

开源开放的RISC-V嵌入式软件平台探索

方华启
芯来科技
2020.07

主要内容

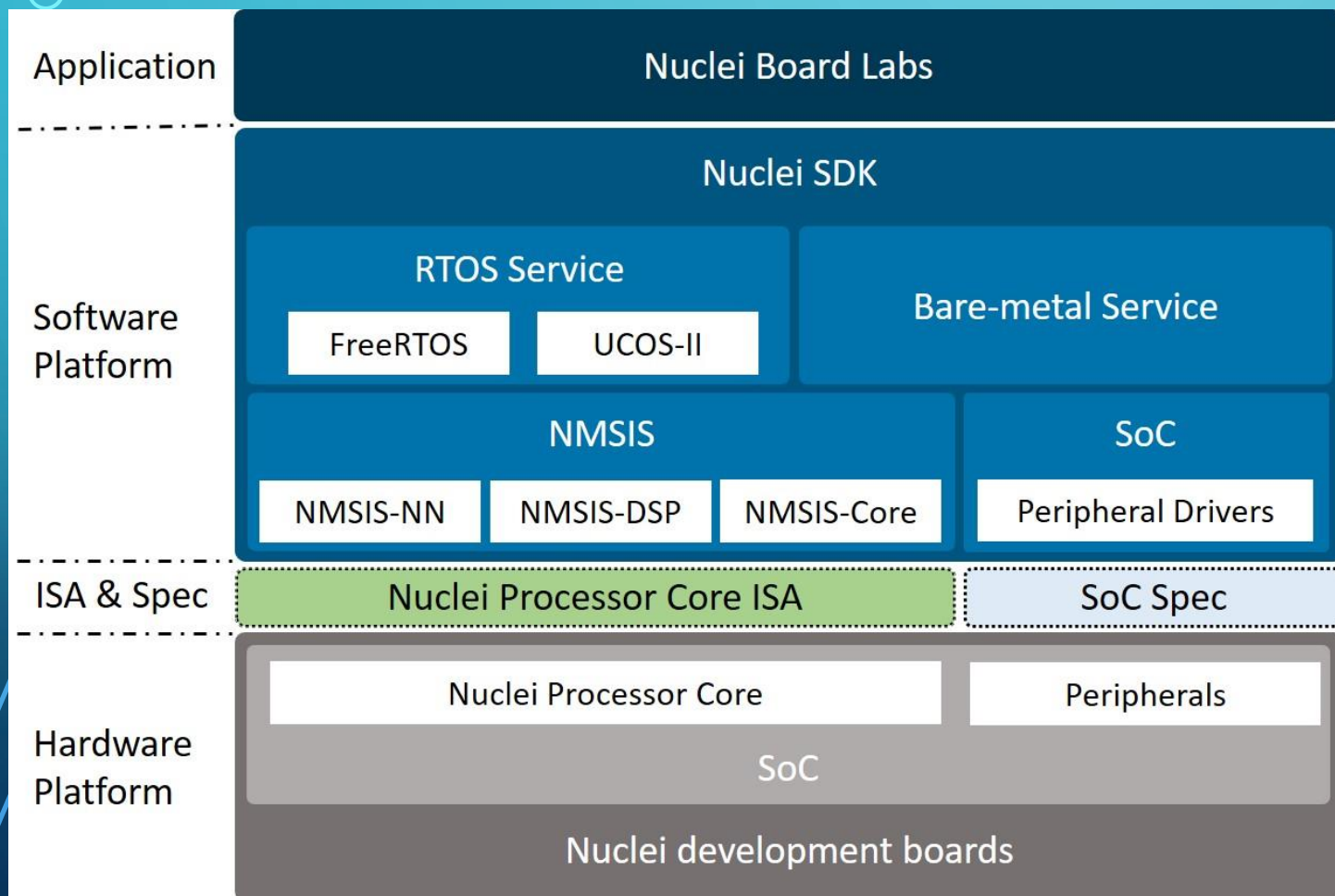
- RISC-V软件生态
- NMSIS软件基础平台
- Nuclei SDK软件开发平台
- RV-STAR与Board Labs

RISC-V软件生态现状

- RISC-V指令集架构成熟稳定，硬件生态蓬勃发展
- RISC-V基础开发工具日趋完善
 - 开源工具链(GNU/LLVM), GNU MCU Eclipse, QEMU, SPIKE
 - 商用集成开发环境(Segger、IAR等)
 - 事实上标准的OpenSBI (Open Supervisor Binary Interface)
 - Linux Kernel
- RISC-V基础软件平台薄弱且碎片化
 - 没有类似于ARM的CMSIS这样的嵌入式软件接口标准
 - 各家管好各家事，缺少统一的底层软件接口

芯来基础软件平台探索

NUCLEI SOFTWARE PLATFORM



Nuclei Board Labs

- [Online Documentation](#)
- [Source Code](#)

Nuclei Software Development Kit (Nuclei SDK)

