

# Rockchip User Guide USB PHY Tuning

文件标识：RK-YH-YF-297

发布版本：V1.0.0

日期：2021-08-13

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© 2021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

本文档提供 Rockchip 平台部分主控芯片的 USB PHYs 信号调节方法。目的是让用户在遇到 USB 信号等相关问题时，可以通过本文档快速完成各USB端口的 PHY 信号调节工作，以满足实际产品中的需求。

芯片名称	内核版本
RK3566、RK3568、RK3399、RK3288、RK3326、PX30	Linux-4.4、Linux-4.19

## 读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

硬件开发工程师

## 修订记录

日期	版本	作者	修改说明
2021-08-13	V1.0.0	杨斌	初始版本

## 目录

### Rockchip User Guide USB PHY Tuning

- 1 Rockchip USB PHY 信号调节说明
- 2 Rockchip USB PHY 信号调节工具使用说明
  - 2.1 Rockchip USB PHY 信号调节工具介绍
  - 2.2 Rockchip USB PHY 信号调节参数介绍
    - 2.2.1 USB2.0 SQ 调节
    - 2.2.2 USB2.0 噪声阈值调节
    - 2.2.3 USB2.0 断开检测阈值调节
- 3 Rockchip USB PHY 信号调节代码生成
  - 3.1 RK356x USB2.0 PHY
  - 3.2 RK3399 USB2.0 PHY
  - 3.3 RK3326/PX30 USB2.0 PHY
  - 3.4 RK3288 USB2.0 PHY
    - 3.4.1 RK3288 kernel-4.19 版本
    - 3.4.2 RK3288 kernel-4.4 版本

# 1 Rockchip USB PHY 信号调节说明

由于 Rockchip 平台的 USB PHY 手册没有外部开放，且主控芯片的 TRM 中也没有各个 USB PHY 寄存器的详细说明，当用户遇到 USB 信号等相关问题时，想要对 USB PHY 的寄存器参数进行调节是一项十分困难的工作。因此 Rockchip 提供了一个“Rockchip USB PHY Tuning Tool.xlsm”工具来简化 USB PHY 的信号调节工作。

该工具位于 SDK 的以下位置：

[RKTools/windows/Rockchip\\_USB\\_PHY\\_Tuning\\_Tool\\_V1.0.rar](#)

当 USB 模块在实际使用中遇到如下相关问题时，且在硬件环境上已没有优化的可能的情况下可以考虑通过“Rockchip USB PHY Tuning Tool.xlsm”工具调节 USB PHY 信号相关参数来软件优化。

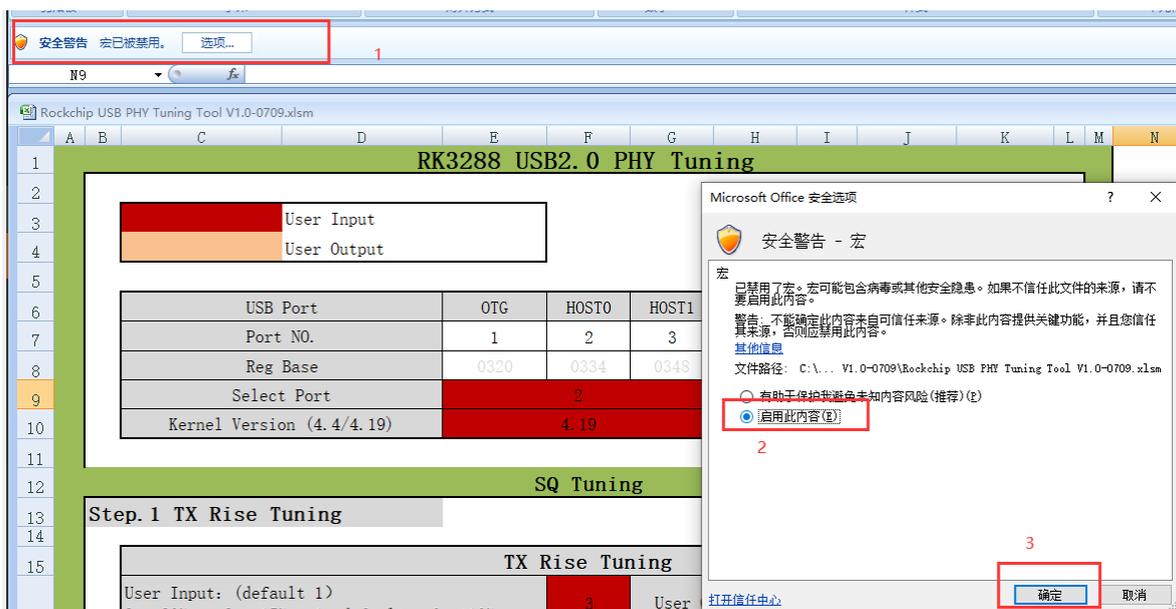
1. USB 眼图指标测试失败问题；
2. 信号质量问题或者 PHY 供电电压差问题引起的 USB 枚举失败；
3. USB 连接外设会自动发生异常断开；
4. USB 连接外设拔掉无法检测到断开事件；

