**SparkRoad Brief**

Designed by Verimake

2019年5月

**目 录**

[第一章 SparkRoad 开发板简介 3](#_Toc8753511)

[1.1开发板资源 3](#_Toc8753512)

[1.2 主控FPGA资源 4](#_Toc8753513)

[1.3开发资料获取方式 4](#_Toc8753514)

[第二章 开发环境准备 5](#_Toc8753515)

[2.1 Windows环境 5](#_Toc8753516)

[2.2 Ubuntu环境 6](#_Toc8753517)

[第三章 Hello LED开发板功能演示 8](#_Toc8753518)

[3.1 Windows环境 8](#_Toc8753519)

[3.2 Ubuntu环境 9](#_Toc8753520)

[第四章 Hello RISC-V软核开发 10](#_Toc8753521)

[3.1 Windows环境 10](#_Toc8753522)

[3.2 Ubuntu环境 12](#_Toc8753523)

[第五章 开发板资源介绍 14](#_Toc8753524)

[5.1电源拓朴 14](#_Toc8753525)

[5.2 按键和RGB LED 15](#_Toc8753526)

[5.3 micro SD 接口 15](#_Toc8753527)

[5.4 ESP32-S module 接口 16](#_Toc8753528)

[5.5 FPGA管脚资源分配表 17](#_Toc8753529)

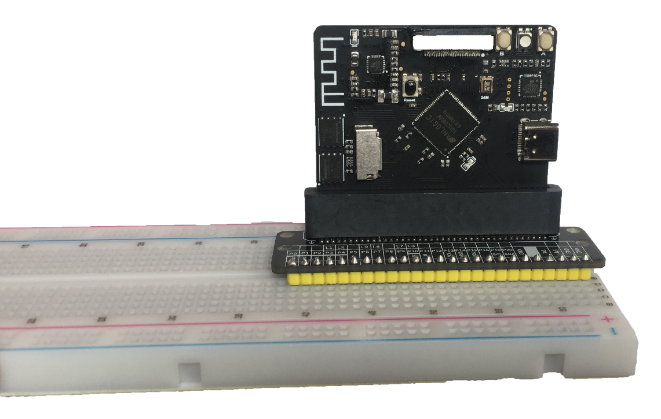
[5.6 micro:bit拓展接口 19](#_Toc8753530)

[版本信息 19](#_Toc8753531)

# 第一章 SparkRoad 开发板简介

## 1.1开发板资源

1. 平台通过USB-C提供稳定电源。
2. 板载USB JTAG 可支持FPGA下载、调试、flash烧录等功能
3. 两个用户按键
4. 三色RGB LED灯
5. FPGA配置flash，六线连接FPGA
6. 8-Pad WSON Package 六线flash接口
7. 六线microSD卡接口
8. DIP-38 Package ESP32-S接口
9. FPGA芯片内嵌64Mbit SDRAM
10. 0.5mmFPC-24P标准DVP Camera接口
11. micro：bit金手指兼容大量BBC外设



**图1.1-1 SparkRoad实物图**

## 1.2 主控FPGA资源

SparkRoad开发板主控采用上海安路科技的高性能FPGA产品EG4S20NG88。

1. 等效23520个四输入查找表
2. 最大156.8K 分布式RAM
3. 64块9Kb嵌入式ram和16块32Kb嵌入式ram
4. 64Mbit SDRAM，最高200MHz工作频率
5. 16个全局时钟，3个PLL
6. 1MSPS 12-bit SAR型ADC

## 1.3开发资料获取方式

Github网址: <https://github.com/verimake-team/SparkRoad-FPGA>

微信公众号：VeriMake

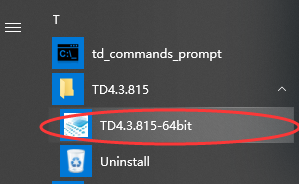
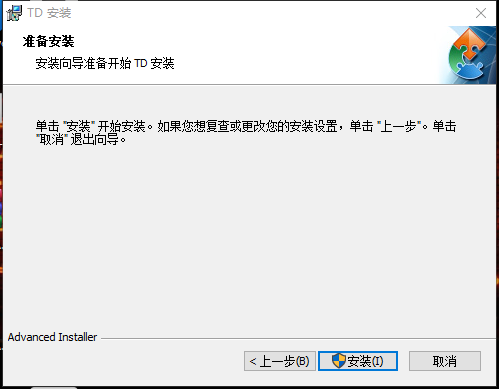
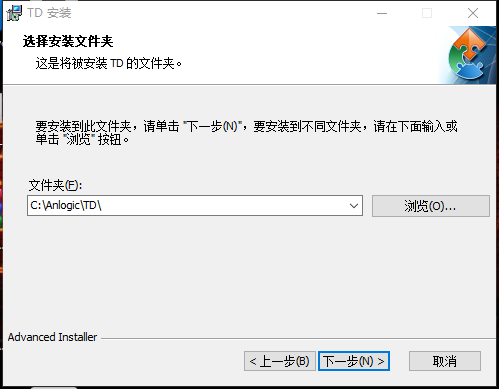
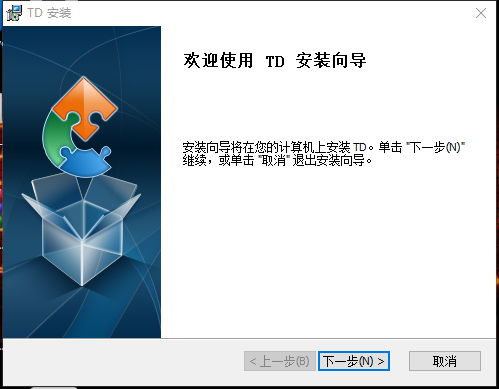
Bilibili ID：VeriMake



