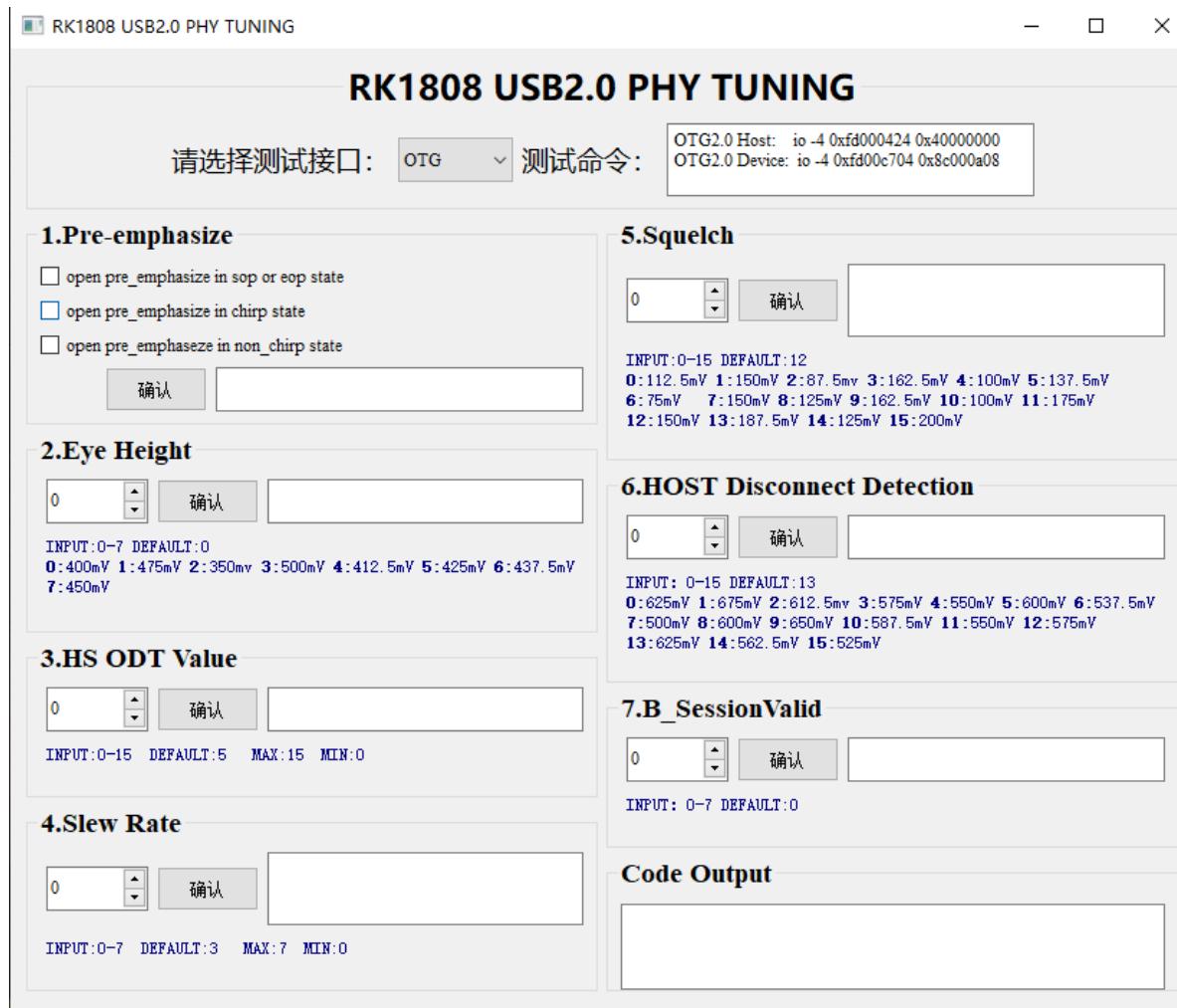


图2 RK1808 USB2.0 PHY Tuning Interface

## 2.3 RK3399 USB2.0 Tuning 界面说明

RK3399 USB2.0 支持 TYPE-C0、TYPE-C1、HOST0、HOST1 共四种类型接口，RK3399 界面的 Tuning 可以分为以下几个步骤：

1. 在红色方框 1 左侧选择所需要测试的接口，红色方框1右侧的测试命令框会自动生成对应的 SQ 测试命令。
2. 在红色方框 2 中选择对应的参数，Pre-emphasize 参数选择与其他 Tuning 项不同，通过打钩的方式选择预加重的参数，其他的 Tuning 项是通过类似红色方框4中的控件获取参数，手动输入或者点击上下按键均可改变参数值，参数输入范围以及参数的描述在每个 Tuning 项中均有提供。
3. 点击红色方框 3 中的“确认”按键，按键右侧的框中会生成对应的 io 命令，并且红色方框 5 中的 Code Output 部分会生成对应的参考代码。