- [Online Documentation](#)
- [Source Code](#)

Nuclei Microcontroller Software Interface Standard (NMSIS)

- [Online Documentation](#)
- [Source Code](#)

Nuclei RISC-V ISA

- [Online Documentation](#)

[Nuclei Development Boards](#)

NUCLEI ISA标准规范

NUCLEI ISA SPEC



Nuclei ISA Spec

1.5.0

NUCLEI RISC-V ISA DOCUMENTATION

- 1. Introduction
- 2. Nuclei RISC-V Instruction Set Overview
- 3. Nuclei RISC-V Privileged Architecture
- 4. Exception Handling in Nuclei processor core
- 5. NMI Handling in Nuclei processor core
- 6. Interrupt Handling in Nuclei processor core
- 7. Nesting of Interrupt, NMI and Exception
- 8. TIMER Unit Introduction
- 9. PLIC Unit Introduction
- 10. ECLIC Unit Introduction
- 11. ECLIC and PLIC Connection Diagram

Nuclei RISC-V ISA

Nuclei RISC-V ISA Documentation

- 1. Introduction
- 2. Nuclei RISC-V Instruction Set Overview
 - 2.1. RISC-V ISA Modules supported by Nuclei Core
 - 2.2. Nuclei Processor Core Classes: N, NX and UX
- 3. Nuclei RISC-V Privileged Architecture
 - 3.1. RISC-V Privileged Architecture supported by Nuclei Core
 - 3.2. Privilege Modes
 - 3.3. Debug Mode
 - 3.4. Machine Sub-Mode added by Nuclei
- 4. Exception Handling in Nuclei processor core
 - 4.1. Exception Overview
 - 4.2. Exception Masking
 - 4.3. Priority of Exception
 - 4.4. Entering Exception Handling Mode
 - 4.4.1. Update the Machine Sub-Mode
 - 4.5. Exit the Exception Handling Mode
 - 4.5.1. Update the Machine Sub-Mode
 - 4.6. Exception Service Routine
 - 4.7. Exception Nesting
- 5. NMI Handling in Nuclei processor core
 - 5.1. NMI Overview
 - 5.2. NMI Masking
 - 5.3. Entering NMI Handling Mode
 - 5.3.1. Execute from the PC Defined by mnvec
 - 5.3.2. Update the CSR mcause
 - 5.3.3. Update the Privilege Mode

- ❑ 开放在线的Nuclei RISC-V ISA文档
- ❑ 完备的Nuclei RISC-V特性文档
- ❑ 完备的Nuclei Processor Core Unit文档
- ❑ 便于开发者快速了解入门Nuclei RISC-V处理器特性
- ❑ 关于更多详细内容请访问

<https://doc.nucleisys.com/nuclei-spec/>

关于NMSIS

ABOUT NMSIS

- NMSIS全称为Nuclei Microcontroller Software Interface Standard
- NMSIS是为芯来科技RISC-V处理器定义的厂商无关的硬件抽象层
 - NMSIS定义了通用软件接口并提供持续的处理器设备支持
 - NMSIS提供了简洁的处理器和外设的软件访问接口API
 - 采用NMSIS框架，可以提供软件的可复用性，缩短RISC-V微处理器开发者的学习时间，大大缩短新处理器设备的上市时间
- NMSIS是基于开源软件CMSIS，针对Nuclei RISC-V处理器进行深度定制优化开发，提供完备的代码以及在线文档，并采用Apache 2.0 License进行发布

