



RV-STAR 开发板

用户指南 V2.0

版权声明

版权所有 © 2018–2020 芯来科技（Nuclei System Technology）有限公司。保留所有权利。

Nuclei™是芯来科技公司拥有的商标。本文件使用的所有其他商标为各持有公司所有。

本文文件描述的产品将不断发展和完善；此处的信息由芯来科技提供，但不做任何担保。

本文件仅用于帮助读者使用该产品。对于因采用本文件的任何信息或错误使用产品造成的任何损失或损害，芯来科技概不负责。

联系我们

若您有任何疑问，请通过电子邮件 support@nucleisys.com 联系芯来科技。

修订历史

版本号	修订日期	修订的章节	修订的内容
1.0	2019/12/20	N/A	1. 初始版本
2.0	2020/11/12	第 4 章	1. 快速开发部分所使用 IDE 更换为 Nuclei Studio

目录

版权声明.....	0
联系我们.....	0
修订历史.....	1
表格清单.....	3
图片清单.....	4
1. 简介	5
2. 硬件功能模块描述	7
2.1. 时钟	7
2.2. 电源.....	8
2.3. 通用串行总线（USB）OTG.....	9
2.4. SD 卡	9
2.5. 指示与用户 LED.....	10
2.6. 用户按键.....	10
3. 扩展接口和跳线	11
3.1. ARDUINO 兼容的扩展接口	11
3.2. PMOD 扩展接口.....	15
3.3. 跳线列表.....	17
4. 快速开发 RV-STAR.....	18
4.1. 新建 HELLOWORLD 示例工程	18
4.2. 运行 HELLOWORLD 示例	19
4.3. 更多示例.....	22
5. 参考文档	25

表格清单

表 3-1 扩展接口 P3 引脚信号	11
表 3-2 扩展接口 P4 引脚信号	13
表 3-3 扩展接口 P5 引脚信号	13
表 3-4 扩展接口 P6 引脚信号	14
表 3-5 扩展接口 PMOD1 引脚信号	15
表 3-6 扩展接口 PMOD2 引脚信号	16
表 3-7 跳线功能说明	17

图片清单

图 1-1 RV-STAR 开发板示意图.....	5
图 2-1 RV-STAR 开发板系统框图	7
图 2-2 时钟电路	7
图 2-3 板载 3V3 电源	8
图 2-4 USB OTG 接口电路	9
图 2-5 SD 卡电路	9
图 2-6 指示和用户 LED.....	10
图 2-7 按键.....	10
图 3-1 ARDUINO 兼容的扩展接口	11
图 4-1 新建 HELLOWORLD 示例工程	19
图 4-2 编译 HELLOWORLD 示例	19
图 4-3 调试连接	20
图 4-4 调试器 WINDOWS 驱动下载及文档获取	21
图 4-5 运行 HELLOWORLD 示例	21
图 4-6 DEMO1 演示效果	22
图 4-7 DEMO2 演示效果.....	23
图 4-8 DEMO3 演示效果.....	24
图 4-9 DEMO4 演示效果.....	24

1. 简介

RV-STAR 是一款基于 GD32VF103 MCU 的 RISC-V 评估开发板，提供了板载调试器、Reset 和 Wakeup 用户按键、RGB LED、USB OTG，以及 EXMC、Arduino 和 PMOD 扩展接口等资源。

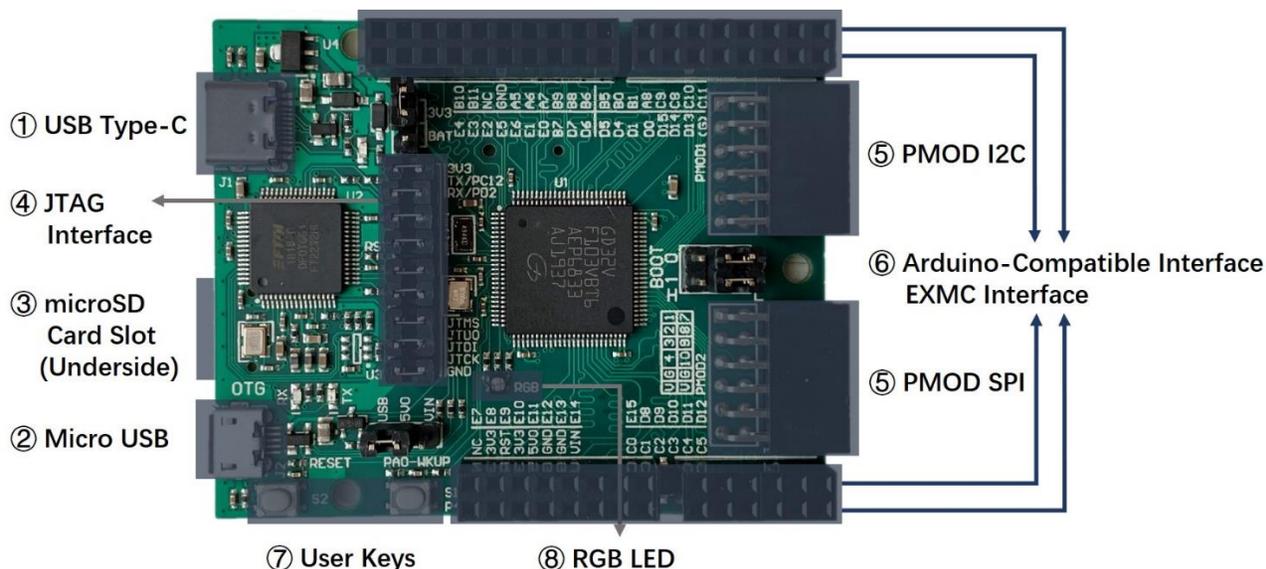


图 1-1 RV-STAR 开发板示意图

RV-STAR 开发板及功能简介：

- 微控制器：GD32VF103VBT6（32 位 RISC-V 处理器）
 - 内核：芯来科技 Bumblebee 内核（RV32IMAC）
 - 主频：108MHz
 - 内存：内置 128KB Flash、32KB SRAM
 - 工作电压：2.6~3.6V
 - 外设资源：Timer（高级 16 位定时器*1，通用 16 位定时器*4）、U(S)ART*5、I2C*2、SPI*3、CAN*2、USBFS*1、ADC*2（16 路外部通道）、DAC*2、EXMC*1

- 供电方式：5V USB 或者 5~9V 外部直流电源（Arduino Vin）
- 尺寸：66mm*53.4mm
- 外设及接口
 - ① USB Type-C 接口：下载、调试、串口通信功能
 - ② Micro USB 接口：USB-OTG 功能
 - ③ microSD 卡插槽（默认没焊接）：外扩 SD 卡存储（SPI 接口）
 - ④ JTAG 接口：可分离 MCU 与调试器，使其各自可单独工作
 - ⑤ PMOD 接口*2：SPI、I2C
 - ⑥ 双排标准 2.54mm 排母接口：Arduino 兼容接口（外侧），EXMC 扩展接口（内侧）
 - ⑦ 用户按键*2：复位、唤醒
 - ⑧ RGB LED*1