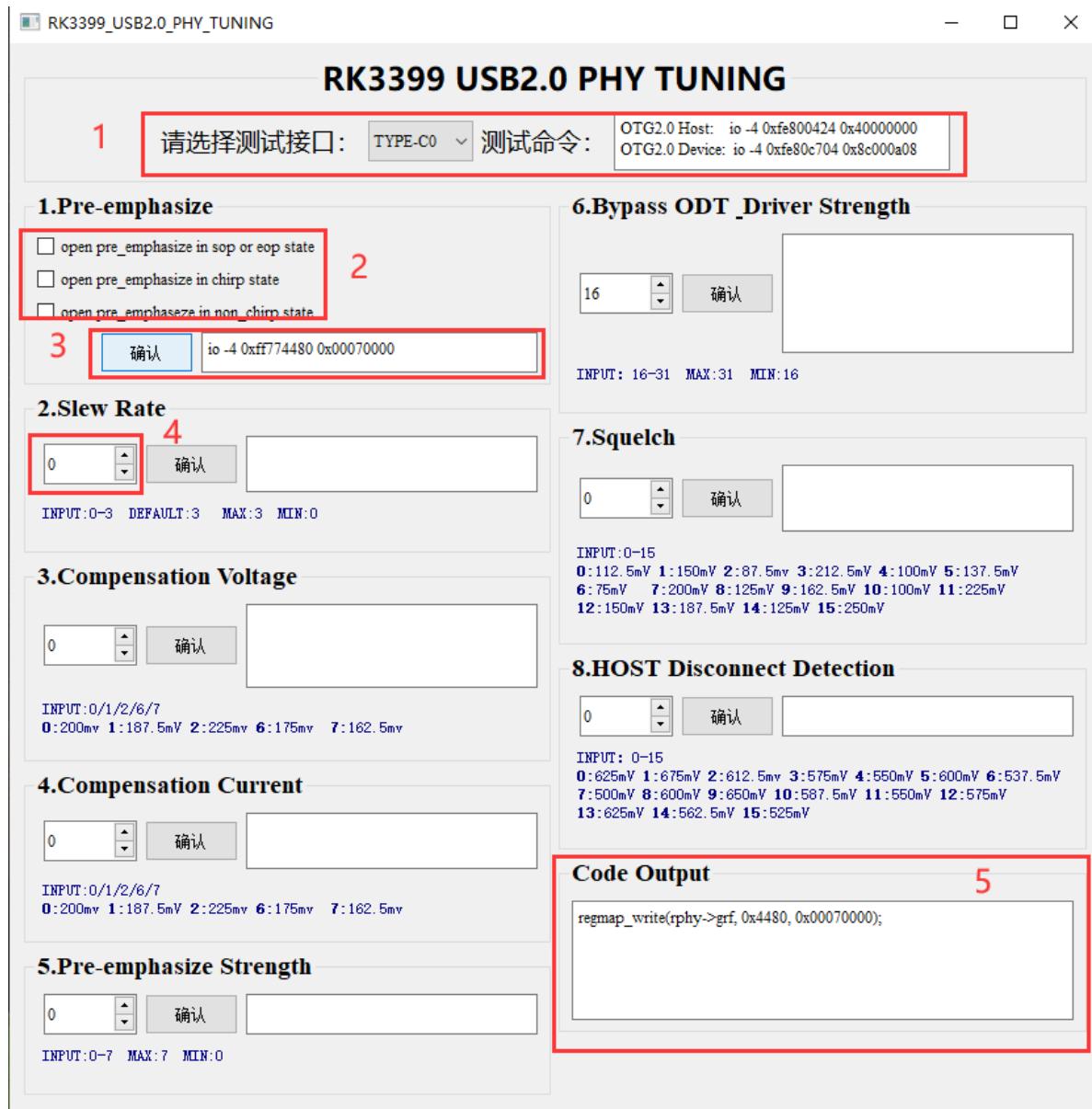


图3 RK3399 USB2.0 PHY Tuning Interface

### RK3399 USB2.0信号调整参数介绍

表3 RK3399 USB2.0信号调整参数

调整参数	描述
Pre-emphasize	调整 HS Tx 预加重
Slew Rate	调整 HS Tx 眼图的 slew rate
Compensation Voltage	调整 HS Tx 电压校准点，调高校准点可以提高 USB 眼图的高度
Compensation Current	调整 HS Tx 电流校准点，调高校准点可以提高 USB 眼图的高度
Pre-emphasize Strength	调整 HS Tx 预加重强度
Bypass ODT & Driver Strength	Bypass comp 电路中的电阻自动调整电路，可以调整 USB 眼图的高度
Squelch	调整 HS Rx 的噪声阈值
HOST Disconnect Detection	调整 HS 断开检测阈值，只用于 HOST

## 2.4 RK3568 USB2.0 Tuning 界面说明

RK3568 USB2.0 支持 OTG、HOST1、HOST2、HOST3 共四种类型接口，RK3568 界面的 Tuning 可以分为以下几个步骤：

1. 在红色方框 1 左侧选择所需要测试的接口，红色方框 1 右侧的测试命令框会自动生成对应的 SQ 测试命令。
2. 由于 RK3568 平台的 USB PHY 寄存器没有高 16 位的“write\_enable bit”，所以为了防止被调整的寄存器中其他不相关或未调整的某些 bit 值被修改，需要用户使用红色方框 2 中提供的读取寄存器命令获取对应寄存器的初始值，填入方框 2 右侧的“输入寄存器初始值”框中。
3. 在红色方框 3 中选择对应的参数，Pre-emphasize 参数选择与其他 Tuning 项不同，通过打勾的方式选择预加重的参数，其他的 Tuning 项是通过类似红色方框 5 中的控件获取参数，手动输入或者点击上下按键均可改变参数值，参数输入范围以及参数的描述在每个 Tuning 项中均有提供。
4. 点击红色方框 4 中的“确认”按键，按键右侧的框中会生成对应的 io 命令，并且红色方框 6 中的 Code Output 部分会生成对应的参考代码。

**注意：**寄存器初始值的框中已经有预设的默认值，如果没有手动输入寄存器值，将会使用默认值，可能会影响最终 Tuning 结果，故测试前务必手动读取当前寄存器值后填入寄存器初始值的框中。

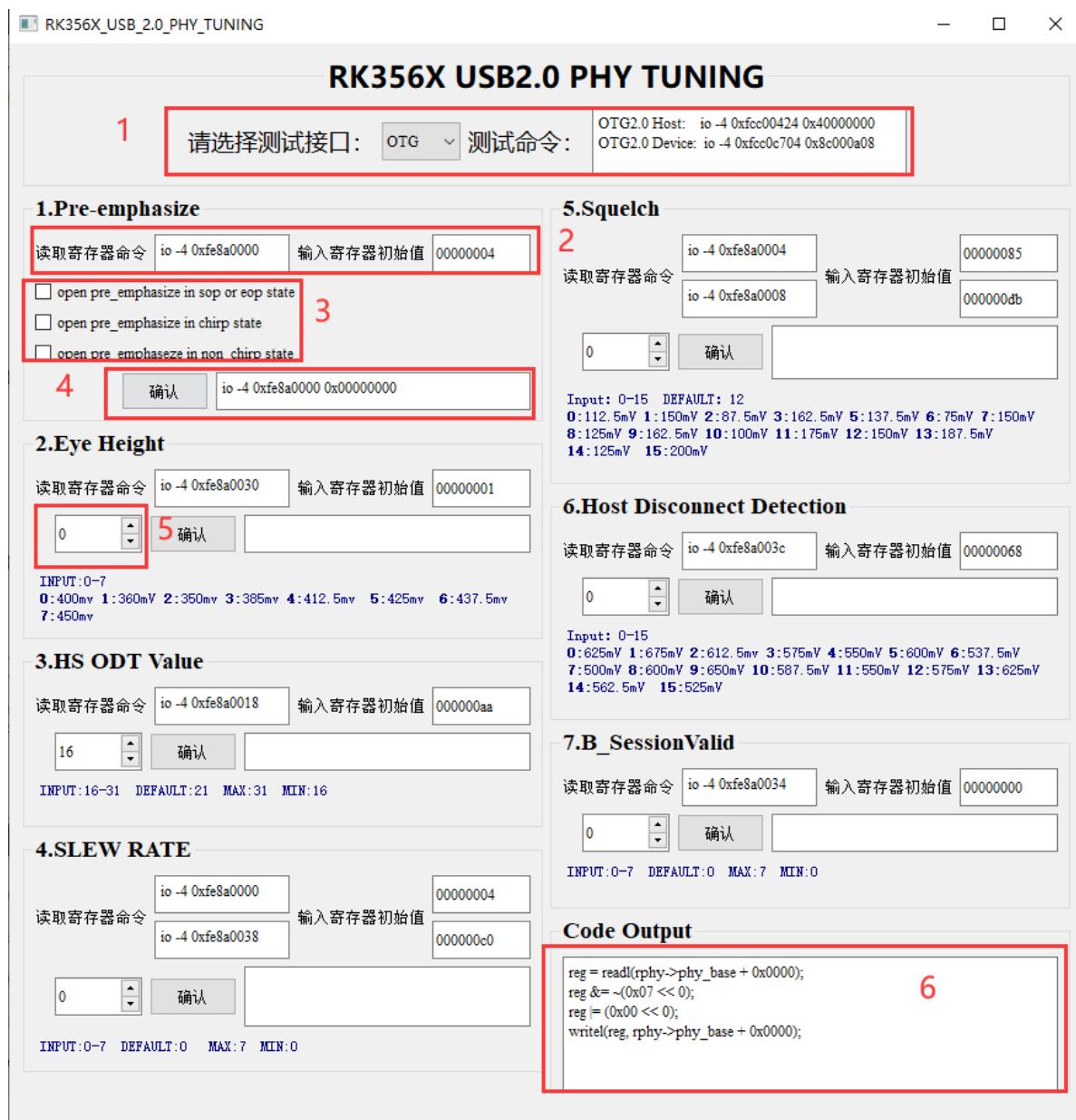


图4 RK3568 USB2.0 PHY Tuning Interface

### RK3568 USB2.0 信号调整参数介绍

RK3568 界面共提供了四类参数调整，包括 USB2.0 SQ 调整、USB2.0 噪声阈值调整、USB2.0 断开检测阈值调整、USB2.0 B\_Sessionvalid 调整。