NMSIS组件

NMSIS COMPONENTS

- **NMSIS-Core**

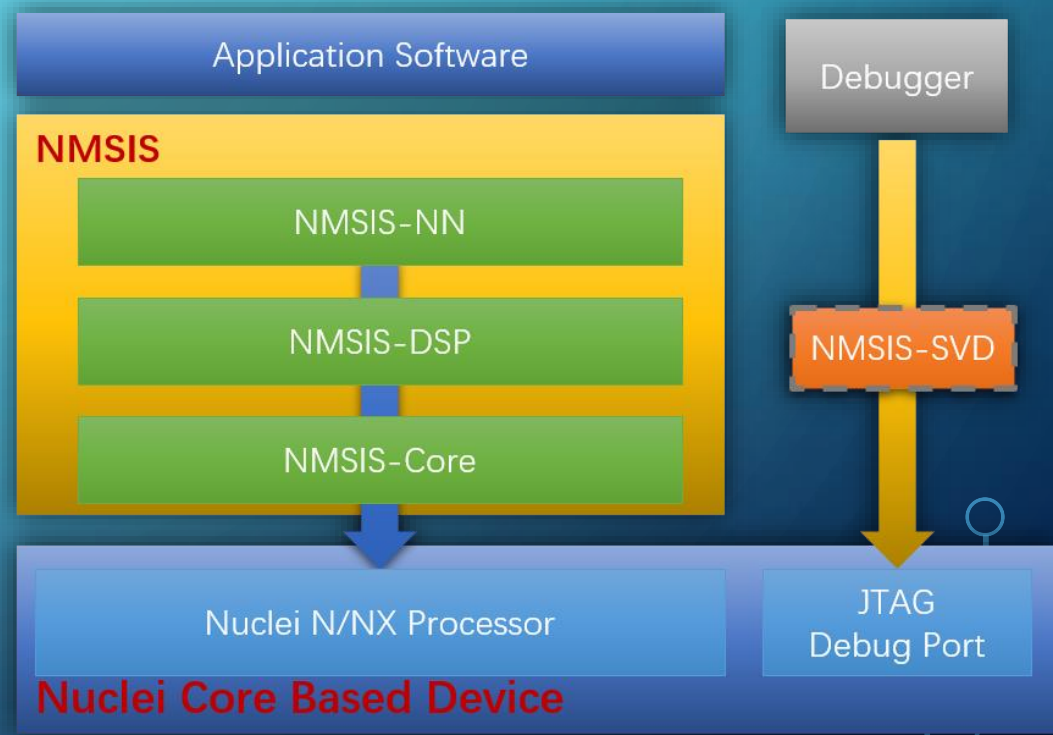
为RISC-V处理器定义的处理器核心和外设访问的标准化API

- **NMSIS-DSP**

专为RISC-V处理器优化的DSP library，支持Q7，Q15以及Q31定点和单精度浮点。

- **NMSIS-NN**

专为RISC-V处理器优化的高效的卷积神经网络(NN) API，支持卷积，全连接，池化，激活等关键层

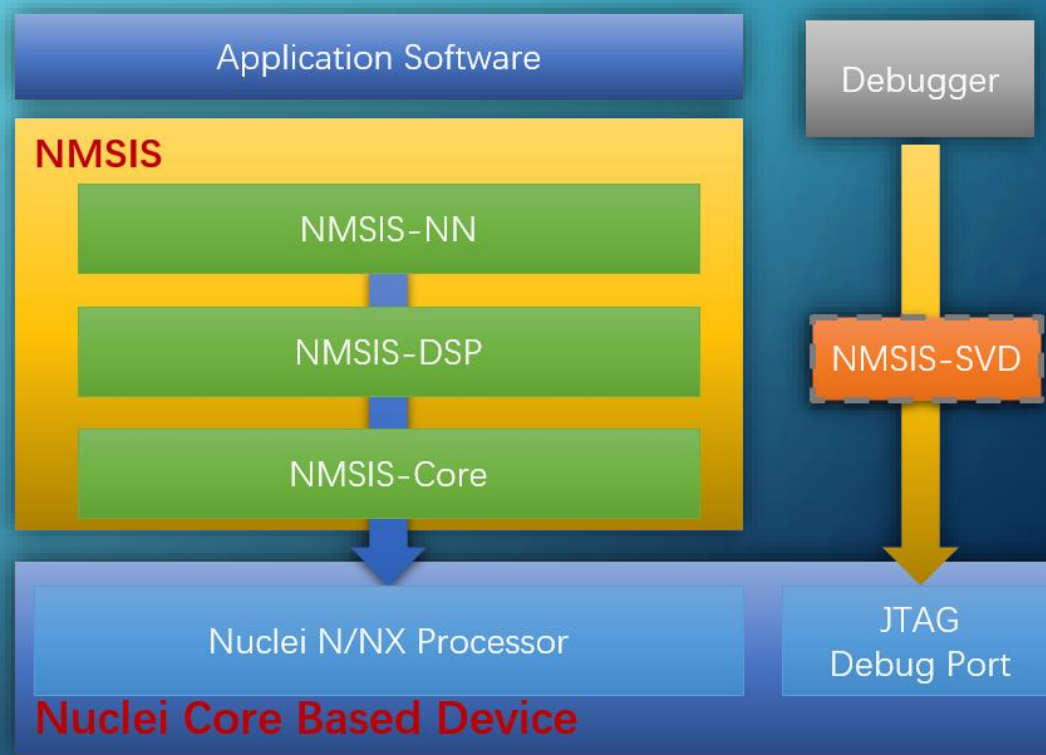


NMSIS设计

NMSIS DESIGN

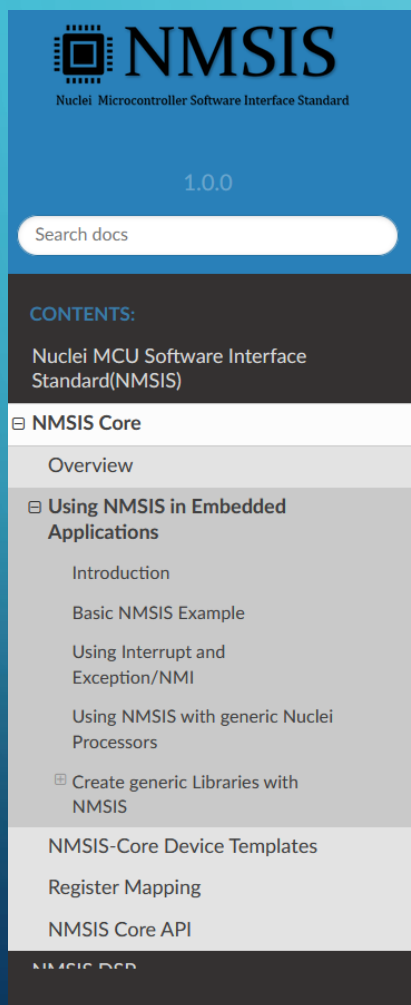
NMSIS设计初衷是为了标准化Nuclei RISC-V系列处理器核心的软件开发，从而使不同的开发工具和微处理器的软件开发更加容易并且便于移植。