注：

关于 GD32V103 MCU 和 Bumblebee 内核的更多详情，请阅读其各自对应的用户手册，可访问 www.rvmcu.com 获取。

2. 硬件功能模块描述

RV-STAR 开发板是四层 PCB 板，板上集成了 GD32VF103 MCU 芯片必需的电源、时钟电路及一些基础外设，系统整体框架结构如下图 2-1 所示。

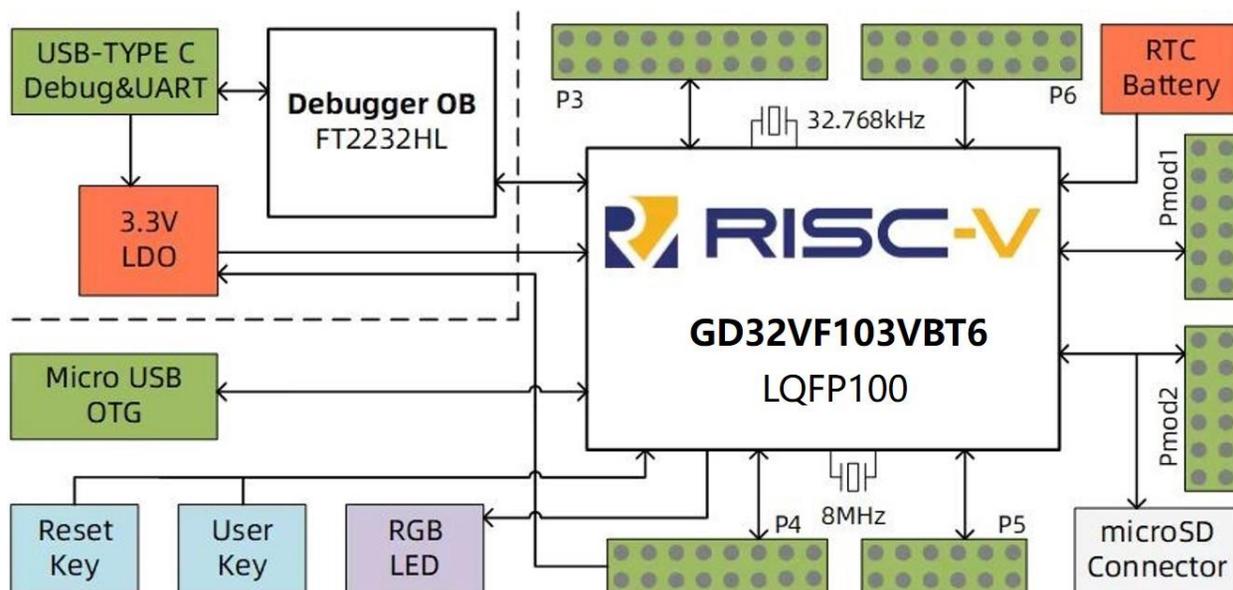


图 2-1 RV-STAR 开发板系统框图

2.1. 时钟

RV-STAR 开发板提供了两个外部时钟源。一个 8MHz 无源晶振（Y3）作为 GD32VF103 MCU 芯片的时钟源，经 MCU 片内 PLL 倍频可产生高达 108MHz 的系统时钟。另一个 32.768KHz 无源晶振（Y2）为 MCU 的实时时钟电路提供低功耗精确的时钟基准。此外，用户也可以不使用外部时钟源，而只使用 MCU 内部的时钟源。

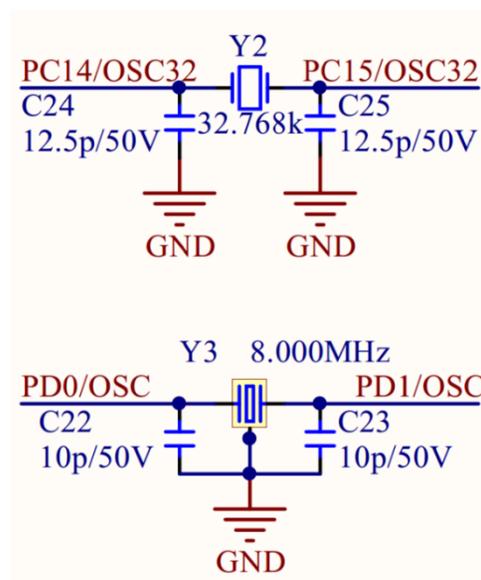


图 2-2 时钟电路

2.2. 电源

RV-STAR 开发板上 GD32VF103 MCU 芯片使用单路 3.3V 电源供电，板载电源稳压电路如下图所示。

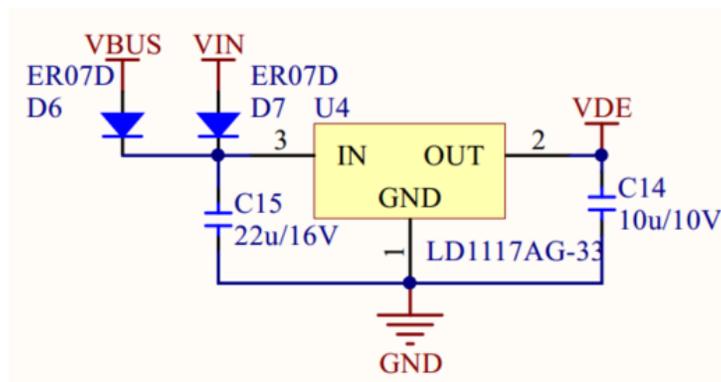


图 2-3 板载 3v3 电源

RV-STAR 开发板支持多种方式的电源输入：

- USB Type-C 接口（调试器）：5V
- Arduino 接口 VIN：5-9V
- Micro-USB 接口（OTG）：5V

在正确连接跳线的条件下允许同时连接多种输入，跳线的连接方式将在第 3 章介绍。建议输入电源供电能力大于等于 500mA。Micro-USB 接口支持双向供电，默认为输入，在仅通过其单一供电的情况下，请将 MCU 的 VBUS_OE#（PD3）信号拉低，使 MOSFET（Q1）完全开通以保证良好的供电。

2.3. 通用串行总线（USB）OTG

GD32VF103 MCU 具有一个全速 USB 模块，支持主机模式、设备模式，以及遵循 HNP（主机协商协议）和 SRP（会话请求协议）的 OTG 模式。板载 USB OTG 接口部分的电路如下图所示，共有 5 个与 MCU 连接的信号：USB_DP，USB_DN，USB_ID，USB_VBUS，VBUS_OE#。SESR12 为 ESD 保护二极管。当通过该接口做电源输出时，需要拉低 VBUS_OE# 使 MOSFET（Q1）开通。