# 第二章 开发环境准备

## 2.1 Windows环境

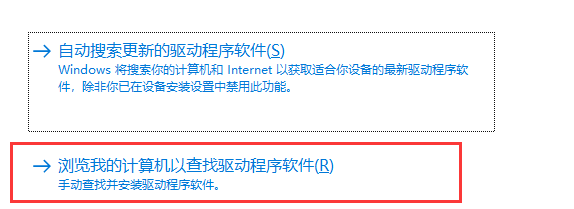
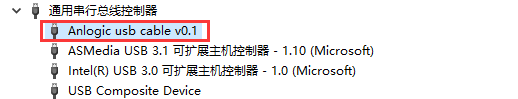
* **TD软件安装**



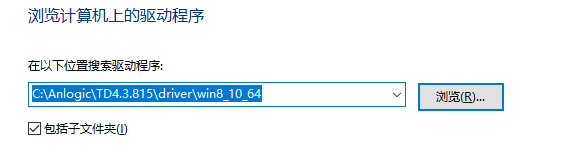
**图2.1-1 windows环境TD安装流程**

* **开发板驱动安装**

1. 开发板通过USB-C线连接到电脑。右键“计算机->设备管理器”，在弹出的新设备中点右键选择“更新驱动程序软件”，在弹出的选项中点击“浏览计算机以查找驱动程序软件”。



1. 选择TD软件安装路径下的“driver”目录，并点击下一步完成更新。



1. 进入启动选项，禁用windows强制数字签名（也可永久禁用避免重复操作，安全后果自负）。“win键->shift键+重启->疑难解答->高级选项->启动设置->禁用驱动程序强制签名”

## 2.2 Ubuntu环境

* **TD软件安装**

1. 确保和TD程序的压缩包在同一目录下
2. 在/opt中创建TD IDE的安装位置

sudo mkdir /opt/TD\_DECEMBER2018

1. 即将TD提取到创建的目录

sudo tar -xvf TD\_DECEMBER2018\_GOLDEN\_RHEL.tar.gz -d /opt/TD\_DECEMBER2018/

1. 在/usr/bin/td目录下创建一个链接文件来执行/opt/TD\_DECEMBER2018/bin/td

sudo ln -s /opt/TD\_DECEMBER2018/bin/td /usr/bin/td

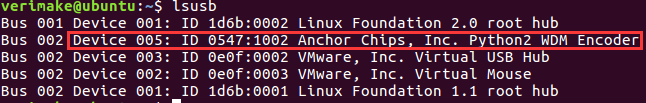
1. 以GUI模式打开TD IDE

td -gui

* **开发板驱动安装**

1）开发板通过USB-C线连接到电脑。执行

lsusb



确保USB设备的VID：PID为0547：1002

2）创建udev rule file

sudo vi /etc/udev/rules.d/91-anlogic-jtag.rules

3）输入以下信息

SUBSYSTEMS=="usb", ATTRS{idVendor}=="0547", ATTRS{idProduct}=="1002", \

GROUP="plugdev", \

MODE="0660"