#### RK3568 USB2.0 SQ 调整

USB2.0 SQ 调整主要调整 USB2.0 信号的预加重、眼图高度、Slew Rate、驱动强度等。

表4 RK3568 USB2.0 SQ 信号调整参数

```

reg = readl(rphy->phy_base + 0x0000);
reg &= ~(0x07 << 0);
reg |= (0x00 << 0);
	writel(reg, rphy->phy_base + 0x0000);

```

6

调整参数	描述
Pre-emphasize	调整 HS Tx 预加重
HS Eye Height	调整 HS Tx 眼图的高度
HS ODT Value	调整 ODT $45\Omega$ 电阻值，输入值越大，ODT 电阻越小，眼图的高度就越大，但同时会加快 FS/LS 的 slew rate
Slew Rate	调整 HS 眼图的 slew rate，但调整作用很小
Squelch	调整 HS Rx 的噪声阈值
HOST Disconnect Detection	调整断开检测阈值，只用于 HOST
B_Session Valid	调整 VBUS DET 阈值，只用于 Device

### RK3568 USB2.0 噪声阈值调整

当使用长度较长、质量较差、阻抗较大的 USB 线缆连接 High-speed 外设，无法被正常枚举。此时可以尝试调整 USB 噪声阈值。USB PHY 的噪声阈值一般默认为 150 mV，当使用阻抗较大的 USB 线缆时，USB 正常信号的幅值会衰减得很厉害，甚至低于 150 mV 的，因此可能正常信号会被当作噪声处理了。此时可以适当的降低 USB2.0 PHY 的噪声阈值，一般可以调整为 125 mV 或者 112.5mV，此项可以通过 Squelch Tuning 改善。

### RK3568 USB2.0 断开检测阈值调整

断开检测阈值调整可以通过 Host Disconnect Detection Tuning 改善，Rockchip 平台 USB Host 端口识别 USB 外设日志如下：

```
[ 1204.092638] usb 1 - 1: new high - speed USB device number 3 using ehci - platform
[ 1204.280373] usb 1 - 1: New USB device found, idVendor=058f, idProduct=6387,
bcdDevice= 1.04
[ 1204.280449] usb 1 - 1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 1204.280476] usb 1 - 1: Product: Mass Storage
[ 1204.280498] usb 1 - 1: SerialNumber: FA45A19C
```

Rockchip 平台 USB Host 端口断开 USB 外设连接日志如下：

```
[ 985.341233] usb 1 - 1: USB disconnect, device number 2
```

#### 1) 无法检测 USB2.0 外设断开

当 USB Host 拔掉 USB2.0 外设，无法检测到断开（拔掉 USB 外设时不会打印以上的断开连接的日志），此时可以尝试如下步骤优化：

1. 开启 EOP 的预加重（SDK 默认关闭 EOP 的预加重），开启 EOP 预加重位置如图 5 所示。



图5 开启 EOP 预加重

2. 减小 USB Host 的断开检测阈值；

注意：

- 有些 Rockchip 平台主控芯片没有调节EOP预加重的操作，可忽略 Step 1步骤，具体以工具页面为准；
- 只有 Step1 操作无明显改善效果的情况下，才需要在 Step1 的基础上再做 Step2 的操作；
- Step2 减小断开检测阈值存在一定的风险，如果调整过度会导致 USB 外设在正常使用中发生自动断开情况，所以用户需要自己把控风险；

## 2) USB2.0 外设异常断开

当 USB Host 连接 USB2.0 外设在正常工作中发生异常断开，我们可以尝试增大 USB Host 的断开检测阈值。

注意：

- 并不是所有的异常断开都是因为断开检测阈值的原因，往往更多的情况是 USB 外设自身问题或者硬件环境问题导致的，所以不应该一出现此类问题就盲目的去调整断开检测阈值。
- Rockchip 的 SDK 默认是关闭 EOP 的预加重，所以此时不需要配置寄存器再去关闭该操作；
- 增大断开检测阈值同样存在一定的风险，如果调整过度会导致 Host 无法检测到 USB 外设断开连接的情况，所以用户需要自己把控风险；

## RK356X USB2.0 B\_Sessionvalid 调整

B\_Sessionvalid 调整主要用于 PC 无法识别 USB Device (如：ADB/MTP)或者识别到 USB Device 后又断开的现象。此时可以尝试调整此项，降低电压判决阈值。默认值为 3'b000，建议改为 3'b111（参数选择7）或 3'b101（参数选择5）。导致 USB Device 连接失败的硬件问题主要有 3 种：

- Vbus 电压太低 (一般低于 4.7V 容易出现问题)。
- USB\_AVDD1V0 纹波太高，或者 USB\_AVDD1V0 被抬高到 1.2V，导致 PHY 检测不到 Bvalid 有效信号。
- Logic 电压纹波太高，导致 USB 控制器工作异常。

## 2.5 RK3588 USB2.0 Tuning 界面说明

RK3588 USB2.0 支持 TYPE-C0、TYPE-C1、USB2.0 HOST0、USB2.0 HOST1 共四种类型接口，RK3588 USB2.0 界面的 Tuning 可以分为以下几个步骤：

1. 在红色方框 1 左侧选择所需要测试的接口，红色方框 1 右侧的测试命令框会自动生成对应的 SQ 测试命令。

- 在红色方框 2 中选择对应的参数，通过手动输入或者点击上下按键均可改变参数值，参数输入范围以及参数的描述在每个 Tuning 项中均有提供。
- 点击红色方框 2 中的“确认”按键，按键右侧的框中会生成对应的 io 命令，并且红色方框 3 中的 Code Output 部分会生成对应的参考代码。