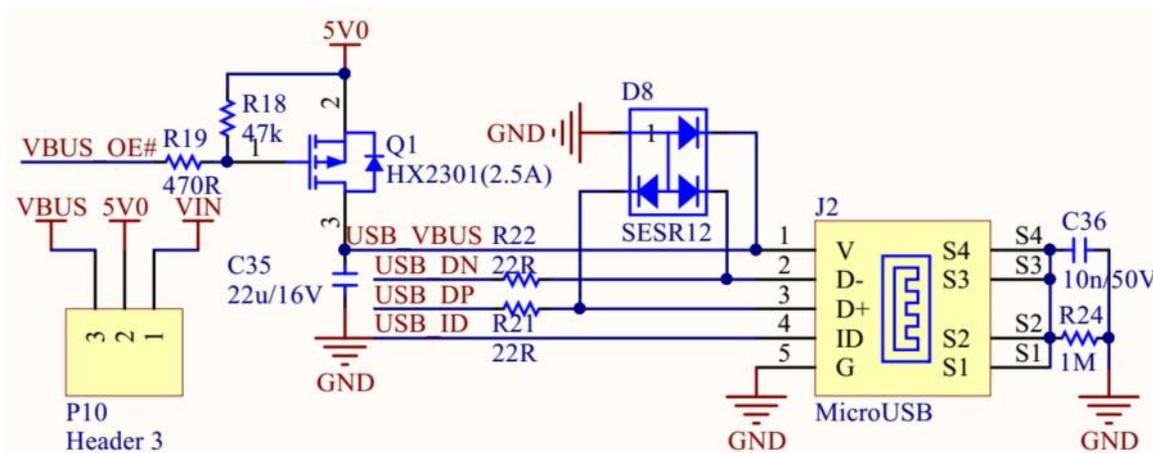


图 2-4 USB OTG 接口电路

2.4. SD 卡

GD32VF103 MCU 可以通过 SPI 总线协议与 micro SD 卡通信。RV-STAR 开发板上 micro SD 卡座默认不焊接，如果用户需要可自行焊接。此外，与 SD 卡连接的四根 SPI 信号同时引出到 PMOD2 扩展接口上。

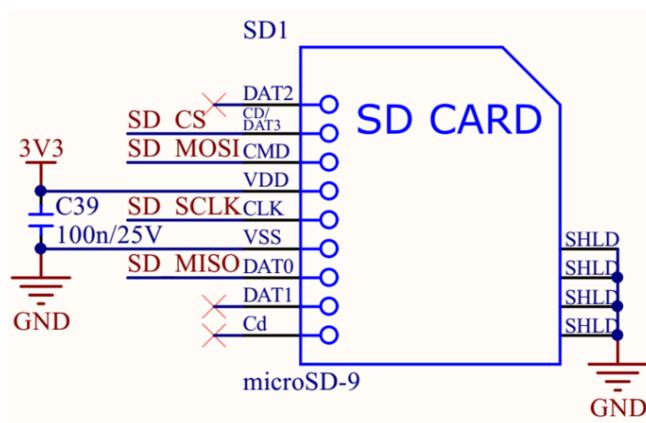


图 2-5 SD 卡电路

2.5. 指示与用户 LED

板载蓝色 LED D2 指示板载 LDO 电源稳压工作正常，开发板有电源输入；

板载绿色 LED D3 指示调试器虚拟串口收到数据；

板载黄色 LED D4 指示调试器虚拟串口发送数据；

板载 RGB LED 为用户可控的全彩共阳 LED，可供应用程序作为指示信息使用。

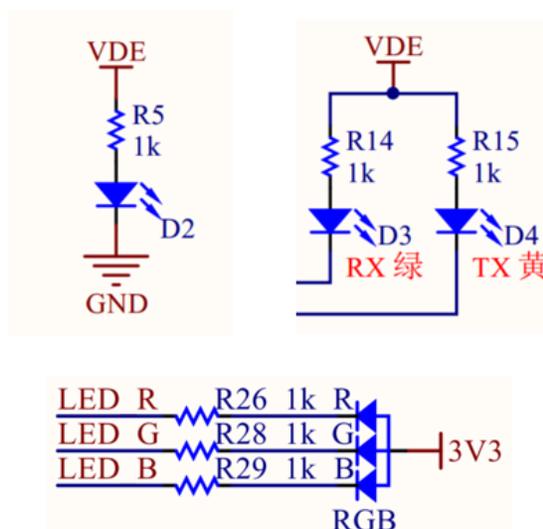


图 2-6 指示和用户 LED

2.6. 用户按键

RV-STAR 开发板提供了一个复位按键和一个唤醒按键，唤醒按键默认作为用户按键使用，电路如下图所示。注意唤醒按键有外部下拉，按下时为高电平。

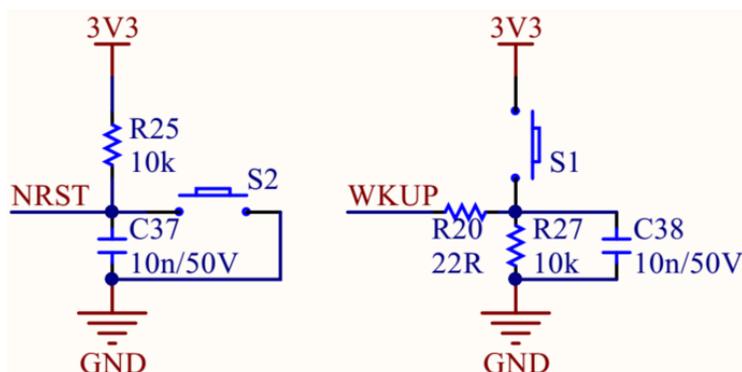


图 2-7 按键

3. 扩展接口和跳线

3.1. Arduino 兼容的扩展接口

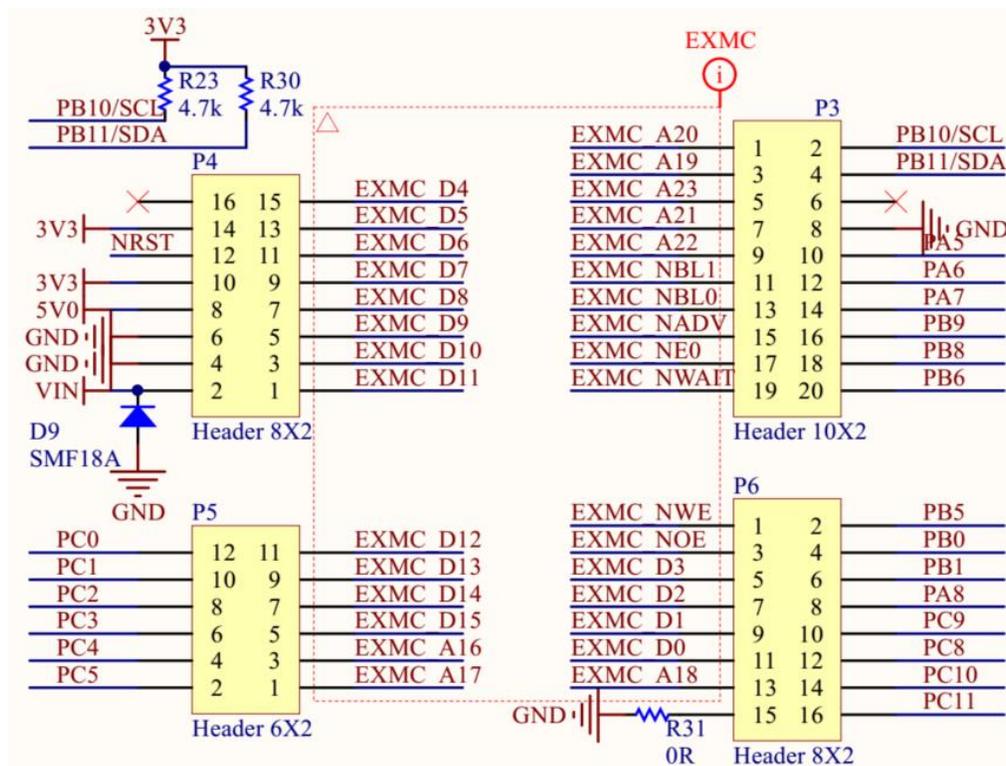


图 3-1 Arduino 兼容的扩展接口

P3-P6 的外侧扩展接口同 Arduino 物理接口相同，且引脚功能兼容，可直接连接 Arduino 接口的相关扩展板。下面分别列出了 P3-P6 接口上的信号与 MCU 对应的引脚及功能说明。