1. 重启udev service

sudo service udev restart

1. 查看TD IDE能否识别驱动设备

./td -gui

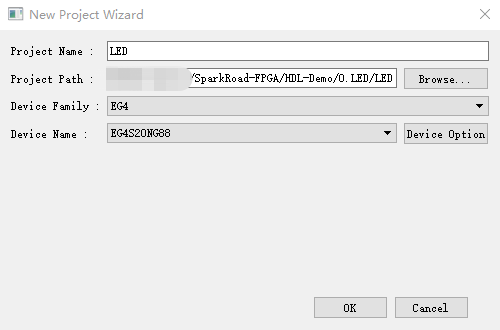
点击”tools->download”查看是否识别出usb设备

# 第三章 Hello LED开发板功能演示

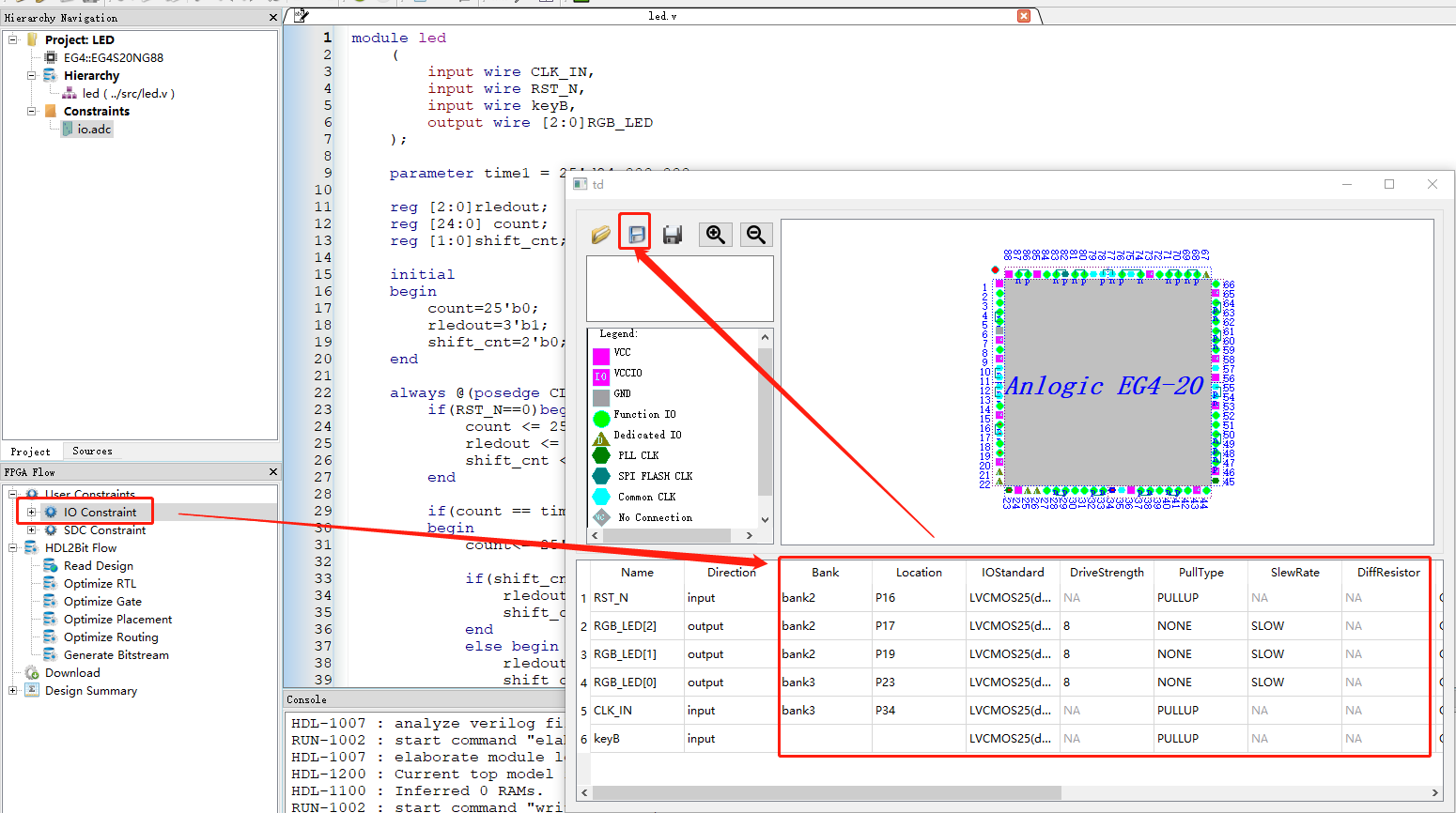
此处添加led演示流程，包括建立工程、编写代码、编译、下载、演示

## 3.1 Windows环境

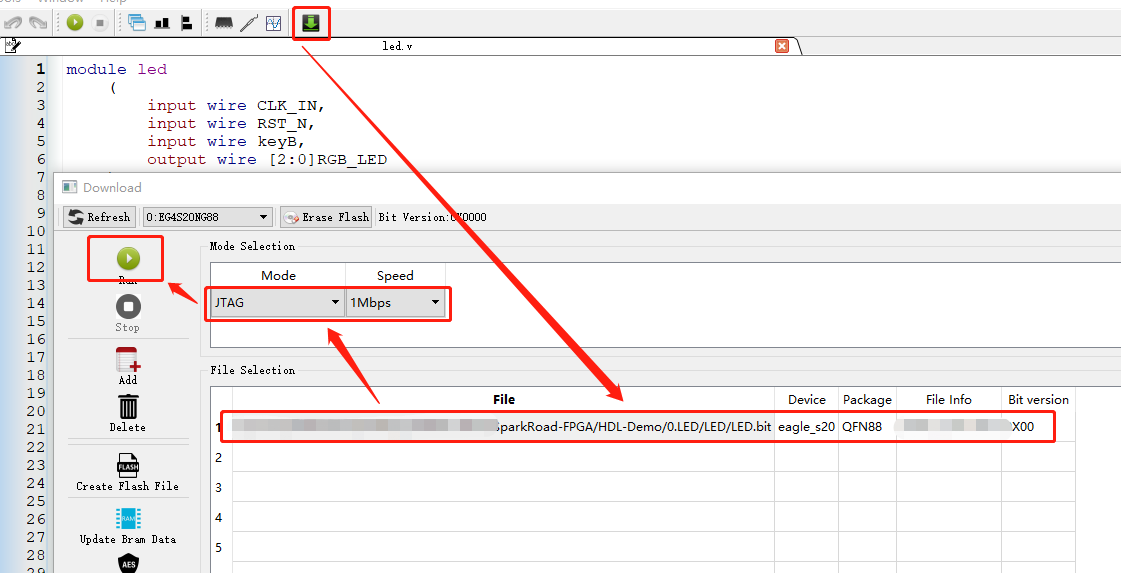
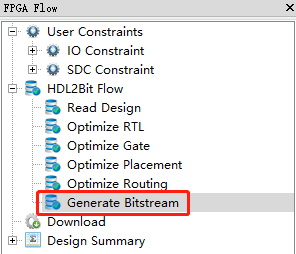
1. 打开TD软件，顶部菜单选择“Project->New Project”按照下图，方式新建工程。目录自选