- NMSIS可以有效减少学习成本，开发成本，缩短上市时间。开发者通过标准化的软件接口可以更快的开发软件
- 一致化的软件接口有效的提高了软件的可移植性以及可复用性，通用的library也可以提供一致性的软件框架
- 提供编译器无关的软件层，也为未来支持不同的编译器提供可能性，便于开发者专注于软件核心功能开发
- 更多详细内容请访问 <https://doc.nucleisys.com/nmsis/>



NMSIS文档

NMSIS ONLINE DOC



NMSIS
Nuclei Microcontroller Software Interface Standard

1.0.0

Search docs

CONTENTS:

Nuclei MCU Software Interface Standard(NMSIS)

▢ NMSIS Core

- Overview
- ▢ Using NMSIS in Embedded Applications
 - Introduction
 - Basic NMSIS Example
 - Using Interrupt and Exception/NMI
 - Using NMSIS with generic Nuclei Processors
- ▢ Create generic Libraries with NMSIS
- NMSIS-Core Device Templates
- Register Mapping
- NMSIS Core API

peripherals may be available.

- [Interrupts and Exceptions](#) can be accessed with standardized symbols and functions for the **ECLIC** are provided.
- [CPU Intrinsic Functions](#) allow to access special instructions, for example for activating sleep mode or the NOP instruction.
- [Intrinsic Functions for SIMD Instructions](#) provide access to the DSP-oriented instructions.
- [Systick Timer\(SysTimer\)](#) function to configure and start a periodic timer interrupt.
- [Core CSR Register Access](#) function to access the core csr registers.
- [Cache Functions](#) to access the I-CACHE and D-CACHE unit
- [FPU Functions](#) to access the Floating point unit.
- [PMP Functions](#) to access the Physical Memory Protection unit
- [Version Control](#) which defines NMSIS release specific macros.
- [Compiler Control](#) is compiler agnostic `#define` symbols for generic C/C++ source code

NMSIS-Core Device Files(Silicon Vendor)

startup_<device>.S

NMSIS Device Startup & Interrupt Vectors

gcc_<device>.ld

NMSIS Linker Script

intexc_<device>.S

NMSIS Device Interrupt & Exception Handling

system_<device>.c/h

NMSIS System & Clock Configuration

<device>.c/h

NMSIS Device Peripheral Access

NUCLEI SDK设计

NUCLEI SDK DESIGN

General Application
using only Nuclei Core
Related Resource

HBird Application using
HummingBird SoC and
Board Resource

GD32VF103 Application
using GD32VF103 SoC
and Board Resource

Nuclei SDK Application

FreeRTOS

UCOS-II

RT-Thread

RTOS Service

Bare-metal Service

Core
Startup

Exception/Interrupt
Handling

Core/CSR/DSP/FPU
/ECLIC/TIMER API

Nuclei Core API provided via NMSIS-Core

NMSIS-NN
NMSIS-DSP

Various APIs
in SoC
Firmware
Package

HummingBird
Evaluation Kit

GD32VF103
Evaluation Kit

GD32VF103
RVStar Kit

Nuclei Processor Based Boards

- ❑ Nuclei Software Development Kit (SDK) 是专为基于Nuclei处理器内核的SoC开发的开源软件平台
- ❑ Nuclei SDK以NMSIS为基础进行开发, 提供NMSIS上的所有功能, 包括NMSIS-Core, NMSIS-NN, NMSIS-DSP
- ❑ Nuclei SDK提供了裸机(Baremetal)或者实时操作系统(FreeRTOS, UCOS-II以及RT-Thread)开发环境
- ❑ 提供完备的在线用户手册, 方便开发者进行开发
- ❑ 不仅提供了基于Makefile的开发环境, 还提供了Segger embedded studio和PlatformIO IDE的IDE开发环境
- ❑ 支持蜂鸟FPGA开发板以及RV-STAR开发板
- ❑ 支持Windows和Linux开发环境
- ❑ 更多详细内容请访问
https://doc.nucleisys.com/nuclei_sdk/

NUCLEI SDK文档

NUCLEI SDK ONLINE DOC



SDK

Nuclei embedded Software Development Kit

0.2.5-dev

CONTENTS:

- Overview
- Quick Startup**
 - Setup Tools and Environment
 - Get and Setup Nuclei SDK
 - Build, Run and Debug Sample Application
 - Create helloworld Application
 - Advanced Usage
- Developer Guide
- Contributing
- Design and Architecture
- Changelog
- FAQ
- License
- Glossary
- Appendix

```
/usr/bin/make
/bin/rm
hqfang@hqfang-ubuntu [13:01:38]:~/workspace/software/nuclei-sdk
$ make help
make -C application/baremetal/helloworld help
make[1]: 进入目录"/home/hqfang/workspace/software/nuclei-sdk/application/baremetal/helloworld"
Nuclei N/NX-series RISC-V Embedded Processor Software Development Kit
== Make variables used in Nuclei SDK ==
SOC:      Select SoC built in Nuclei SDK, will select hbird by default
BOARD:    Select SoC's Board built in Nuclei SDK, will select hbird_eval by default
CORE:     Not required for all SoCs, currently only hbird require it, n307fd by default
DOWNLOAD: Not required for all SoCs, use ilm by default, optional flashxip/ilm/flash
V:        V=1 verbose make, will print more information, by default V=0
== How to Use with Make ==
1. Build Application:
all [PROGRAM=flash/flashxip/ilm]
   Build a software program to load with the debugger.
2. Upload Application to Board using OpenOCD and GDB:
upload [PROGRAM=flash/flashxip/ilm]
   Launch OpenOCD to flash your program to the on-board Flash.
3:(Option 1) Debug Application using OpenOCD and GDB
3.1: run_openocd
3.2: run_gdb [PROGRAM=flash/flashxip/ilm]
   Step 1: Launch OpenOCD for Debugger connection: make run_openocd
   Step 2: Launch GDB to connect openocd server, you can set breakpoints using gdb and debug it.
   If you want to load your application, you need to run load in gdb command terminal
   to load your program, then use gdb to debug it.
3:(Option 2) Debug Application using OpenOCD and GDB
debug [PROGRAM=flash/flashxip/ilm]
   Launch GDB and OpenOCD to debug your application on-board, you can set breakpoints using gdb and debug it.
   If you want to load your application, you need to run load in gdb command terminal
```

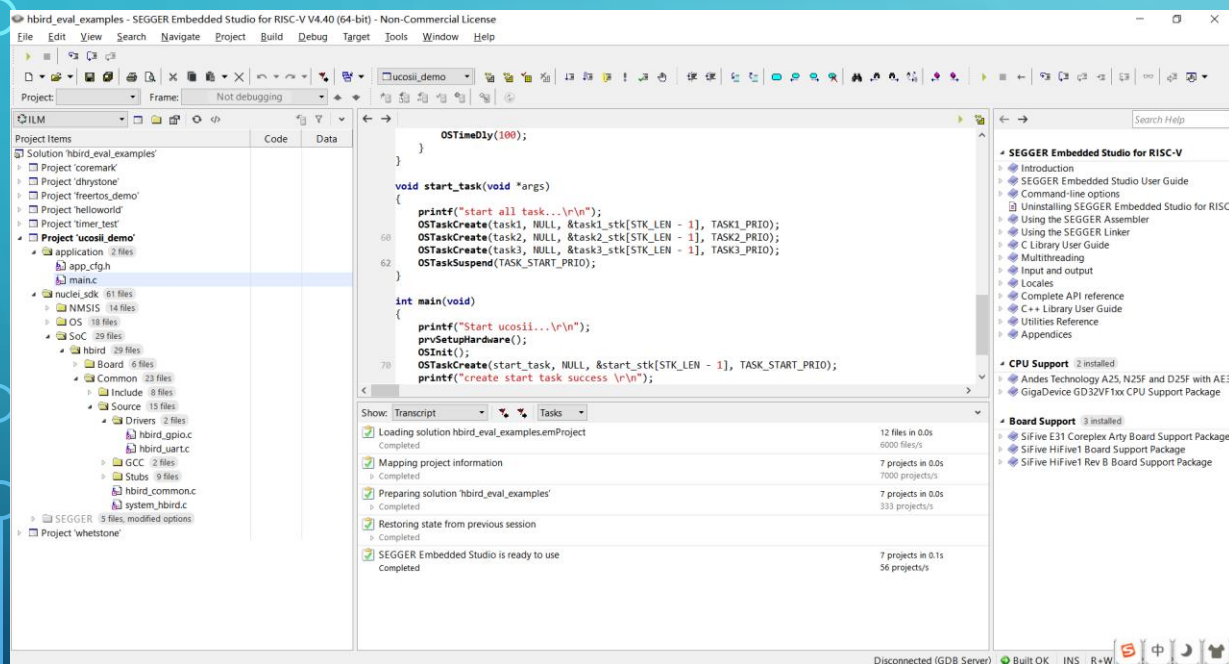
Setup Build Environment for Nuclei SDK in Linux Bash