**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，应该优先考虑优化下硬件环境。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节。

# 2 Rockchip USB PHY 信号调节工具使用说明

## 2.1 Rockchip USB PHY 信号调节工具介绍

1. Rockchip USB PHY 信号调节工具里面使用了Excel 的VBA，所以在第一次打开的时候需要按照如下图所示启用宏，否则无法正常获取USB PHY的寄存器和配置参数等。



2. 不同的主控芯片平台调节的参数都会不一样，每一个芯片独立一个sheet页面，只要用户根据芯片型号切换到需要调节的芯片页面进行参数调节即可。
3. 页面中的 User Input 对应的单元格背景颜色表示用户输入栏，在 USB PHY 参数调节的时候只需要根据提示在输入单元格中输入对应的配置参数即可；页面中 User Output 对应的单元格背景颜色表示用户输出的单元格，输出单元格为根据输入的参数自动生成的 Rockchip 平台操作寄存器的 IO 命令。有些主控芯片页面会根据 IO 命令直接生成对应的代码段，有些没有生成代码段，只有转化代码示例，具体参考下一章节的描述或以工具页面具体显示情况为准。



4. 在各个参数调节前，需要在“Select Port”栏中输入待调节的USB端口对应的端口号，USB 端口号从“Port NO.”中获取。比如 RK3568 平台需要调节 HOST1 端口，则需要在“Select Port”栏中输入“2”。其中 USB 端口名称跟 Rockchip 提供的硬件参考电路图上的端口名称是对应。

USB Port	OTG	HOST1	HOST2	HOST3
Port NO.	1	2	3	4
Reg Base (High)	0xFE8A	0xFE8A	0xFE8B	0xFE8B
Reg Base (Low)	0000	0400	0000	0400
Select Port	2			

5. 由于 RK356x 平台的 USB PHY 寄存器没有高16位的“write\_enable bit”，所以为了防止被调节的寄存器中其他不相关或未调节的某些bit值被修改，可以使用IO命令(输入“Select Port”后会自动生成需要的IO命令)，在调节前先获取当前待调节的环境中各个寄存器值，并将这些值全部复制到如下图所示的单元格中(目前仅 RK356x 平台需要此操作，其他平台不需要)。

io -4 -1 0x40 0xFE8B0400				
fe8a0000:	00000004	00000085	000000db	00000060
fe8a0010:	000000e7	00000082	000000aa	00000002
fe8a0020:	00000056	00000055	00000055	00000075
fe8a0030:	00000001	00000000	000000c0	00000068

**注意：**通过IO命令获取的寄存器值一定要保证是原始SDK状态下未经过任何修改的值，如果不确定这些寄存器是否有被修改过，最好重新启动下机器，让寄存器还原成原始SDK中配置的状态。

## 2.2 Rockchip USB PHY 信号调节参数介绍

Rockchip USB PHY 信号调节工具中共提供了三类参数调节，包括 USB2.0 SQ 调节、USB2.0 噪声阈值调节、USB2.0 断开检测阈值调节。

### 2.2.1 USB2.0 SQ 调节

USB2.0 SQ 调节主要调节 USB2.0 信号的眼图高度、驱动强度、预加重、slew rate等。Rockchip平台不同的主控芯片使用不同的USB PHY，所以调节的寄存器参数也都不一样，具体请以工具页面中的调节步骤为准。这里简单介绍下几个常用的调节参数：

调节参数	描述
Tuning Slew Rate	调节 High-speed 眼图的 slew rate ，这个调节一般作用比较小，目前比较少使用。
Tuning HS Eye Height	调节HS眼图幅值, 默认值 400mV。
Tuning HS ODT Value	调节ODT 45Ω电阻值，输入值越大，电阻越小，眼图就越大。
Tuning ODT Compensation Voltage/Current	调节电压/电流校准点，调高校准点可以提高USB眼图的幅度。
Tuning Pre-emphasize Strength	调节预加重强度。
Bypass ODT & Driver Strength Tuning	bypass ODT后调节眼图的驱动强度。

**注意：**USB2.0 SQ Tuning 请按照工具页面中调节顺序进行调节，只有在前面的调节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节（有些参数的调节可能对结果影响很小，那么可以忽略跳过这些步骤）。USB2.0 信号的调节具有一定的风险，不可盲目的调节，最好能测试眼图时对着眼图去适当的调节，调节到能满足需求即可，而不是调节越大越好。用户需要自己把控调节风险。