1. 顶部工具栏“Source->Add Source”选择/SparkRoad/HDL-Demo/0.LED/src目录下的led.v添加进工程之后，再按照下图方式添加并保存引脚约束文件



1. 综合工程生成bitstream文件，将SparkRoad用USB线缆连接到电脑上，并按照下图方式将bit文件下载到SparkRoad上。下载选项可选Jtag仿真或者Flash下载。



## 3.2 Ubuntu环境

./td -gui

启动TD-gui界面，其余流程与windows一致。

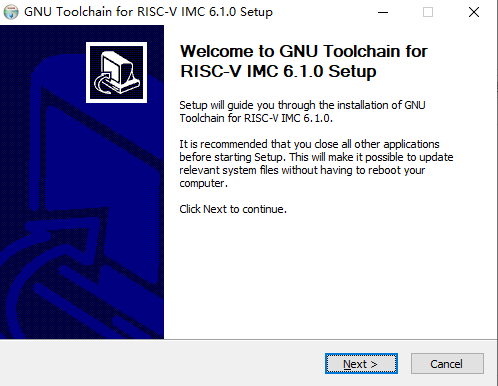
# 第四章 Hello RISC-V软核开发

## 3.1 Windows环境

1. 下载安路科技提供的riscv-toolchain.exe并安装

链接：https://pan.baidu.com/s/160Iu03p4NvlcNUw18\_msDQ

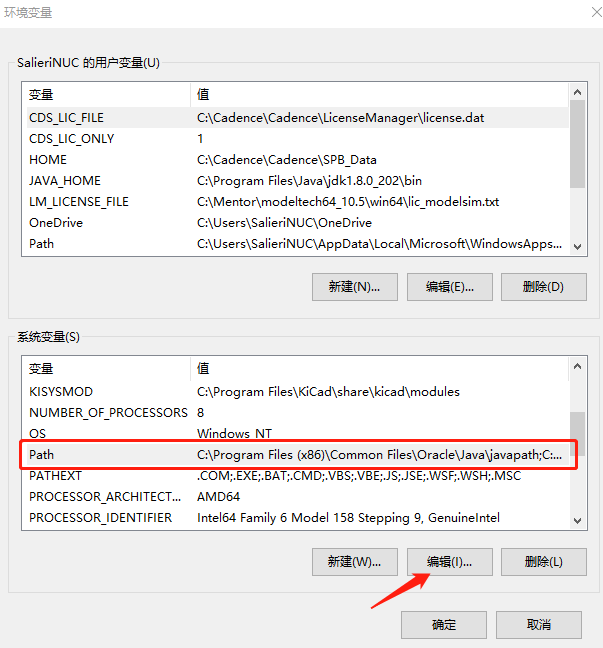
提取码：fv59



**图3.1-1 GNU toolchain for RISC-V**

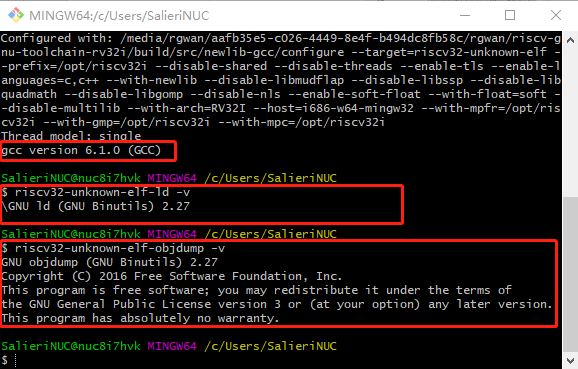
1. 配置环境变量

右键“计算机->属性->高级系统设置”把上面toolchain的安装目录下的bin目录添加到PATH



**图3.1-2 windows环境添加环境变量**

1. 在任意terminal中测试toolchain



**图3.1-3 Git bash演示**

1. 从github上clone资料包

git clone https://github.com/verimake-team/SparkRoad-FPGA.git

进入软件代码目录

\SparkRoad-FPGA\RISC-V\firmware

在此目录下，依次用terminal执行以下指令编译软件代码（以git bash为例）

riscv32-unknown-elf-gcc -ffreestanding -nostdlib -o firmware.elf firmware.S firmware.c -march=rv32i --std=gnu99 -Wl,-Bstatic,-T,firmware.lds,-Map,firmware.map,--strip-debug -lgcc

riscv32-unknown-elf-objcopy -O binary firmware.elf firmware.bin

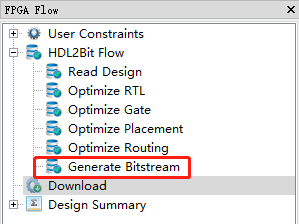
rom2mif firmware.bin hi.mif mh.mif ml.mif lo.mif