表 3-1 扩展接口 P3 引脚信号

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
Arduino	2	PB10/SCL	47	PB10	I2C1_SCL, USART2_TX, TIMER1_CH2
	4	PB11/SDA	48	PB11	I2C1_SDA, USART2_RX, TIMER1_CH3
	6	NC	/	/	/
	8	GND	/	电源地	GND

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
Arduino	10	PA5	30	PA5	SPI0_SCK, ADC01_IN5, DAC_OUT1
	12	PA6	31	PA6	SPI0_MISO, ADC01_IN06, TIMER2_CH0, TIMER0_BRKIN
	14	PA7	32	PA7	SPI0_MOSI, ADC01_IN7, TIMER2_CH1, TIMER0_CH0_ON
	16	PB9	96	PB9	TIMER3_CH3, I2C0_SDA, CAN0_TX
	18	PB8	95	PB8	TIMER3_CH2, I2C0_SCL, CAN0_RX
	20	PB6	92	PB6	I2C0_SCL, TIMER3_CH0, USART0_TX, CAN1_TX
EXMC	1	EXMC_A20	3	PE4	EXMC_A20
	3	EXMC_A19	2	PE3	EXMC_A19
	5	EXMC_A23	1	PE2	EXMC_A23
	7	EXMC_A21	4	PE5	EXMC_A21
	9	EXMC_A22	5	PE6	EXMC_A22
	11	EXMC_NBL1	98	PE1	EXMC_NBL1
	13	EXMC_NBL0	97	PE0	TIMER3_ETI, EXMC_NBL0
	15	EXMC_NADV	93	PB7	I2C0_SDA, TIMER3_CH1, EXMC_NADV
	17	EXMC_NE0	88	PD7	EXMC_NE0, USART1_CK
	19	EXMC_NWAIT	87	PD6	EXMC_NWAIT, USART1_RX

表 3-2 扩展接口 P4 引脚信号

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
Arduino	2	VIN	/	电源	5-9V 电源输入
	4	GND	/	电源地	GND
	6	GND	/	电源地	GND
	8	5V0	/	电源	板上 5V 输出, 跳线选择来源
	10	3V3	/	电源	板上 3.3V 输出
	12	NRST	14	复位	MCU 复位, 低有效
	14	3V3	/	电源	板上 3.3V 输出
	16	NC	/	/	/
EXMC	1	EXMC_D11	45	PE14	EXMC_D11, TIMER0_CH3
	3	EXMC_D10	44	PE13	EXMC_D10, TIMER0_CH2
	5	EXMC_D9	43	PE12	EXMC_D9, TIMER0_CH2_ON
	7	EXMC_D8	42	PE11	EXMC_D8, TIMER0_CH1
	9	EXMC_D7	41	PE10	EXMC_D7, TIMER0_CH1_ON
	11	EXMC_D6	40	PE9	EXMC_D6, TIMER0_CH0
	13	EXMC_D5	39	PE8	EXMC_D5, TIMER0_CH0_ON
	15	EXMC_D4	38	PE7	EXMC_D4, TIMER0_ETI

表 3-3 扩展接口 P5 引脚信号

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
Arduino	2	PC5	34	PC5	ADC01_IN15
	4	PC4	33	PC4	ADC01_IN14
	6	PC3	18	PC3	ADC01_IN13
	8	PC2	17	PC2	ADC01_IN12
	10	PC1	16	PC1	ADC01_IN11
	12	PC0	15	PC0	ADC01_IN10
EXMC	1	EXMC_A17	59	PD12	EXMC_A17, TIMER3_CH0, USART2_RTS

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
EXMC	3	EXMC_A16	58	PD11	EXMC_A16, USART2_CTS
	5	EXMC_D15	57	PD10	EXMC_D15, USART2_CK
	7	EXMC_D14	56	PD9	EXMC_D14, USART2_RX
	9	EXMC_D13	55	PD8	EXMC_D13, USART2_TX
	11	EXMC_D12	46	PE15	EXMC_D12, TIMER0_BRKIN

表 3-4 扩展接口 P6 引脚信号

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
Arduino	2	PB5	91	PB5	I2C0_SMBA, SPI2_MOSI, I2S2_SD, TIMER2_CH1, SPI0_MOSI, CAN1_RX
	4	PB0	35	PB0	ADC01_IN8, TIMER2_CH2, TIMER0_CH1_ON
	6	PB1	36	PB1	ADC01_IN9, TIMER2_CH3, TIMER0_CH2_ON
	8	PA8	67	PA8	USART0_CK, TIMER0_CH0, CK_OUT0, USBFS_SOF
	10	PC9	66	PC9	TIMER2_CH3
	12	PC8	65	PC8	TIMER2_CH2
	14	PC10	78	PC10	UART3_TX, USART2_TX, SPI2_SCK, I2S2_CK
	16	PC11	79	PC11	UART3_RX, USART2_RX, SPI2_MISO
EXMC	1	EXMC_NWE	86	PD5	EXMC_NWE, USART1_TX
	3	EXMC_NOE	85	PD4	EXMC_NOE, USART1_RTS
	5	EXMC_D3	82	PD1	EXMC_D3, CAN0_TX, OSCOUT
	7	EXMC_D2	81	PD0	EXMC_D2, CAN0_RX, OSCIN
	9	EXMC_D1	62	PD15	EXMC_D1, TIMER3_CH3

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
EXMC	11	EXMC_D0	61	PD14	EXMC_D0, TIMER3_CH2
	13	EXMC_A18	60	PD13	EXMC_A18, TIMER3_CH1
	15	*GND	/	*电源地	*焊接 R31 (0 欧电阻) 以连接 GND

3.2. PMOD 扩展接口

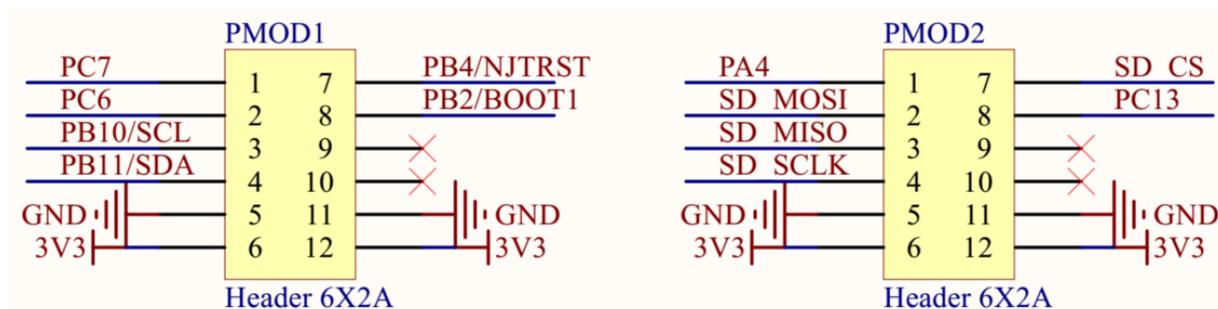


图 4-2 RV-STAR 开发板的 Pmod 扩展接口

RV-STAR 开发板提供了两组标准 PMOD 接口，可以方便的连接 PMOD 接口的相关扩展板。下面分别列出了 PMOD1、PMOD2 扩展接口上的信号与 MCU 对应的引脚及功能说明。