### 2.2.2 USB2.0 噪声阈值调节

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。此时可以尝试调节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，一般可以调节为125 mV 或者 112.5mV。

### 2.2.3 USB2.0 断开检测阈值调节

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
```

### 2.2.3.1 无法检测USB 2.0 外设断开

当 USB Host 拔掉 USB2.0 外设无法被检测到（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：

**Step 1**：开启 EOP 的预加重（SDK默认都是关闭EOP的预加重）；

**Step 2**：减小USB Host的断开检测阈值；

**注意：**

1. 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；
2. 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 Step 2 的操作；
3. Step 2 减小断开检测阈值存在一定的风险，如果调节过度会导致USB外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。

### 2.2.3.2 USB2.0 外设异常断开

当 USB Host 连接 USB2.0 外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：**

1. 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 外在自身问题或者硬件环境问题等因素导致的，所以不应该一出现此类问题就盲目的去调节断开检测阈值。
2. Rockchip 的SDK默认都是关闭EOP的预加重的，所以此时不需要配置寄存器再去关闭该操作；
3. 增大断开检测阈值同样存在一定的风险，如果调节过度会导致 Host 无法检测到 USB 外在断开连接的情况，所以用户需要自己把控风险。

## 3 Rockchip USB PHY 信号调节代码生成

USB 信号调节过程都是使用IO命令完成的，在 USB PHY 寄存器参数调节完成后可以将IO命令转化成驱动代码集成在固件里面。由于 Rockchip 平台不同的主控芯片使用的 USB PHY 都各不相同，所以IO命令对应的驱动代码和代码路径也都不相同。

### 3.1 RK356x USB2.0 PHY

Rockchip USB PHY 信号调节工具中的 RK356x 页面会根据生成的IO命令在页面右侧直接转化生成对应的驱动代码段，只要将需要调节的部分代码段复制添加到 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 中的rk3568\_usb2phy\_tuning 函数末尾进行编译即可。

由于 RK356x 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基地址进行判断区分不同的 USB 端口。

OTG 口和HOST1 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 */  
}
```

## 3.2 RK3399 USB2.0 PHY

由于 RK3399 平台的IO命令转化为驱动代码比较简单，所以在 Rockchip USB PHY 信号调节工具中的 RK3399 页面中没有自动生成IO命令代码，但是里面有IO命令转换代码示例可以参考。我们可以将 USB PHY 调节后的IO命令转化成代码添加在 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，如下图所示位置。

```

if (rphy->phy_cfg->reg == 0xe450) {
    /*
     * Disable the pre-emphasize in eop state
     * and chirp state to avoid mis-trigger the
     * disconnect detection and also avoid hs
     * handshake fail for PHY0.
     */
    ret |= regmap_write(rphy->grf, 0x4480,
                       GENMASK(17, 16) | 0x0);
    ret |= regmap_write(rphy->grf, 0x44b4,
                       GENMASK(17, 16) | 0x0);
} else {
    /*
     * Disable the pre-emphasize in eop state
     * and chirp state to avoid mis-trigger the
     * disconnect detection and also avoid hs
     * handshake fail for PHY1.
     */
    ret |= regmap_write(rphy->grf, 0x4500,
                       GENMASK(17, 16) | 0x0);
    ret |= regmap_write(rphy->grf, 0x4534,
                       GENMASK(17, 16) | 0x0);
}

```

```

if (!of_property_read_bool(node, "rockchip,u2phy-tuning"))
    return ret;

```

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 属于一组 PHY (区分地址：0xe460)。所以IO命令转化后的代码段前面需要需要加上寄存器的地址判断来区分不同的USB端口。

TYPE-C0 口和HOST0 口代码添加如下：

```

if (rphy->phy_cfg->reg == 0xe450) {
    ... /* TYPE-C0/HOST0 PHY Tuning Code */
}

```

TYPE-C1 口和HOST1 口代码添加如下：

```

if (rphy->phy_cfg->reg == 0xe460) {
    ... /* TYPE-C1/HOST1 PHY Tuning Code */
}

```

### 3.3 RK3326/PX30 USB2.0 PHY

由于 RK3326/PX30 平台的IO命令转化为驱动代码比较简单，所以在Rockchip USB PHY 信号调节工具的RK3326/PX30 页面中没有自动生成IO命令对应的代码，但是里面有IO命令转换代码示例可以参考。我们可以将USB PHY 信号调节的IO命令转化为代码添加在 kernel/drivers/phy/rockchip/phy-rockchip-inno-usb2.c 的rk3328\_usb2phy\_tuning 函数末尾进行编译即可。

## 3.4 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kernel版本Kernel Version都选择4.4， kernel-4.19及以上kernel版本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）。

### 3.4.1 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 口代码添加如下：

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

HOST1 口代码添加如下：

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

HOST2 口代码添加如下：

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

### 3.4.2 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。