Note

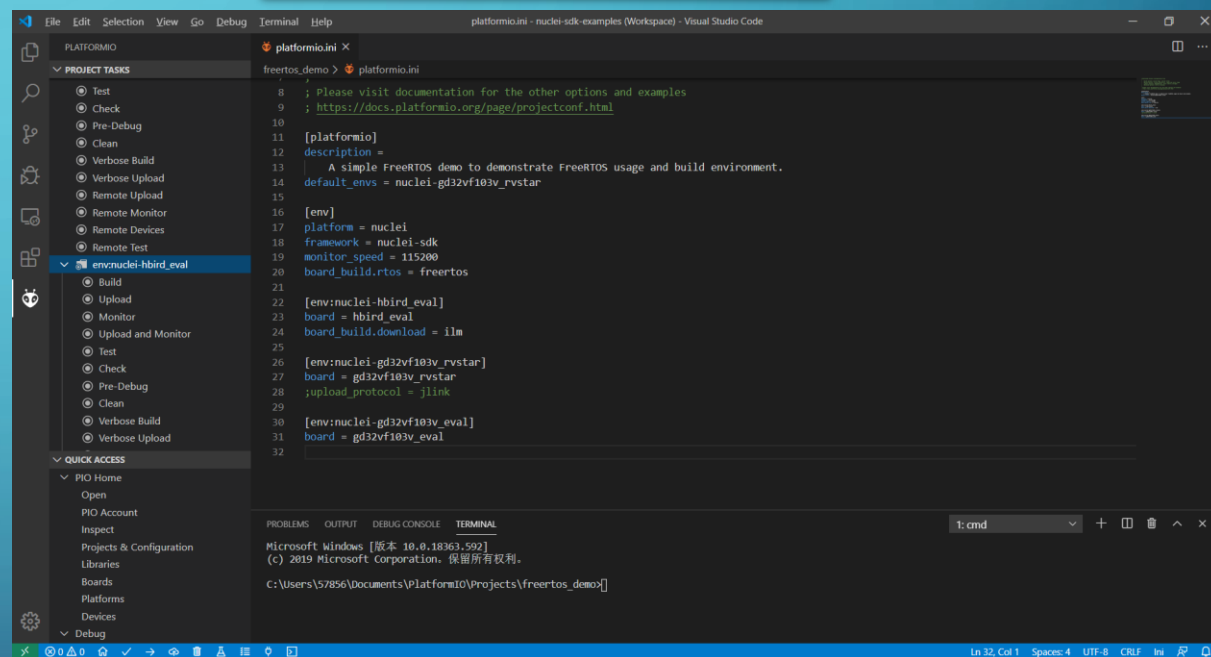
- Only first line `setup.bat` or `source setup.sh` are required before build, run or debug application. The `setup.bat` and `setup.sh` are just used to append Nuclei RISC-V GCC Toolchain, OpenOCD and Build-Tools binary paths into environment variable **PATH**
- line 2-4 are just used to check whether build environment is setup correctly, especially the **PATH** of Nuclei Tools are setup correctly, so we can use the `riscv-nuclei-elf-xxx`, `openocd`, `make` and `rm` tools
- If you know how to append Nuclei RISC-V GCC Toolchain, OpenOCD and Build-Tools binary paths to **PATH** variable in your OS environment, you can also put the downloaded Nuclei Tools as you like, and no need to run `setup.bat` or `source setup.sh`