表 3-5 扩展接口 PMOD1 引脚信号

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
I2C	1	PC7	64	PC7	I2S2_MCK, TIMER2_CH1
	2	PC6	63	PC6	I2S1_MCK, TIMER2_CH0
	3	PB10/SCL	47	PB10	I2C1_SCL, USART2_TX, TIMER1_CH2
	4	PB11/SDA	48	PB11	I2C1_SDA, USART2_RX, TIMER1_CH3
	5	GND	/	电源地	GND
	6	3V3	/	电源	板上 3.3V 输出
	7	PB4/NJTRST	90	NJTRST	SPI2_MISO, TIMER2_CH0, PB4, SPI0_MISO

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
I2C	8	PB2/BOOT1	37	PB2, BOOT1	启动后可作普通 IO 使用
	9	NC	/	/	/
	10	NC	/	/	/
	11	GND	/	电源地	GND
	12	3V3	/	电源	板上 3.3V 输出

表 3-6 扩展接口 PMOD2 引脚信号

类型	位置	名字	MCU 管脚	默认功能	其他功能/说明
SPI	1	PA4	29	PA4	SPI0_NSS, USART1_CK, ADC01_IN4, DAC_OUT0, SPI2_NSS, I2S2_WS
	2	SD_MOSI	54	PB15	SPI1_MOSI, TIMER0_CH2_ON, I2S1_SD
	3	SD_MISO	53	PB14	SPI1_MISO, USART2_RTS, TIMER0_CH1_ON
	4	SD_SCLK	52	PB13	SPI1_SCK, USART2_CTS, TIMER0_CH0_ON, I2S1_CK, CAN1_TX
	5	GND	/	电源地	GND
	6	3V3	/	电源	板上 3.3V 输出
	7	SD_CS	51	PB12	SPI1_NSS, I2C1_SMBA, USART2_CK, TIMER0_BRKIN, I2S1_WS, CAN1_RX
	8	PC13	7	PC13	TAMPER-RTC
	9	NC	/	/	/
	10	NC	/	/	/
	11	GND	/	电源地	GND
	12	3V3	/	电源	板上 3.3V 输出

3.3. 跳线列表

RV-STAR 开发板的跳线连接选项与功能如下表所示。

表 3-7 跳线功能说明

跳线	说明	选项	默认	功能
P1	GND	2-1	√	P1 跳线用来连接调试器与 MCU，除标准 4 线 JTAG 接口外，还提供一组串口连接。断开此处连接便可分离 MCU 与调试器，使其各自可单独工作（调试器可用以调试其他设备，MCU 也可用其他调试器来进行调试）
	JTAG-TCK	4-3	√	
	JTAG-TDI	6-5	√	
	JTAG-TDO	8-7	√	
	JTAG-TMS	10-9	√	
	Reset	12-11	√	
	UART-TX	14-13	√	
	UART-RX	16-15	√	
	VDE-3v3	18-17	√	
P2	BOOT0 (必须选一)	4-6 L	√	从 Flash 启动
		4-2 H		启动方式取决于 BOOT1
	BOOT1	3-5 L	√	若 BOOT0=1 则从片上 Bootloader 启动，详情请参考 GD32VF103 用户手册
		3-1 H		若 BOOT0=1 则从片上 SRAM 启动
P7	RTC 电源 (必须选一)	2-1 3v3	√	不使用外接电池，使用板载 3v3 电源
		2-3 BAT		使用外接电池为 RTC 供电
P10	5V 电源 来源	2-1 VIN		5V 来自 Arduino 扩展接口 VIN。 当 VIN 输入为 5V 时，方能选择此连接方式
		2-3 VBUS	√	5V 来自 USB Type-C 接口

4. 快速开发 RV-STAR

Nuclei Studio 是由芯来科技公司所推出的支持其自研处理器产品（包括商业 Nuclei 系列处理器和开源蜂鸟处理器）的集成开发环境。它是基于开源 Eclipse 框架进行的实现，且充分与 Nuclei SDK（用于商业 Nuclei 系列处理器）和 HBird SDK（用于开源蜂鸟处理器）进行了整合，可以结合需求便捷的新建模板工程，以及修改工程设置选项。

此文档是面向已经有一定使用基础的用户，在此省略了下载安装、编译和调试运行等流程的详细介绍，主要指导用户如何创建 RV-STAR 开发板的简单示例并在 IDE 中运行，用户可以基于此快速上手 RV-STAR 开发板的应用。

注：

- ① 关于 Nuclei Studio 的下载安装、编译、调试运行等详细介绍，请参阅 RISC-V MCU 开放社区的《Nuclei Studio 的快速上手》文档（https://www.rvmcu.com/quickstart-doc-u-rvstar_nucleistudio_quickstart.html）
- ② 关于 Nuclei Studio 的编译链接选项设置等进阶内容，请参阅 RISC-V MCU 开放社区的《Nuclei Studio 的进阶学习》文档（https://www.rvmcu.com/quickstart-doc-u-rvstar_nucleistudio_advanced.html）。

4.1. 新建 helloworld 示例工程

下载 Nuclei Studio 后，直接双击 NucleiStudio 文件夹下面的可执行文件 eclipse.exe，便可启动 Nuclei Studio，第一次打开需要配置工作区间路径。

打开后在菜单栏选择“File→New → C/C++ Project”，在弹窗中按照以下内容设置各页面：

- 输入工程名，并选择“Nuclei SDK Project For GD32VF103 SoC”，点击“Next”。
- 新打开的页面中切换“Board”选项为“gd32vf103v_rvstar”，点击“Next”。
- 新打开的页面仅需修改“Project Example”为“baremetal_helloworld”，其它选项使用