mv \*.mif ../hardware/Picorv32/al\_ip/

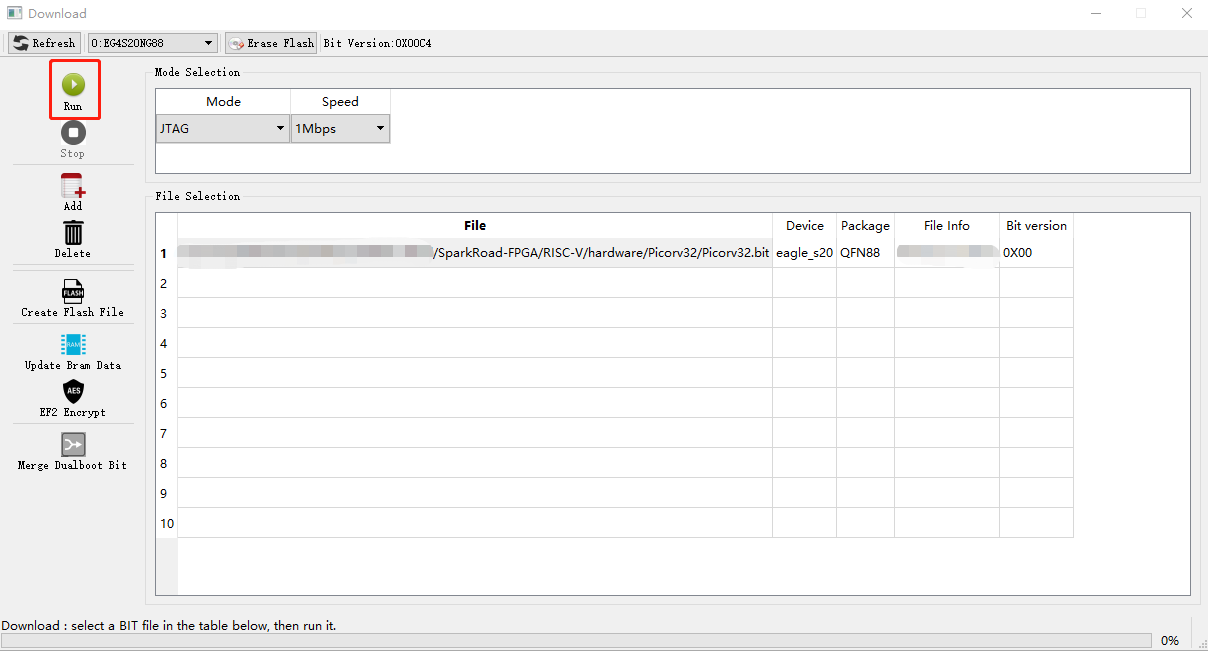
1. 进入RTL代码目录

\SparkRoad-FPGA\RISC-V\hardware\Picorv32

用TD工具打开工程文件Picorv32.al对FPGA工程进行综合



随后从顶部菜单进入“Tools->Download”仿真/下载上一步生成的Bit文件



将金手指的P13（TX）和P14（RX）连接上USB-TTL串口。打开任意串口助手类工具，按下SparkRoad最右端按键，复位系统即可看见以下log



## 3.2 Ubuntu环境

1. 获取软件包并进入目录

   git clone https://github.com/verimake-team/SparkRoad-FPGA.git

   cd SparkRoad-FPGA

1. 安装编译环境。由于网络环境问题，我们也额外提供一份百度云云盘。

   链接：https://pan.baidu.com/s/160Iu03p4NvlcNUw18\_msDQ

 提取码：fv59

1. 将百度云盘下载好的riscv-gnu-toolchain-rv32i.tar.gz放在/SparkRoad-FPGA目录下再依次执行以下命令

sudo mkdir /opt/riscv32i

    sudo chown $USER /opt/riscv32i

    make get-tools

    sudo tar -zxvf riscv-gnu-toolchain-rv32i.tar.gz

    sudo chown $USER riscv-gnu-toolchain-rv32i

    cd riscv-gnu-toolchain-rv32i; mkdir build; cd build

    ../configure --with-arch=rv32i --prefix=/opt/riscv32i

    make -j$(nproc)

    cd ../../RISC-V/firmware

    gcc -Wall rom2mif -o rom2mif

    sudo mv rom2mif /opt/riscv32i/bin

1. 执行完以上步骤之后，编译环境就已经被安装在/opt/rv32i目录下。只需要再配置环境变量即可(配置完环境变量后，系统用户重新登陆才能生效)。

vi /etc/profile

并在末尾加入

export PATH="$PATH:/opt/riscv32i/bin"

1. 系统重新登陆后进入/SparkRoad-FPGA/RISC-V/firmware依次执行以下指令编译软件代码

make clean

make firmware.elf

make firmware.bin

make lo.mif

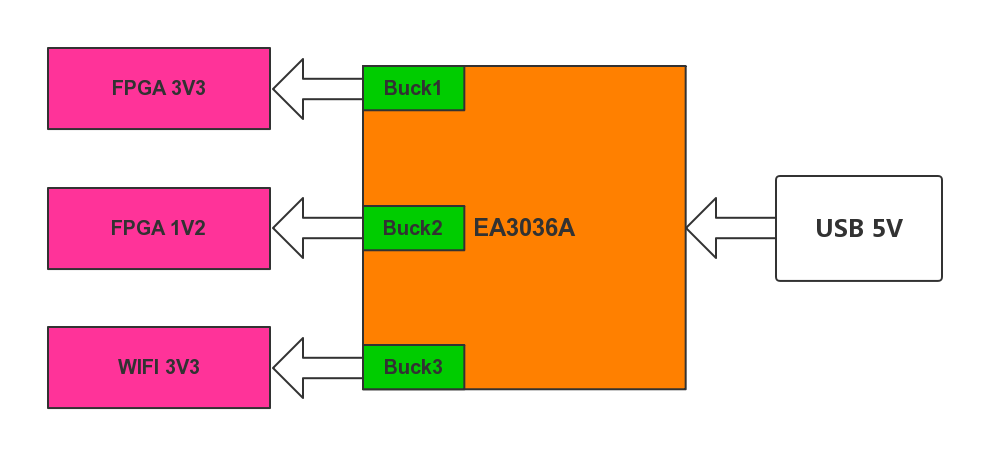
1. 进入RTL工程目录/SparkRoad-FPGA/RISC-V/hardware/Picorv32。之后的流程及现象与windows环境下相同，参考上一节。