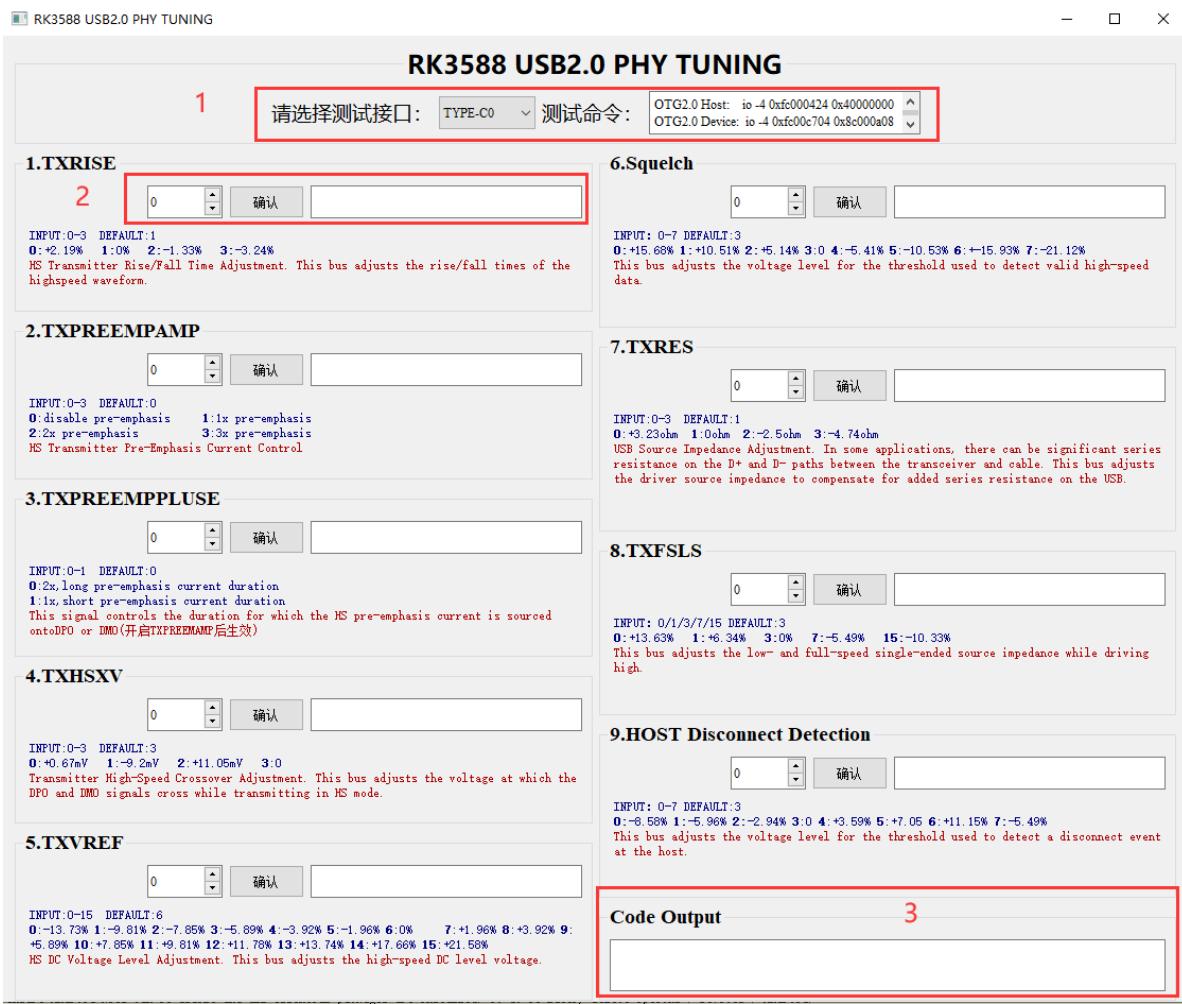


图6 RK3588 USB2.0 PHY Tuning Interface

## RK3588 USB2.0 信号调整参数介绍

表5 RK3588 USB2.0 信号调整参数

调整参数	描述
TXRISE	调整 HS Tx slew rate， 可改变上升下降时间
TXPREEMPAMP	调整 HS Tx 预加重电流
TXPREEMPPLUSE	调整 HS Tx 预加重电流占空比， 此项必须在开启 <b>TXPREEMPAMP TUNE</b> 后才生效
TXHSXV	调整 HS Tx DP/DM 交叉点电压， 此项调整无明显变化
TXVREF	调整 HS Tx DC 电压， 可以明显改变眼图幅值
TXRES	调整 HS Tx 阻抗， 可改变眼图幅值/上升下降时间
TXFSLS	调整 FS/LS Tx 阻抗， 可改变眼图幅值/上升下降时间/交叉点电压/占空比
HOST DISCONNECT DETECTION	调整断开检测阈值， 只用于 HOST

## 2.6 RV1106 USB2.0 Tuning 界面说明

RV1106 USB2.0 Tuning 界面如下图所示，支持预加重、预加重强度、ODT 调整、眼高调整、噪声阈值调整、Bvalid 阈值调整、断开阈值调整。RV1103 因为芯片封装少了 VBUS DET 脚，所以，RV1103 不支持 Bvalid 阈值调整，其它调整项与 RV1106 相同。

### RV1106 USB2.0 PHY TUNING

请选择测试接口: **OTG** 测试命令: **OTG2.0 Host: io -4 0xffb00424 0x40000000**

**1.Pre-emphasize**

读取寄存器命令 **io -4 0xff3e0030** 输入寄存器初始值 **0000000f**

open pre\_emphasize in sop or eop state  
 open pre\_emphasize in chirp state  
 open pre\_emphasize in non\_chirp state

**确认**

**5.Squelch**

读取寄存器命令 **io -4 0xff3e0064** 输入寄存器初始值 **000000e0**

**0** **确认**

**Input: 0~15 DEFAULT: 12  
0:112.5mV 2:87.5mV 4:100mV 9:162.5mV 11:175mV 12:150mV 14:125mV  
15:200mV**

**2.Pre-emphasize Strength**

读取寄存器命令 **io -4 0xff3e0040** 输入寄存器初始值 **00000051**

**0** **确认**

**INPUT:0~3 DEFAULT:2 MAX:3 MIN:0**

**6.BValid**

读取寄存器命令 **io -4 0xff3e010c** 输入寄存器初始值 **00000000**

**0** **确认**

**INPUT:0~3 DEFAULT:0  
0:2.52V 1:2.30V 2:2.15V 3:2.70V**

**3.HS ODT Value**

读取寄存器命令 **io -4 0xff3e011c** 输入寄存器初始值 **000000b7**

**16** **确认**

**INPUT:16~31 DEFAULT:23 MAX:31 MIN:16**

**7.HOST Disconnect Detection**

读取寄存器命令 **io -4 0xff3e0060** 输入寄存器初始值 **00000003**

读取寄存器命令 **io -4 0xff3e0068** 输入寄存器初始值 **000000e8**

读取寄存器命令 **io -4 0xff3e0064** 输入寄存器初始值 **000000e0**

**0** **确认**

**INPUT: 0~15 DEFAULT:13  
0:625mV 1:675mV 2:612.5mV 3:575mV 4:550mV 5:600mV 6:637.5mV 7:500mV  
8:600mV 9:650mV 10:587.5mV 11:550mV 12:575mV 13:625mV 14:562.5mV  
15:525mV**

**Code Output**

图7 RV1106 USB2.0 PHY Tuning Interface

表6 RV1106 USB2.0 信号调整参数

调整参数	描述
Pre-emphasize	调整 HS Tx 预加重， 默认配置已开启
Pre-emphasize Strength	调整 HS Tx 预加重强度， 该寄存器只有在预加重开启后， 才能生效
HS ODT Value	调整 ODT 45Ω 电阻值， 输入值越大， ODT 电阻越小， 眼图的高度就越大。 并且， 不会影响 FS/LS 的 slew rate。
Eye Height	调整 HS Tx 眼图的高度。其中， 档位 4: 470mV 必须同时保证芯片的 logic 电压 > 850mV
Squelch	调整 HS Rx 的噪声阈值
BValid	调整 VBUS DET 阈值， 只用于 Device
HOST Disconnect Detection	调整断开检测阈值， 只用于 HOST

## 2.7 RV1109 USB2.0 Tuning 界面说明

RV1109 USB2.0 Tuning 界面如下图所示，支持 Slew Rate、眼高调整、噪声阈值调整、断开阈值调整。RV1126 调整项与 RV1109 相同。