NUCLEI SDK的第三方软件支持

<https://www.segger.com/>



<https://platformio.org/>



NUCLEI SDK +



Nuclei Studio

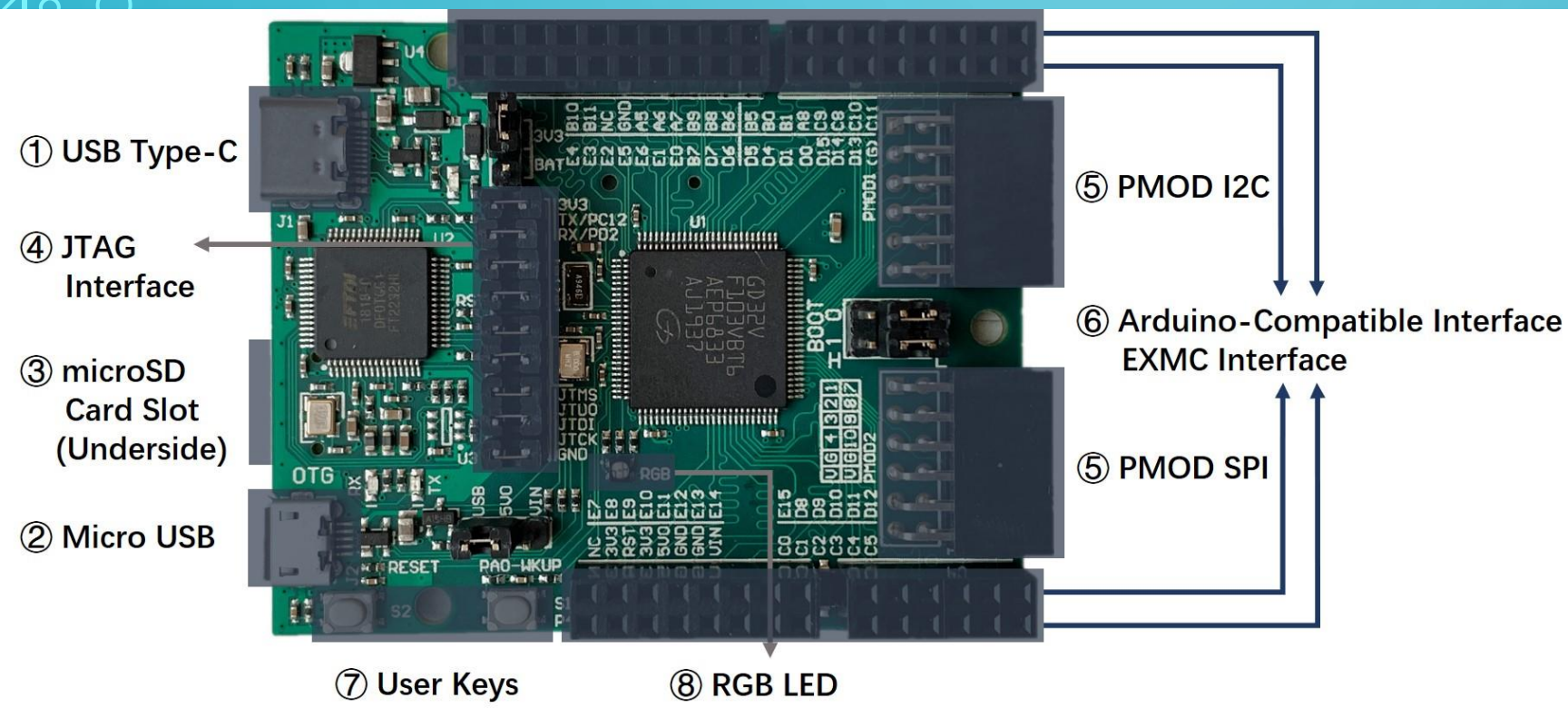


TencentOS
Tiny



芯来硬件开发平台

NUCLEI DEVELOPMENT BOARDS



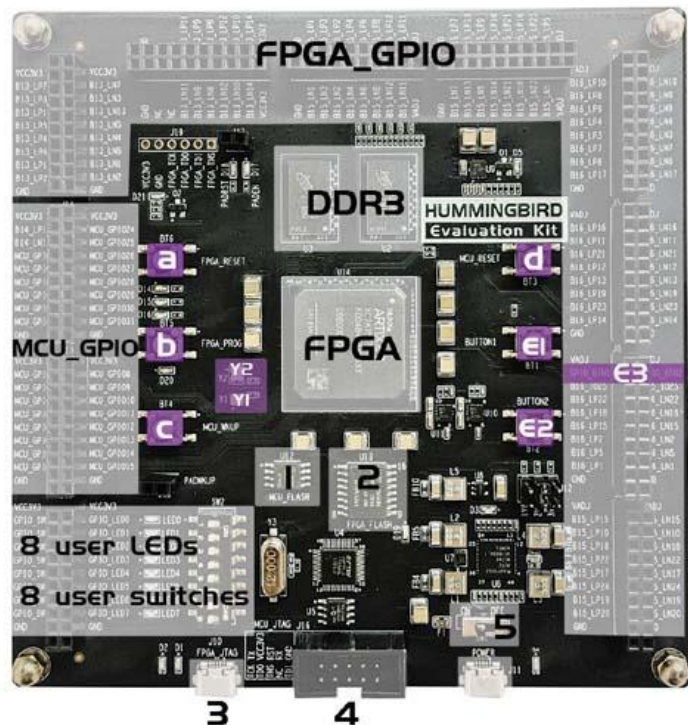
- ❑ 基于GD32VF103 MCU (Nuclei Bumblebee(RV32IMAC)内核)的RISC-V开发板
- ❑ 提供USB-Type-C板载调试器
- ❑ Arduino兼容外设接口
- ❑ PMOD兼容外设接口
- ❑ 复位, 唤醒按键以及RGB LED
- ❑ 提供Micro SD卡接口, USB接口