# 第五章 开发板资源介绍

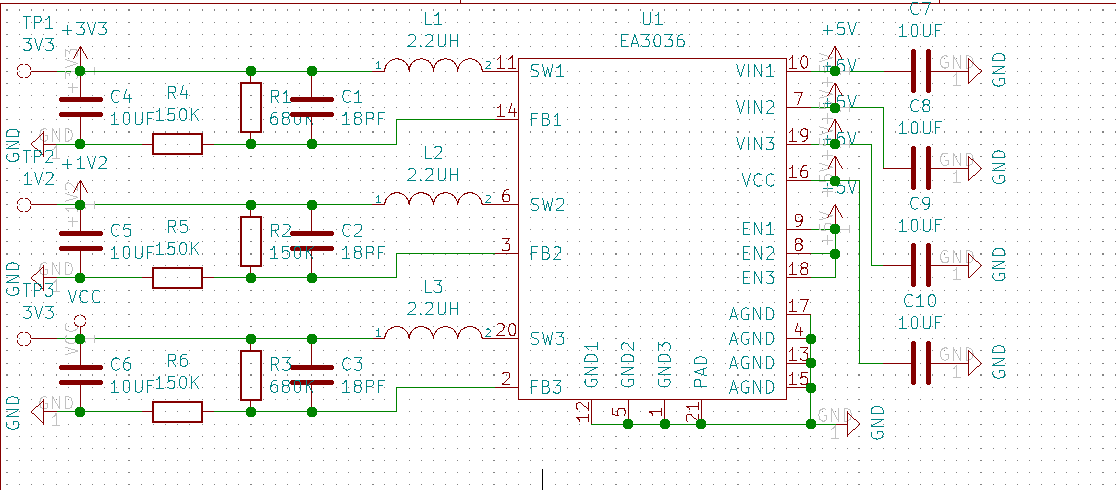
## 5.1电源拓朴

开发板采用USB-C供电，一般使用普通电脑USB口供电足够，若焊接了ESP32-S模块，可能会超过电脑USB供电能力，供电不足会导致开发板不稳定，可通过micro：bit上的金手指提供外部5V供电。

开发板的5V电源输入通过三路Buck DC-DC电源管理芯片EA3036转换，其中两路转换为3.3V和1.2V供FPGA主控使用。额外一路Buck输出连接到ESP32-S模块的供电接口上，确保WIFI能够稳定工作。

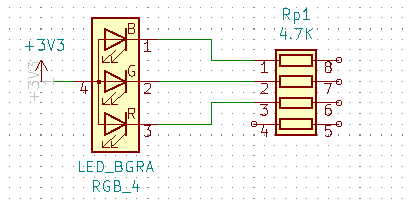
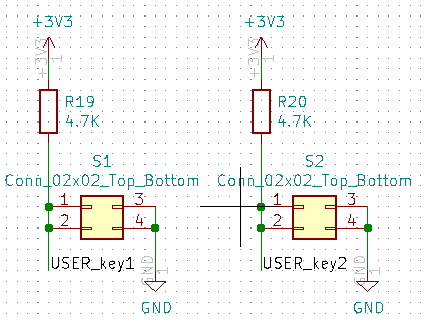