默认选项即可，点击“Next”直到“Finish”。

经过以上步骤，RV-STAR 的 helloworld 示例工程项目便创建完成了，IDE 界面如图 4-1 所示。

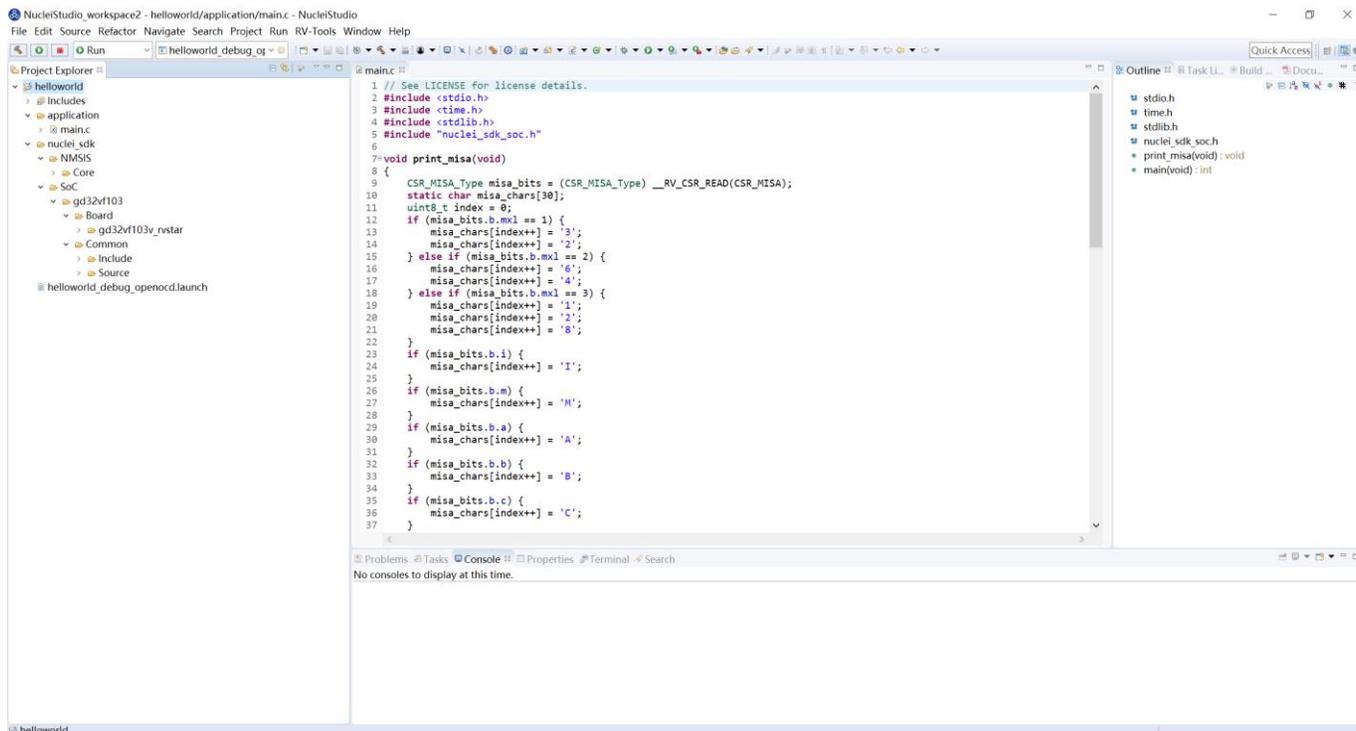


图 4-1 新建 helloworld 示例工程

4.2. 运行 helloworld 示例

编译 helloworld 示例，鼠标右击工程选择“Build Project”，编译成功后会出现如图 4-2 所示内容。

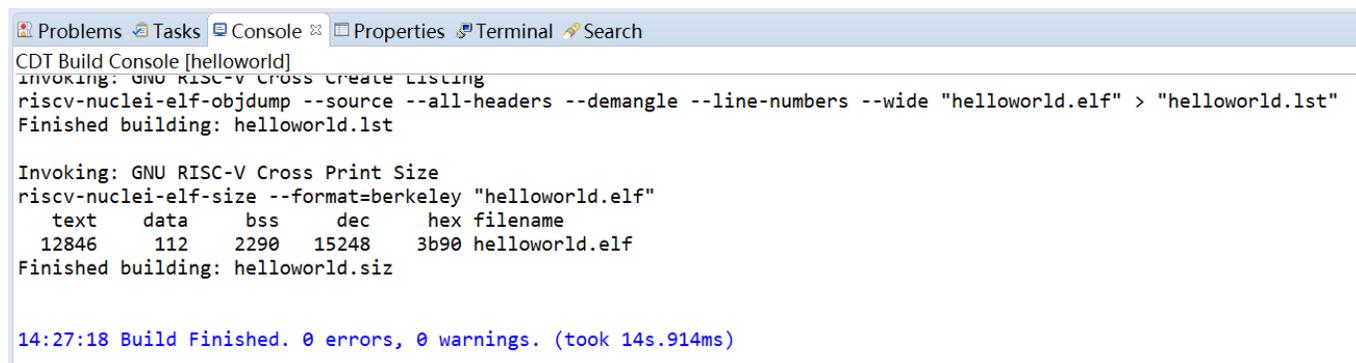


图 4-2 编译 helloworld 示例

示例程序编译成功后，便可将编译好的可执行文件下载到 RV-STAR 开发板运行。首先需要进行硬件的连接，通过一根 USB Type-C 电缆将 RV-STAR 开发板与 PC 进行连接，线缆连接好后，板上的蓝色电源灯会点亮，如下图所示。

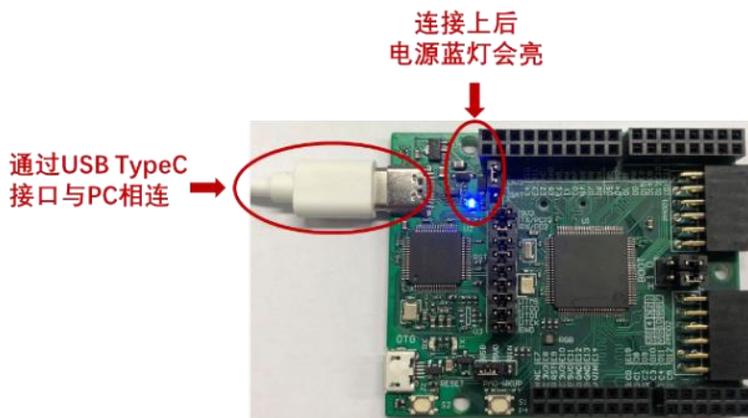


图 4-3 调试连接

注：

- ① 如果第一次使用 RV-STAR 开发板，为了使 PC 的 Windows 系统能识别到 RV-STAR 板载调试器（同蜂鸟调试器），需要安装驱动 HBird-Driver.exe，双击即可完成此驱动的安装。Linux 系统环境下的驱动安装请参考《Nuclei_FPGA_DebuggerKit_Intro》文档。驱动下载和文档获取请访问（<https://www.nucleisys.com/developboard.php>），如图 4-4 标识处。
- ② 由于板载调试器还包含了“将 MCU 的 UART 转换成 USB 输出”功能，因此将 RV-STAR 开发板与 PC 连接后，将被主机识别成一个 COM 串口，可通过 PC 的设备管理器查看该串口端口号。