RV-STAR开发板

芯来硬件开发平台

NUCLEI DEVELOPMENT BOARDS

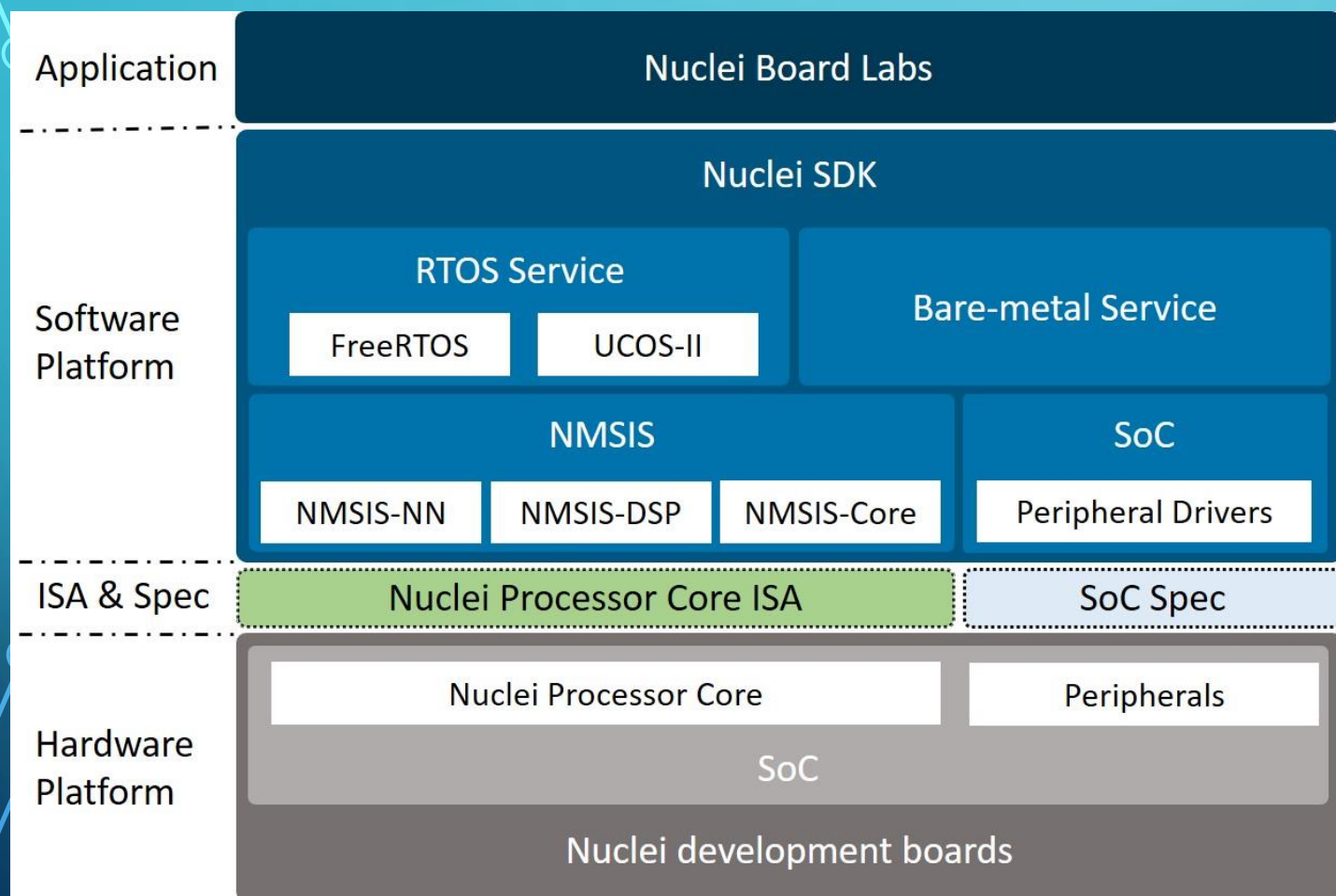
- a: FPGA_RESET
- b: FPGA_PROG
- c: MCU_WKUP
- d: MCU_RESET
- e1: User button 1
- e2: User button 2
- e3: User button header
- Y1: GCLK
- Y2: RTC_CLK
- 1: MCU_FLASH
- 2: FPGA_FLASH
- 3: FPGA_JTAG
- 4: MCU_JTAG
- 5: Power switch



- ❑ 专为芯来科技处理器内核IP的SoC原型评估而设计
- ❑ 主芯片FPGA型号为Xilinx XC7A100T
- ❑ 配备多达126个引出的FPGA GPIO，用于用户自定义使用
- ❑ 备独立的MCU_FLASH芯片，此Flash用于存储RISC-V内核运行的程序文件
- ❑ 配备独立的FPGA_Flash芯片，此Flash用于存储mcs格式的比特流文件
- ❑ 配备用户LEDs、按键及拨码开关
- ❑ 配备蜂鸟调试器接口（2线N-LINK及4线JTAG协议支持）

蜂鸟FPGA评估板

NUCLEI BOARD LABS



- ❑ Nuclei Board Labs是基于Nuclei SDK开发针对RV-STAR开发板开发的实验手册
- ❑ 提供手把手的实验手册，方便RV-STAR开发板使用者入门以及教学
- ❑ 更多详细内容请访问 https://doc.nucleisys.com/nuclei_board_labs/

Q&A

THANKS



www.nucleisys.com

www.rvmcu.com

谢谢关注



官方公众号



硅农亚历山大



个人微信