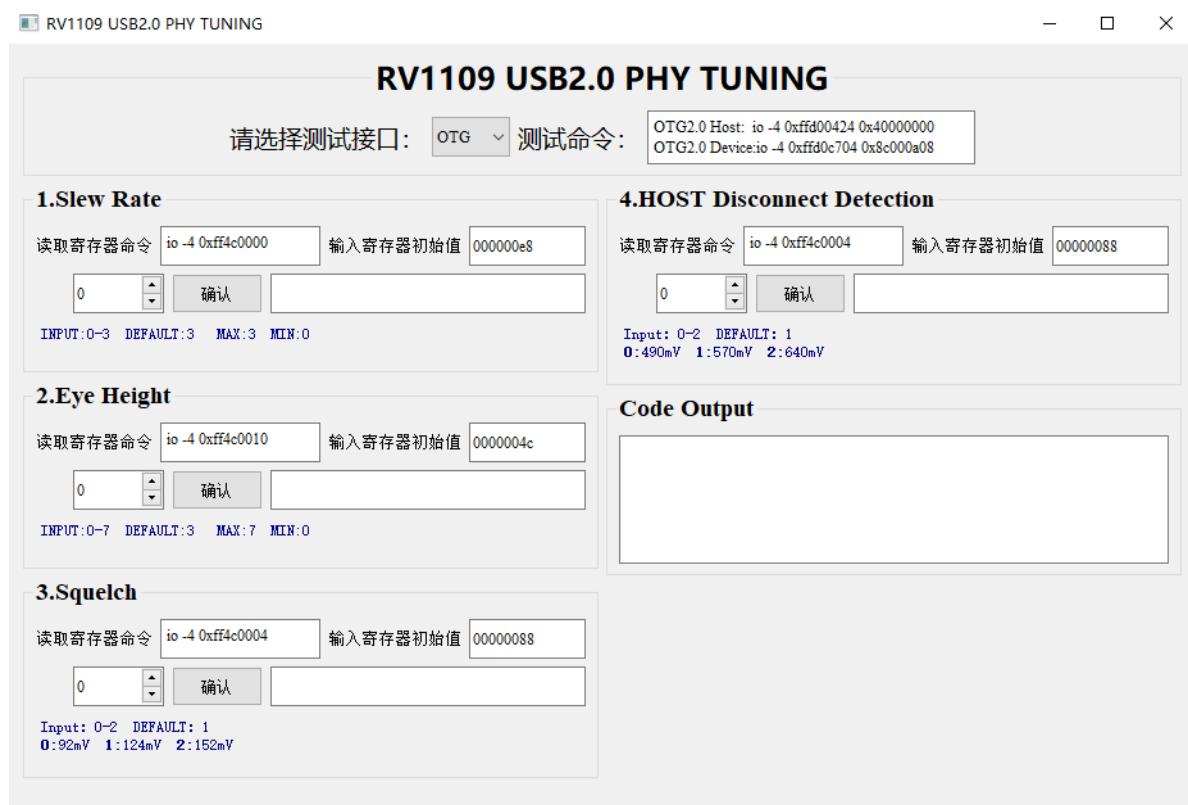


图8 RV1109 USB2.0 PHY Tuning Interface

表7 RV1109 USB2.0 信号调整参数

调整参数	描述
Slew Rate	调整 HS Tx 眼图的 slew rate
Eye Height	调整 HS Tx 眼图的高度。
Squelch	调整 HS Rx 的噪声阈值
HOST Disconnect Detection	调整断开检测阈值，只用于 HOST

## 2.8 RK1808 USB3.1 Tuning 界面说明

RK1808 USB3.1 Tuning 界面如下图所示，支持 Tx driver voltage、Tx driver swing、Tx driver pre-emphasis strength。



图9 RK1808 USB3.1 PHY Tuning Interface

表8 RK1808 USB3.1 信号调整参数

调整参数	描述
TX driver voltage	TX driver output common mode voltage tuning, tune the common voltage of TXP/TXN differential signals
TX driver swing	TX driver swing tuing
TX driver pre-emphasis strength	TX driver pre-emphasis strength tuning

## 2.9 RK3399 USB3.1 Tuning 界面说明

RK3399 USB3.1 支持 OTG0、OTG1 两种接口类型，RK3399 USB3.1 界面的 Tuning 可以分为以下几个步骤：

1. 在红色方框 1 左侧选择需要测试的接口，红色方框 1 右侧的测试命令框会自动生成对应的 SQ 测试命令。
2. RK3399 USB3.1 PHY 支持 Type-C 正面和反面检测，所以需按照红色方框 2 所示，先输入读取寄存器的命令以获取正反面信息，再将读取的寄存器值填入“请输入寄存器值”窗口中，否则会提示异常。注意该值需要在 Type-C 线正常连接的情况下读取。
3. 点击红色方框 3 中的“start”按键，此时方框 1 中的“测试接口”会无法选定，同时方框 5 中的“确认”按键变成可选定状态。
4. 由于 RK3399 平台的 USB 3.1 PHY 寄存器没有高 16 位的“write\_enable bit”，所以为了防止被调整的寄存器中其他不相关或未调整的某些 bit 值被修改，需要用户使用红色方框 4 中提供的读取寄存器命令获取对应寄存器的初始值，填入方框 4 右侧的“输入寄存器初始值”框中。
5. 在红色方框 5 中选择对应的参数，点击“确认”按键，按键右侧的框中会生成对应的 io 命令，并且红色方框 6 中的“Code Output”部分会生成对应的参考代码。



图10 RK3399 USB3.1 PHY Tuning Interface

#### RK3399 USB3.1 信号调整参数介绍

表9 RK3399 USB3.1 信号调整参数

调整参数	描述
BOOST LEVEL	TX boost level adjust signal

#### 2.10 RK3528 USB3.1 Tuning 界面说明

RK3528 USB3.1 Tuning 界面如下图所示，支持 Full Txswing and Txmargin、Low Swing and Txmargin、Tx De-emphasis、Boost Level。RK3562、RK3568 调整项与RK3528相同。



图11 RK3528 USB3.1 PHY Tuning Interface

表10 RK3528 USB3.1 信号调整参数

调整参数	描述
Full Txswing and TXmargin	Full swing and Txmargin Tuning. TX_SWING and TX_MARGIN[2:0] are combined together to control TX output amplitude.
Low swing and Txmargin	Low swing and Txmargin Tuning (与 Full Tx swing 作用一样，只是档位不同)
TX De-emphasis	Transmitter de-emphasis level configuration.
Boost Level	TX boost level adjust signal

## 2.11 RK3588 USB3.1 Tuning 界面说明

RK3588 USB3.1 支持 USB3-OTG0（USB3&Dp CombPhy0）、USB3-OTG1（USB3&Dp CombPhy1）、USB3-HOST2（USB3&PCIE&SATA CombPhy）共三种类型接口的 Tuning，RK3588 USB3.1 界面的 Tuning 可以分为以下几个步骤：

1. 在红色方框 1 左侧选择所需要测试的接口，红色方框 1 右侧的测试命令框会自动生成对应的 SQ 测试命令。
2. 如果是测试 USB3-OTG0/1，因为 CombPhy0/1 支持正面和反面，所以需按照红色方框2所示，先输入读取寄存器的命令以获取 CombPhy0/1 的正反面信息，再将读取的寄存器值填入“输入寄存器值”窗口中，该寄存器的值必须是 0xCx 或者 0x3x，否则窗口会提示异常。如果读取寄存器值失败（通常发生在 Type-C 接口类型），请先确认 USB 线已经插入待调试的 USB3-OTG0/1 接口，目的是让 CombPhy0/1 进入工作状态，再输入读取寄存器的命令。
3. 如果是测试 USB3-HOST 2 口，请跳过红色方框 2 的步骤，直接点击红色方框 3 中的 START 按键即可。