**图5.1-1 EA3036电源拓扑**



**图5.1-2 EA3036硬件原理图**

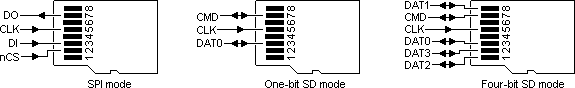
## 5.2 按键和RGB LED



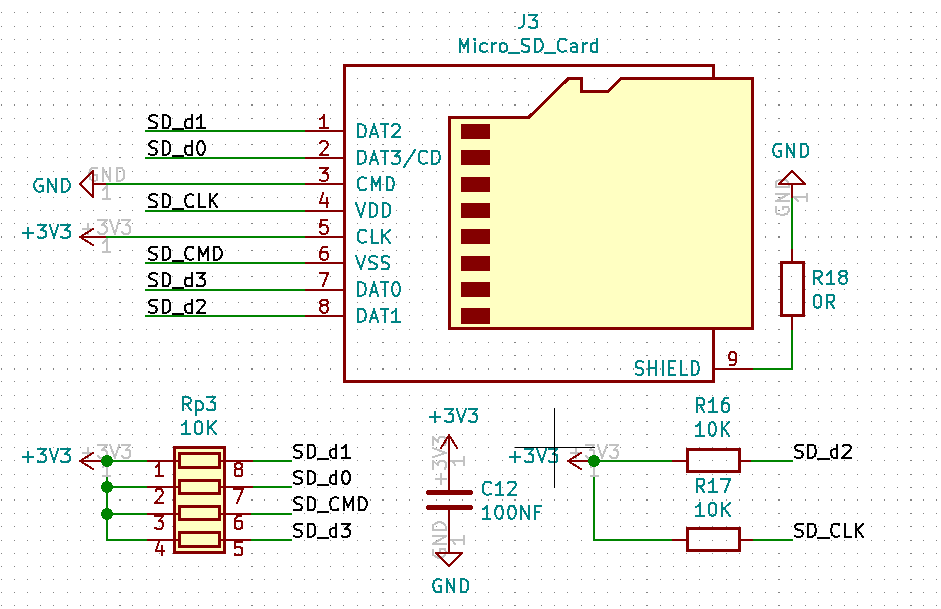
**图5.2-1 按键和RGB硬件原理图**

## 5.3 micro SD 接口

microSD卡接了六个IO到FPGA上，并且预留了上拉焊盘。可根据具体需求设计成不同读写模式。

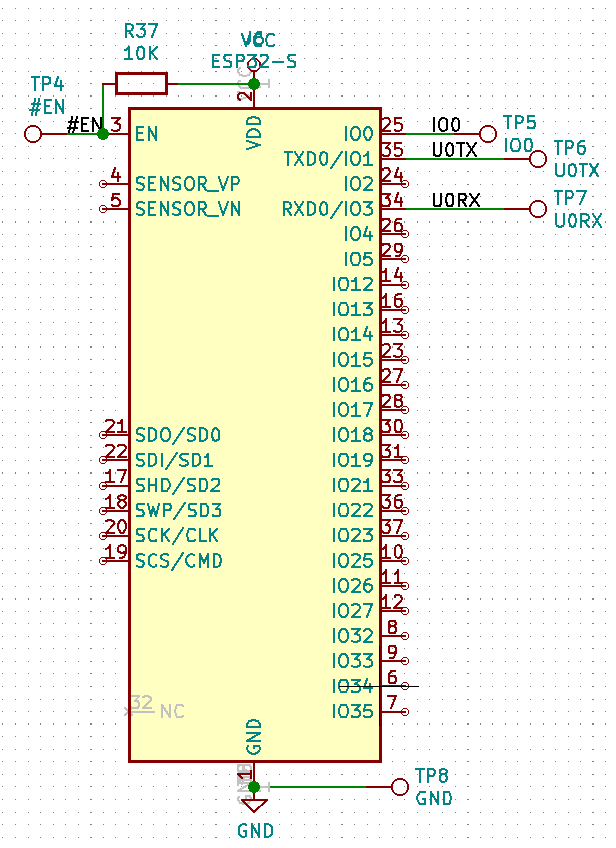


**图5.3-1 SD卡不同控制模式**

****

**图5.3-2 SD硬件原理图**

## 5.4 ESP32-S module 接口

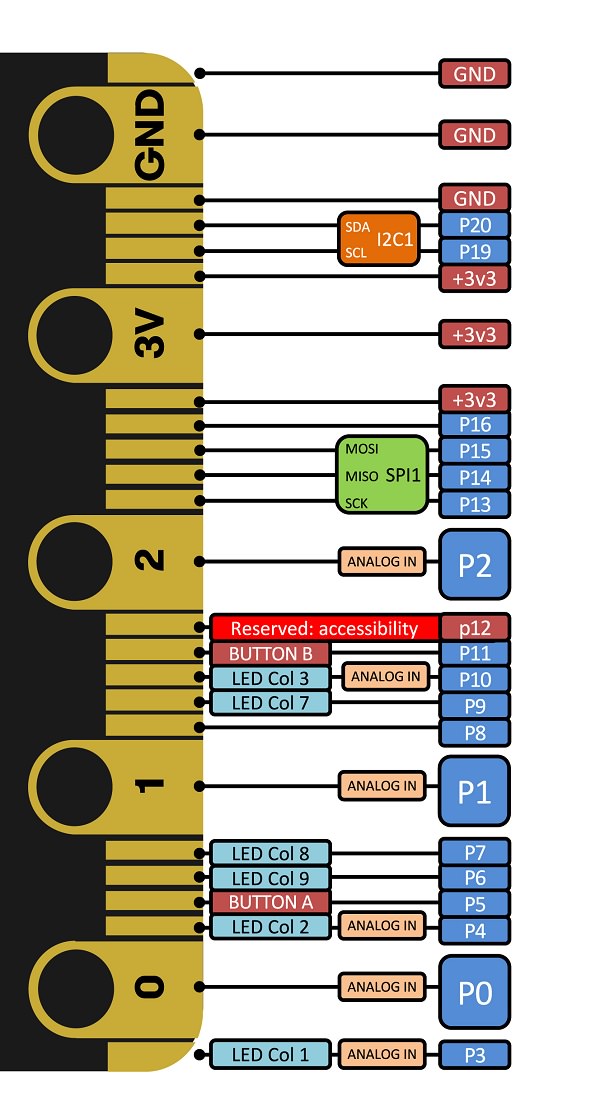
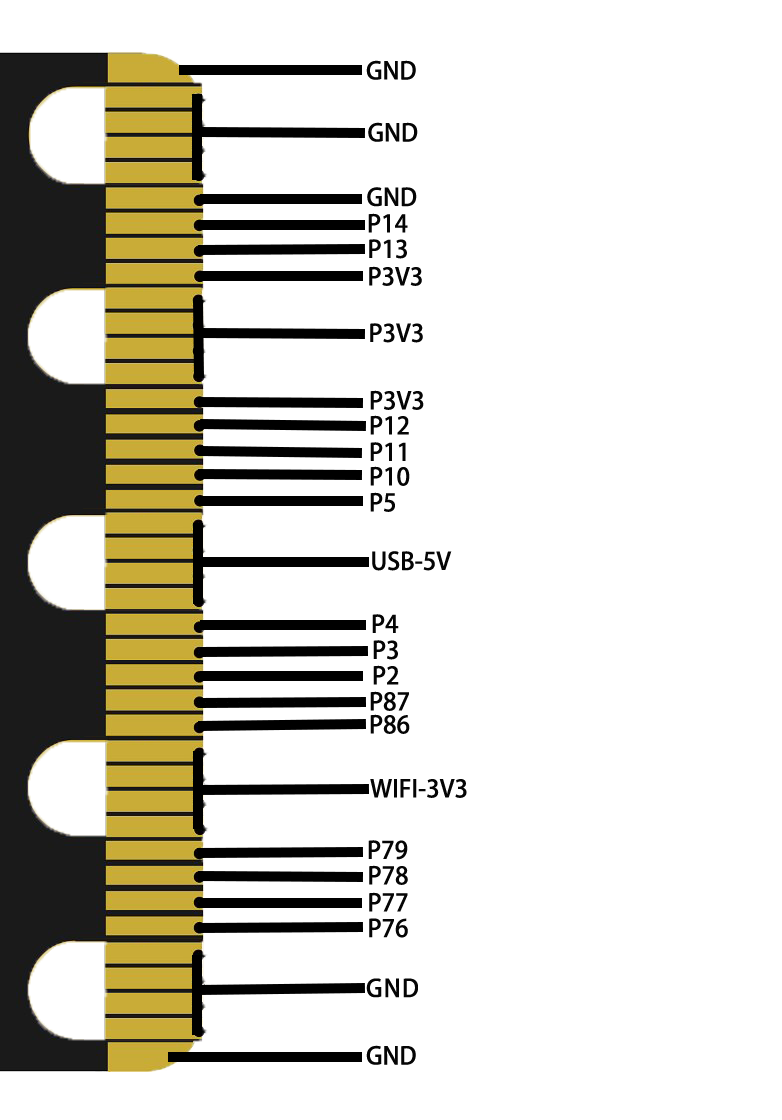