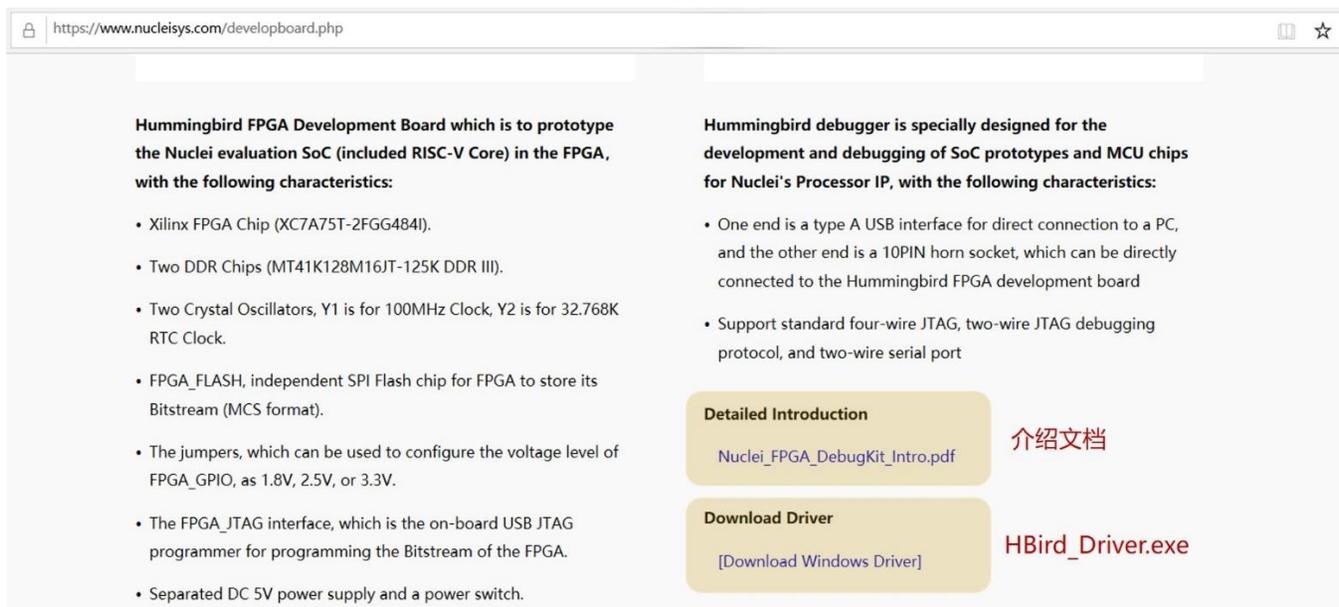


图 4-4 调试器 Windows 驱动下载及文档获取

驱动安装完成后，打开 Nuclei Studio 自带的串口工具。在菜单栏选择“Window—>Show View—>Terminal”，再按下“Ctrl+Shift+Alt+T”打开串口设置页面，修改“Choose Terminal”为“Serial Terminal”，再修改“Serial Port”为对应的 COM 号，“Baud rate”为“115200”，其它选项使用默认设置即可。

鼠标右击工程选择“Run as—>Local C/C++ Appliaction”，即可开始运行示例工程，示例运行后打印结果如图 4-5。

```

Problems Tasks Console Properties Terminal Search
COM27
NuclNuclei SDK Build Time: Nov 12 2020, 14:27:16
Download Mode: FLASHXIP
CPU Frequency 108778846 Hz
MISA: 0x40901105
MISA: RV32IMACUX
0: Hello World From Nuclei RISC-V Processor!
1: Hello World From Nuclei RISC-V Processor!
2: Hello World From Nuclei RISC-V Processor!
3: Hello World From Nuclei RISC-V Processor!
4: Hello World From Nuclei RISC-V Processor!
5: Hello World From Nuclei RISC-V Processor!
6: Hello World From Nuclei RISC-V Processor!
7: Hello World From Nuclei RISC-V Processor!
8: Hello World From Nuclei RISC-V Processor!
9: Hello World From Nuclei RISC-V Processor!
10: Hello World From Nuclei RISC-V Processor!
11: Hello World From Nuclei RISC-V Processor!
12: Hello World From Nuclei RISC-V Processor!
13: Hello World From Nuclei RISC-V Processor!
14: Hello World From Nuclei RISC-V Processor!
15: Hello World From Nuclei RISC-V Processor!
16: Hello World From Nuclei RISC-V Processor!
17: Hello World From Nuclei RISC-V Processor!
18: Hello World From Nuclei RISC-V Processor!
19: Hello World From Nuclei RISC-V Processor!

```

图 4-5 运行 helloworld 示例

4.3. 更多示例

关于 RV-STAR 开发板，芯来科技提供了丰富的实验示例和详实的配套介绍文档，可在 Nuclei Board Labs 中获取。用户通过替换 helloworld 示例中 application 文件夹内容为对应实验文件夹内容，即可编译运行 Nuclei Board Labs 中的示例。

注：

Nuclei Board Labs，基于 Nuclei SDK 所实现的 Nuclei 开发板的实验教程，提供了手把手的板级实验例程，极大方便了 RV-STAR 开发板使用者的快速入门，会持续进行更新。

- 源码

Github: <https://github.com/Nuclei-Software/nuclei-board-labs>

Gitee: <https://gitee.com/Nuclei-Software/nuclei-board-labs>

- 文档

https://doc.nucleisys.com/nuclei_board_labs/

■ Demo1: running_led

实验目的：通过 GPIO 读取用户按键状态，并控制 RGB LED；串口通信。

演示效果：每按一次板上的 PA0-WKUP 用户按键，RGB 灯会切换一下颜色，按照“红-绿-蓝-白”的亮色顺序循环进行，调试信息将通过串口传输至 PC 端串口调试助手显示出来，如图 4-6 所示。