- 点击 START 按键后，如果是测试 USB3-OTG0/1 口，则界面左侧的 USB3-OTG0/1 相关 Tuning 项的“确认”按键会生效，如红色方框 5 所示，而右侧 USB3-HOST2 相关 Tuning 项的“确认”按键会失效，如红色方框 6 所示。
- 由于 RK3588 平台的 USB 3.1 PHY 寄存器没有高 16 位的“write\_enable bit”，所以为了防止被调整的寄存器中其他不相关或未调整的某些 bit 值被修改，需要用户使用红色方框 4 中提供的读取寄存器命令获取对应寄存器的初始值，填入方框 4 右侧的“输入寄存器初始值”框中。
- 在红色方框 5 中选择对应的参数，点击“确认”按键，按键右侧的框中会生成对应的 io 命令，并且红色方框 7 中的 Code Output 部分会生成对应的参考代码。

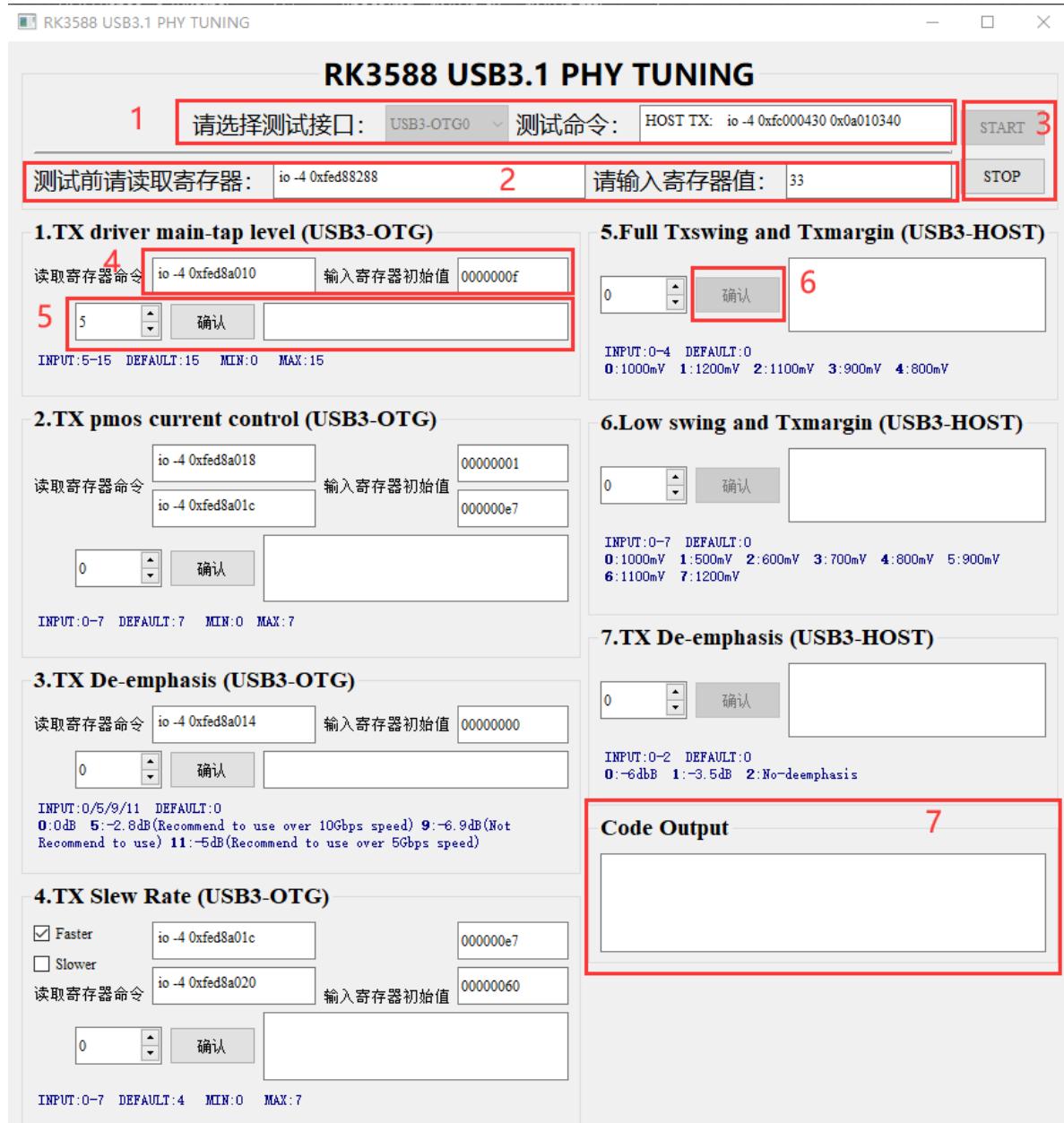


图12 RK3588 USB3.0 PHY Tuning Interface

### RK3588 USB3.1信号调整参数介绍

表11 RK3588 USB3.1 信号调整参数

调整参数	描述
TX Driver main-tap level(USB3-OTG)	调整 SS Tx 电压
TX Pmos current control(USB3-OTG)	调整 SS Tx 电流，建议在调压不满足时再设置
TX De-emphasis(USB3-OTG)	调整 SS Tx 去加重
TX Slew Rate(USB3-OTG)	调整 SS Tx Slew Rate
Full Txswing and TXmargin(USB3-HOST)	Full swing and Txmargin Tuning. TX_SWING and TX_MARGIN[2:0] are combined together to control TX output amplitude.
Low swing and Txmargin(USB3-HOST)	Low swing and Txmargin Tuning (与 Full Tx swing 作用一样，只是档位不同)
TX De-emphasis(USB3-HOST)	Transmitter de-emphasis level configuration.

### 3. Rockchip USB PHY Tuning 代码生成说明

#### 3.1 RK1808 USB2.0 PHY

RK1808 平台 USB2.0 PHY Tuning 的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的 rk1808\_usb2phy\_tuning() 函数中。

```
static int rk1808_usb2phy_tuning(struct rockchip_usb2phy *rphy)
{
    .../* OTG/Host PHY Tuning Code */
}
```

#### 3.2 RK2118 USB2.0 PHY

RK2118 平台 USB2.0 PHY Tuning 的代码添加在 bsp/rockchip/rk2118/board/evb/board.c 的 usb\_phy\_init() 函数中。

```
void usb_phy_init(void)
{
#ifndef RT_USING_USB
    ...
#else
    .../* OTG PHY Tuning Code */
#endif
}
```

### 3.3 RK3036/RK312X USB2.0 PHY

RK3036 平台目前只支持 kernel3.10 版本，代码添加在 kernel/drivers/usb/dwc\_otg\_310/usbdev\_rk3036.c 的 usb20otg\_hw\_init() 函数中。

```
static void usb20otg_hw_init(void) {
    .../* OTG/Host PHY Tuning Code */
}
```

RK312X 平台的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的 rk312x\_usb2phy\_tuning() 函数中。

```
static int rk312x_usb2phy_tuning(struct rockchip_usb2phy *rphy)
{
    .../* OTG/Host PHY Tuning Code */
}
```