**图5.4-1 ESP32-S接口原理图**

## 5.5 FPGA管脚资源分配表

|  |  |  |
| --- | --- | --- |
|  | **网络标号** | **管脚名称** |
| **24M OSC** | **24M** | **P34** |
| **PROGRAM RST BOTTON** | **PROGRAM** | **P67** |
| **USER BOTTON A** | **USER\_key1** | **P16** |
| **USER BOTTON B** | **USER\_key2** | **P18** |
| **RGB LED** | **LED\_B** | **P17** |
| **RGB LED** | **LED\_G** | **P19** |
| **RGB LED** | **LED\_R** | **P23** |
| **FPGA JTAG** | **JTAG\_TDO** | **P21** |
| **FPGA JTAG** | **JTAG\_TMS** | **P22** |
| **FPGA JTAG** | **JTAG\_TDI** | **P25** |
| **FPGA JTAG** | **JTAG\_TCK** | **P26** |
| **DVP** | **d0** | **P27** |
| **DVP** | **d1** | **P28** |
| **DVP** | **d2** | **P32** |
| **DVP** | **d3** | **P30** |
| **DVP** | **d4** | **P29** |
| **DVP** | **d5** | **P31** |
| **DVP** | **d6** | **P33** |
| **DVP** | **d7** | **P37** |
| **DVP** | **d8** | **P38** |
| **DVP** | **d9** | **P40** |
| **DVP** | **PCLK** | **P35** |
| **DVP** | **HREF** | **P41** |
| **DVP** | **VSYNC** | **P45** |
| **DVP** | **XCLK** | **P39** |
| **DVP** | **PWDN** | **P42** |
| **DVP** | **CAMSIOC** | **P48** |
| **DVP** | **CAMSIOD** | **P49** |
| **DVP** | **CAMRST** | **P47** |
| **USER\_FLASH** | **USER\_cs** | **P54** |
| **USER\_FLASH** | **USER\_miso** | **P55** |
| **USER\_FLASH** | **USER\_wp** | **P57** |
| **USER\_FLASH** | **USER\_hold** | **P61** |
| **USER\_FLASH** | **USER\_cclk** | **P60** |
| **USER\_FLASH** | **USER\_mosi** | **P59** |
| **microSD** | **SD\_d1** | **P62** |
| **microSD** | **SD\_d0** | **P63** |
| **microSD** | **SD\_CLK** | **P64** |
| **microSD** | **SD\_CMD** | **P66** |
| **microSD** | **SD\_d3** | **P74** |
| **microSD** | **SD\_d2** | **P75** |
| **ESP32-S** | **U0TX** | **P51** |
| **ESP32-S** | **U0RX** | **P50** |
| **ESP32-S** | **#EN** | **P72** |
| **ESP32-S** | **IO0** | **P52** |

## 5.6 micro:bit拓展接口



**图5-5-1 SparkRoad(左)对比micro:bit(右)接口**

# 版本信息

|  |  |  |
| --- | --- | --- |
| 日期 | 版本 | 修订记录 |
| 5/5/2019 | 1.0 | 初版建立 |
|  |  |  |
|  |  |  |