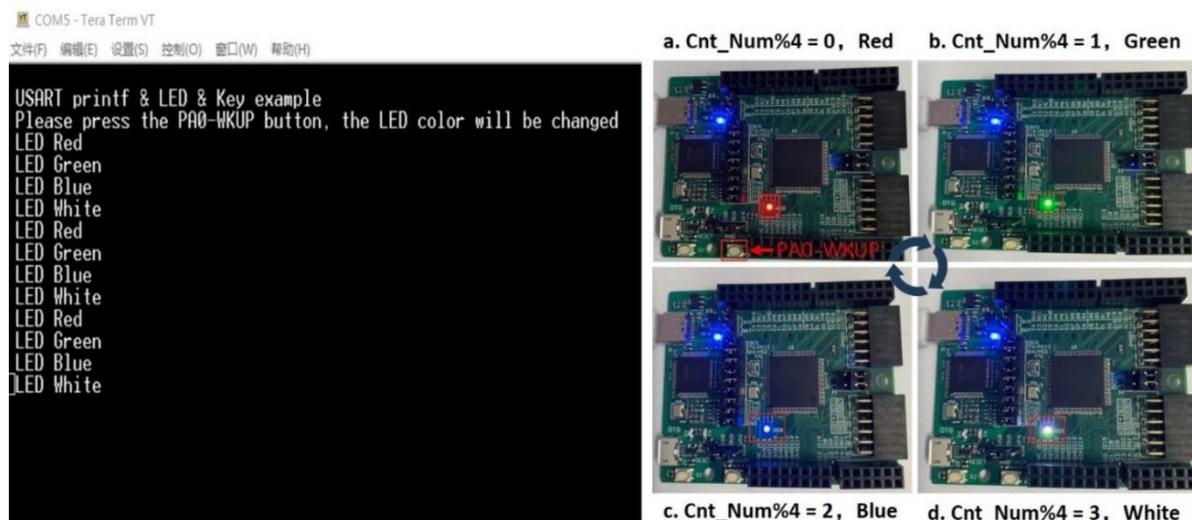


图 4-6 Demo1 演示效果

■ Demo2: timer_pwm

实验目的：通过 Timer 来产生 PWM 信号，并将其输出到 IO 引脚来控制 RGB LED。

演示效果：RGB LED 颜色由“红-绿-蓝”依次变化，变化效果如同呼吸灯。

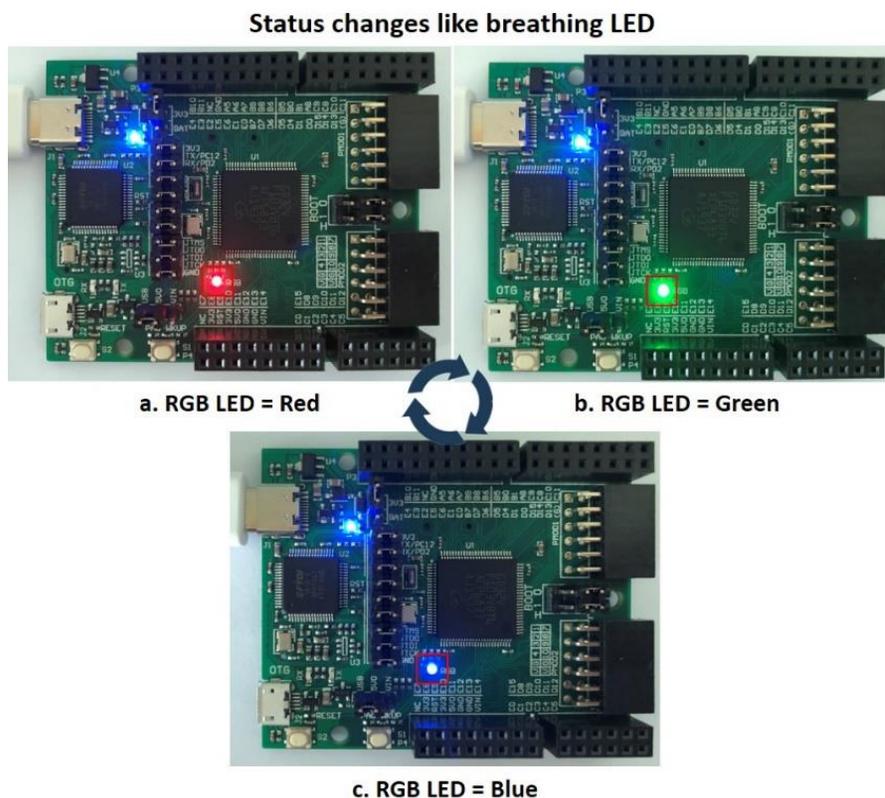


图 4-7 Demo2 演示效果

■ Demo3: exti_key_interrupt

实验目的：通过用户按键来产生一个外部中断；通过 GPIO 控制 RGB LED。

演示效果：第一次按下按键时，RGB LED 亮红色，再次按下按键，RGB LED 熄灭，实现按键来控制灯的开关功能。

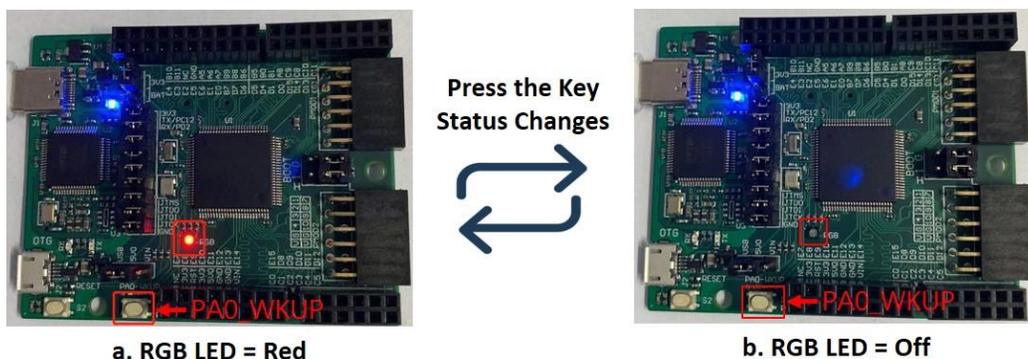


图 4-8 Demo3 演示效果

■ Demo4: nesting_of_interrupts

实验目的：通过 Timer 来产生一个定时器中断；通过用户按键来产生一个外部中断；实现中断嵌套，设置用户按键中断具备更高优先级；通过 GPIO 控制 LED。

演示效果：RGB LED 颜色每隔 5 秒在红色和绿色间切换，若按下用户按键，RGB LED 切换为蓝色，1 秒后切换为之前的颜色。

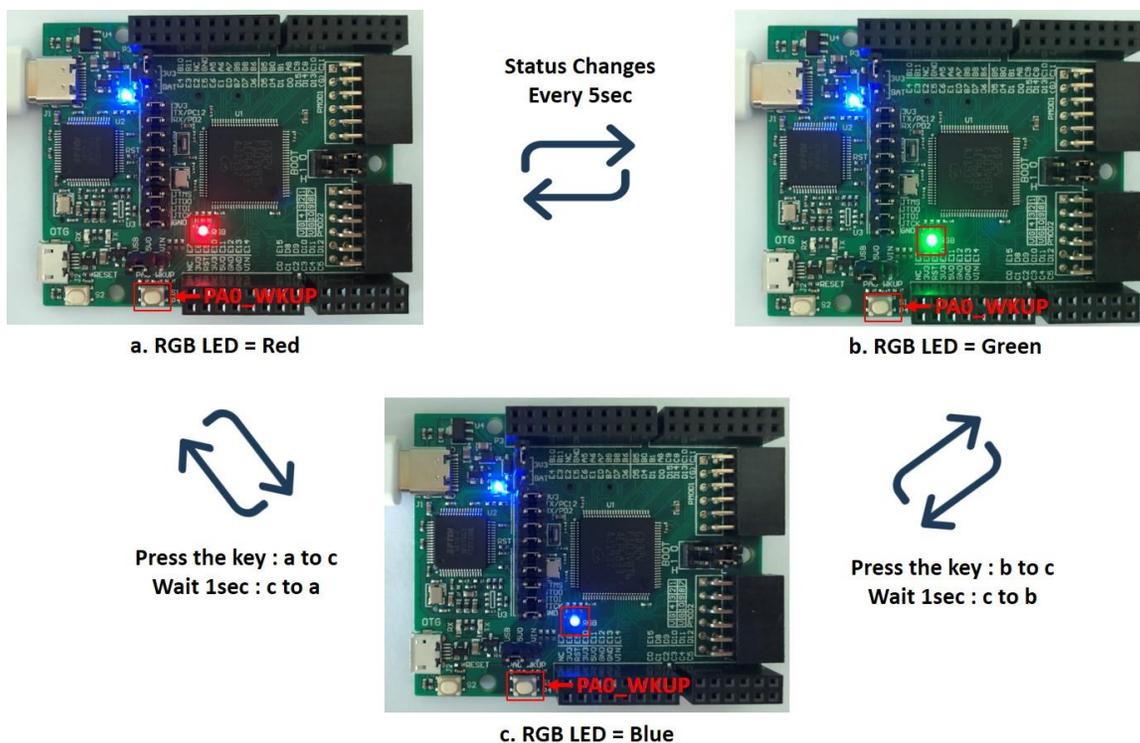


图 4-9 Demo4 演示效果

5. 参考文档

以下参考文档均可从 www.rvmcu.com 网站获取：

- RV-STAR 电路原理图
- GD32VF103 用户手册
- GD32VF103 数据手册
- Bumblebee 用户手册
- Bumblebee 数据手册