### 3.4 RK3228 USB2.0 PHY

RK3228 平台的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的 rk3228\_usb2phy\_tuning() 函数中。

```
static int rk3228_usb2phy_tuning(struct rockchip_usb2phy *rphy)
{
    .../* OTG/Host PHY Tuning Code */
}
```

### 3.5 RK3288 USB2.0 PHY

由于 RK3288 的 USB PHY 驱动代码具有两个不同的版本，对应的 USB PHY 修改的代码也不相同，所以在 RK3288 页面中生成代码的时候需要先选择下 kernel 版本（使用 kernel-4.4 及以下版本的 Kernel Version 都选择 4.4，而 kernel-4.19 及以上版本的 Kernel Version 都选择 4.19，选择不同的 kernel 版本仅仅只是生成的代码格式不一样而已，对生成的 io 命令等不会有影响）。

#### RK3288 kernel-4.19 版本

代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-usb.c 的 rk3288\_usb\_phy\_probe\_init() 函数的末尾。由于 RK3288 有 3 个 USB 端口，所以 io 命令转化后的代码段前面需要加上 USB 端口判断来区分不同的 USB 端口。

OTG 接口代码添加如下：

```
if (rk_phy->reg_offset == 0x320) {
    ... /* OTG PHY Tuning Code */
}
```

HOST1 接口代码添加如下：

```
if (rk_phy ->reg_offset == 0x334) {  
    ... /* HOST1 PHY Tuning Code */  
}
```

HOST2 接口代码添加如下：

```
if (rk_phy ->reg_offset == 0x348) {  
    ... /* HOST2 PHY Tuning Code */  
}
```

### RK3288 kernel-4.4 版本

驱动代码路径：kernel/drivers/usb/dwc\_otg\_310/usbdev\_rk32.c

RK3288 有 3 个 USB 端口，其中 OTG 口代码添加在 usb20otg\_hw\_init() 函数末尾，HOST1 口代码添加在 rk\_ehci\_hw\_init() 函数末尾，HOST2 口代码添加在 usb20host\_hw\_init() 函数末尾。

## 3.6 RK3308B\_S/RK3308 USB2.0 PHY

RK3308B\_S 平台的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的 rk3308\_usb2phy\_tuning() 函数中，在 soc\_is\_rk3308bs() 判断条件末尾。

```
if (soc_is_rk3308bs ()) {  
    ... /* OTG/HOST PHY Tuning Code */  
}
```

RK3308 平台的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的 rk3308\_usb2phy\_tuning() 函数中的 else 判断条件末尾。

```
if (soc_is_rk3308bs ()) {  
    ...  
} else {  
    .../* OTG/HOST PHY Tuning Code */  
}
```

## 3.7 RK3326S/PX30S/RK3326/PX30/RK3328 USB2.0 PHY

RK3326S/PX30S 平台的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的 rk3328\_usb2phy\_tuning() 函数中，在 soc\_is\_px30s() 判断条件的末尾。

```
if (soc_is_px30s ()) {  
    .../* OTG/HOST PHY Tuning Code */  
}
```

RK3326/PX30/RK3328 平台的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的 rk3328\_usb2phy\_tuning() 函数中的 else 判断条件末尾。

```
if (soc_is_px30s()) {  
    ...  
} else {  
    .../* OTG/HOST PHY Tuning Code */  
}
```

### 3.8 RK3399 USB2.0 PHY

RK3399 平台 USB 2.0 PHY Tuning 的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的 rk3399\_usb2phy\_tuning() 函数中的 “if (!of\_property\_read\_bool(node, "rockchip,u2phy-tuning"))” 之前。由于 RK3399 有 4 个 USB 端口，其中 TYPE-C0 口和 HOST0 属于一组 PHY (区分地址: 0xe450)，TYPE-C1 和 HOST1 属于另一组 PHY (区分地址: 0xe460)。所以代码段前面需要加上寄存器的地址判断来区分不同的 USB 端口。

```
if (rphy->phy_cfg->reg == 0xe450) {  
    .../* TYPE - C0/HOST0 PHY Tuning Code */  
} else {  
    ... /* TYPE - C1/HOST1 PHY Tuning Code */  
}  
if (!of_property_read_bool(node, "rockchip,u2phy-tuning"))  
    return ret;
```

### 3.9 RK3528 USB2.0 PHY

RK3528 平台 USB2.0 PHY Tuning 的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 中 rk3528\_usb2phy\_tuning() 函数中。

```
static int rk3528_usb2phy_tuning(struct rockchip_usb2phy *rphy)  
{  
    .../* USB2.0 PHY Tuning Code */  
}
```

### 3.10 RK3562 USB2.0 PHY

RK3562 平台 USB2.0 PHY Tuning 的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 中 rk3562\_usb2phy\_tuning() 函数中。

```
static int rk3562_usb2phy_tuning(struct rockchip_usb2phy *rphy)  
{  
    .../* USB2.0 PHY Tuning Code */  
}
```

### 3.11 RK3568/RK3566 USB2.0 PHY

RK3568/RK3566 平台 USB2.0 PHY Tuning 的代码是共用的，添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 中 rk3568\_usb2phy\_tuning() 函数末尾。由于 RK3568 有 4 个 USB 端口，其中 OTG 口和 HOST1 属于一组 PHY (基地址 0xfe8a0000)，HOST2 和 HOST3 属于一组 PHY (基地址 0xfe8b0000)。所以生成的代码段前面需要加上寄存器地址进行判断区分不同的 USB 端口。OTG 接口和 HOST1 接口代码添加如下：

```
if (rphy->phy_cfg->reg == 0xfe8a0000) {
    ... /* OTG/Host1 PHY Tuning Code */
}
```

HOST2 口和 HOST3 口代码添加如下：

```
if (rphy->phy_cfg->reg == 0xfe8b0000) {
    ... /* Host2/Host3 PHY Tuning Code */
}
```

### 3.12 RK3576 USB2.0 PHY

RK3576 平台 USB2.0 PHY Tuning 的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 文件的 rk3576\_usb2phy\_tuning() 函数中。由于 RK3576 有 OTG0、OTG1 共 2 个 USB 端口，2 个端口的区分根据 rphy->phy\_cfg->reg 的值来判断，代码添加位置如下：

```
static int rk3576_usb2phy_tuning(struct rockchip_usb2phy *rphy)
{
    unsigned int reg;
    int ret = 0;

    if (rphy->phy_cfg->reg == 0x0) {
        .../* OTG0 PHY Tuning Code */
    } else if (rphy->phy_cfg->reg == 0x2000) {
        .../* OTG1 PHY Tuning Code */
    }
    return ret;
}
```

### 3.13 RK3588 USB2.0 PHY

RK3588 平台 USB2.0 PHY Tuning 的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 文件的 rk3588\_usb2phy\_tuning() 函数中。由于 RK3588 有 TYPE-C0、TYPE-C1、HOST0、HOST1 共 4 个 USB 端口，4 个端口的区分根据 rphy->phy\_cfg->reg 的值来判断，代码添加位置如下：

```

if (rphy->phy_cfg->reg == 0x0000) {
    .../* TYPE-C0 PHY Tuning Code */
} else if (rphy->phy_cfg->reg == 0x4000) {
    .../* TYPE-C1 PHY Tuning Code */
} else if (rphy->phy_cfg->reg == 0x8000) {
    .../* HOST0 PHY Tuning Code */
} else if (rphy->phy_cfg->reg == 0xc000) {
    .../* HOST1 PHY Tuning Code */
}

```

### 3.14 RV1103/RV1106 USB2.0 PHY

RV1103/RV1106 平台的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的 rv1106\_usb2phy\_tuning() 函数中。

```

static int rv1106_usb2phy_tuning(struct rockchip_usb2phy *rphy)
{
    .../* OTG PHY Tuning Code */
}

```

### 3.15 RV1109/RV1126 USB2.0 PHY

RV1109/RV1126 平台的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-naneng-usb2.c 的 rv1126\_usb2phy\_tuning() 函数中。

```

static int rv1126_usb2phy_tuning(struct rockchip_usb2phy *rphy)
{
    if (rphy->phy_cfg->reg == 0xff4c0000) {
        .../* OTG PHY Tuning Code */
    }
    if (rphy->phy_cfg->reg == 0xff4c8000) {
        .../* HOST PHY Tuning Code */
    }
}

```

### 3.16 RK1808 USB3.1 PHY

RK1808 平台 USB3.1 PHY Tuning 的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-combphy.c 的 rk1808\_combphy\_cfg 函数中。

```

static int rk1808_combphy_cfg(struct rockchip_combphy_priv *priv)
{
    ...
    if (priv->phy_type == PHY_TYPE_PCIE) {
        ...
    } else if (priv->phy_type == PHY_TYPE_USB3) {
        ...
        /*
         * Tuning Tx:
         */
    }
}

```

```

        * offset 0x21b8 bit[7:4]: lane 0 TX driver swing
        * tuning bits with weight, "1111" represents the
        * largest swing and "0000" the smallest.
    */
    reg = readl(priv->mmio + 0x21b8);
    reg = (reg & ~0xf0) | 0xe0;
    writel(reg, priv->mmio + 0x21b8);

    .../* USB3.1 PHY Tuning Code */

    ...
} else {
    dev_err(priv->dev, "failed to cfg incompatible PHY type\n");
    return -EINVAL;
}
return 0;
}

```

### 3.17 RK3399 USB3.1 PHY

RK3399 平台 USB3.1 PHY Tuning 的代码添加在 kernel/drivers/phy/phy-rockchip-typec.c 的 tcpphy\_tx\_usb3\_cfg\_lane() 函数中。

```

static void tcpphy_tx_usb3_cfg_lane(struct rockchip_typec_phy *tcpphy, u32 lane)
{
    .../* OTG0/OTG1 PHY Tuning Code */
}

```

### 3.18 RK3528/RK3562/RK3566/RK3568 USB3.1 PHY

RK3528、RK3562、RK3568 平台 USB3.1 PHY Tuning 的代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-naneng-combphy.c 文件对应的 combphy\_cfg 函数中。

RK3528 平台代码添加在 rk3528\_combphy\_cfg 函数的如下位置。

```

case PHY_TYPE_USB3:
    ...
    /* Enable adaptive CTLE for USB3.0 Rx */
    val = readl(priv->mmio + 0x200);
    val &= ~GENMASK(17, 17);
    val |= 0x01;
    writel(val, priv->mmio + 0x200);
    ...
    .../* USB3.1 PHY Tuning Code */
    ...
    param_write(priv->phy_grf, &cfg->pipe_txcomp_sel, false);
    param_write(priv->phy_grf, &cfg->pipe_txelec_sel, false);
    param_write(priv->phy_grf, &cfg->usb_mode_set, true);
    break;
}

```

RK3562 平台代码添加在 rk3562\_combphy\_cfg 函数的如下位置。

```

case PHY_TYPE_USB3:
    /* Set SSC downward spread spectrum */
    val = readl(priv->mmio + (0x1f << 2));
    val &= ~GENMASK(5, 4);
    val |= 0x01 << 4;
    writel(val, priv->mmio + 0x7c);
    ...
    .../* USB3.1 PHY Tuning Code */
    ...
    param_write(priv->phy_grf, &cfg->pipe_sel_usb, true);
    param_write(priv->phy_grf, &cfg->pipe_txcomp_sel, false);
    param_write(priv->phy_grf, &cfg->pipe_txelec_sel, false);
    param_write(priv->phy_grf, &cfg->usb_mode_set, true);
    break;

```

RK3566 与 RK3568 平台的代码是共用的，代码添加在 rk3568\_combphy\_cfg 函数的如下位置。

```

case PHY_TYPE_USB3:
    ...
    /* Set PLL KVCO to min and set PLL charge pump current to max */
    writel(0xf0, priv->mmio + (0xa << 2));
    ...
    .../* USB3.1 PHY Tuning Code */
    ...
    param_write(priv->phy_grf, &cfg->pipe_sel_usb, true);
    param_write(priv->phy_grf, &cfg->pipe_txcomp_sel, false);
    param_write(priv->phy_grf, &cfg->pipe_txelec_sel, false);
    param_write(priv->phy_grf, &cfg->usb_mode_set, true);

```

### 3.19 RK3576 USB3.1 PHY

RK3576 USB3-OTG0 的 PHY Tuning 代码添加于 kernel/drivers/phy/rockchip/phy-rockchip-usbdp.c 文件中的 udphy\_init() 函数末尾。

```

ret = udphy_status_check(udphy);
if (ret)
    goto assert_phy;
.../* USB3-OTG0 PHY Tuning Code */
return 0;

```

RK3576 USB3-OTG1 的 PHY Tuning 代码添加于 kernel/drivers/phy/rockchip/phy-rockchip-naneng-combphy.c 文件中的 rk3576\_combphy\_cfg() 函数末尾。

```

case PHY_TYPE_USB3:
    /* Set Rx squelch input filler bandwidth */
    writel(0x0d, priv->mmio + (0x14 << 2));
    ...
    .../* USB3-OTG1 PHY Tuning Code */
    ...
    param_write(priv->phy_grf, &cfg->pipe_txcomp_sel, false);
    param_write(priv->phy_grf, &cfg->pipe_txelec_sel, false);
    param_write(priv->phy_grf, &cfg->usb_mode_set, true);
    break;

```

## 3.20 RK3588 USB3.1 PHY

RK3588 USB3-OTG 与 USB3-HOST 代码添加位置不同，如下所示。

### RK3588 USB3-OTG 代码放置位置

USB3-OTG 的 PHY Tuning 目前还没有类似 USB2.0 专门用于 Tuning 的接口，因此代码可以添加于 kernel/drivers/phy/rockchip/phy-rockchip-usbdp.c 文件中的 rk3588\_udphy\_init() 函数末尾。

```
ret = rk3588_udphy_status_check(udphy);
if (ret)
    goto assert_phy;
.../* USB3-OTG PHY Tuning Code */
return 0;
```

### RK3588 USB3-HOST 代码放置位置

USB3-HOST 的 PHY Tuning 代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-naneng-combphy.c 文件的 rk3588\_combphy\_cfg() 函数中。

```
case PHY_TYPE_USB3:
    /* Set SSC downward spread spectrum */
    val = readl(priv->mmio + (0x1f << 2));
    val &= ~GENMASK(5, 4);
    val |= 0x01 << 4;
    ...
    .../* USB3-HOST PHY Tuning Code */
    ...
    param_write(priv->phy_grf, &cfg->pipe_txcomp_sel, false);
    param_write(priv->phy_grf, &cfg->pipe_txelec_sel, false);
    param_write(priv->phy_grf, &cfg->usb_mode_set, true);
    break;
```