

# RISC-V MCU development boards

## Table of contents

Introduction.....	3
License.....	3
Manufacturer selection.....	3
BouffaloLab.....	4
Documentation, SDK and tools.....	4
BL602.....	5
Third-party development board: PineCone.....	5
Third-party development board: Ai-WB2-32S.....	5
Third-party development board: DT-BL10 or XT-BL12.....	5
BL702.....	5
Third-party development board: M0 Sense.....	5
Third-party development board: XT-ZB1.....	5
BL616.....	6
Third-party development board: M0S Dock.....	6
Third-party development board: Ai-M62-12F.....	6
BL618.....	6
Third-party development board: M0P Dock.....	6
Third-party development board: Ai-M61-32S.....	6
BL808.....	6
Third-party development board: Ox64.....	6
Third-party development board: M1S Dock.....	7
Espressif.....	8
ESP8684 (ESP32-C2).....	8
Official development board: ESP8684-DevKitM-1.....	8
Third-party development board: ESPC2-12.....	8
Documentation.....	8
ESP32-C3 and ESP8685.....	9
Official development board: ESP32-C3-DevKitM-1.....	9
Third-party development board: ESP32-C3FH4.....	9
Third-party development board: nanoESP32-C3.....	9
Third-party development board: YD-ESP32-C3.....	9
Third-party development board: XIAO ESP32C3.....	9
Third-party development board: ESP32C3-MINI-DK.....	9
Documentation.....	10
ESP32-C6.....	10
Official development board: ESP32-C6-DevKitC-1.....	10
Third-party development board: WeAct ESP32-C6-A.....	10
Documentation.....	10
ESP32-H2.....	10
Official development board: ESP32-H2-DevKitM-1.....	10
Documentation.....	10
GigaDevice.....	11
Official development board: GD32VF103C-START.....	11
Third-party development board: Longan Nano.....	11
Documentation, SDK and tools.....	11
WCH.....	12
Programmer / debugger.....	13
MounRiver IDE.....	13
CH32V003.....	13
Official development board.....	13

Third-party development board: nanoCH32V003.....	13
Third-party development board: QSZNTEC CH32V003.....	14
Third-party development board: Twen CH32V003.....	14
Documentation and SDK.....	14
CH32V103.....	14
Official development board.....	14
Documentation and SDK.....	14
CH32V203.....	14
Official development board.....	14
Third-party development board: BluePill+ CH32V203.....	14
Third-party development board: nanoCH32V203.....	15
Documentation and SDK.....	15
CH32V208.....	15
Official development board.....	15
Documentation and SDK.....	15
CH32V303.....	15
Official development board.....	15
Documentation and SDK.....	15
CH32V305.....	16
Third-party development board: nanoCH32V305.....	16
Documentation and SDK.....	16
CH32V307.....	16
Official development board.....	16
Third-party development board: YD-CH32V307VC.....	16
Third-party development board: CH32V307RC-MINI.....	16
Documentation and SDK.....	16
CH565.....	16
Official development board.....	16
Documentation and SDK.....	16
CH569.....	17
Official development board.....	17
Documentation and SDK.....	17
CH582.....	17
Official development board.....	17
Third-party development board: YD-CH58x.....	17
Third-party development board: 303CH582M01.....	17
Documentation and SDK.....	18
Alternatives to WCH tools.....	18

# Introduction

"How do I get my feet wet with RISC-V?" is a very common question, which is often answered "Buy a RISC-V development board, and practice bare metal development."

This document provides guidance on part selection, as well as pointers to useful resources. It is not a comprehensive guide of RISC-V MCU, but rather a short list of easily approachable parts. To be listed in this guide, parts **must** meet several criteria:

- have decent English documentation,
- have open-source or freely downloadable supporting software,
- be easily available from anywhere in the world, e.g. through AliExpress or LCSC,
- have cheap development boards as easily available as the chips,
- not require a Chinese mobile phone number to download software and documentation.

## License

This document is (c) 2023 Vincent DEFERT and is licensed under the Creative Commons Attribution 4.0 International License.

Information about the license can be found at: <http://creativecommons.org/licenses/by/4.0/>

## Manufacturer selection

MCU manufacturers have different product strategies, leading to different product ranges. An easy and efficient method is to select a manufacturer whose strategy matches your needs, and then see which of their parts best suits your project.

<i>Manufacturer</i>	<i>Wide supply voltage range</i>	<i>Wide package choice</i>	<i>Lots of I/O pins</i>	<i>WiFi</i>	<i>BLE</i>	<i>802.15.4</i>
BouffaloLab				x	x	x
Espressif				x	x	x
GigaDevice			x			
WCH	x	x	x		x	

*Note: manufacturers are sorted in alphabetical order.*

# BouffaloLab

BouffaloLab only manufactures RISC-V chips, so they're undoubtedly committed to this platform. They have a wide range of IoT-oriented MCU, some with SiFive IP, others with T-Head IP. The documentation is good, but their SDK and code examples are a bit chaotic.

However, Pine64 has released a well-documented development board (PineCone) and a module (PineNut), both based on the BL602 MCU, and Lee Lup Yuen has produced excellent training material on this board, which comes in complement to the community built by Pine64 around their products.

It is highly recommended to become quite familiar with the PineCone/BL602's ecosystem before attempting to use other chips from BouffaloLab (e.g. BL702, BL616), with which you will be left a bit on your own.

The following table presents the main differences between BouffaloLab's RISC-V SoC (not the modules):

<i>Part</i>	<i>MHz</i>	<i>RAM</i>	<i>CPU</i>	<i>WiFi</i>	<i>BT</i>	<i>BLE</i>	<i>802.15.4</i>	<i>USB</i>	<i>Ethernet</i>	<i>Camera</i>	<i>AI NPU</i>
BL602	192	276K	RV32IMAFC	b/g/n	no	yes	no	no	no	no	no
BL702	144	132K	RV32IMAFC	no	no	yes	yes	yes	no	no	no
BL706	144	132K	RV32IMAFC	no	no	yes	yes	yes	yes	yes	no
BL616	320	480K	RV32IMAFCP	b/g/n/ax	yes	yes	yes	yes	yes	yes	no
BL808	480	64M	RV32IMAFCP + RV64IMAFCV +RV32EMC	b/g/n/ax	yes	yes	yes	yes	yes	yes	yes

I chose to include the BL808 because it is exceedingly powerful for an MCU, but very limited for a Linux-capable SoC, so it makes sense to include it in both documents.

Finally, the BL604, is a higher pin count version of the BL602, and the BL618, a higher pin count version of the BL616, not included above because otherwise identical.

## Documentation, SDK and tools

Data sheets & reference manuals: [https://github.com/bouffalolab/bl\\_docs](https://github.com/bouffalolab/bl_docs)

SDK:

[https://github.com/bouffalolab/bouffalo\\_sdk](https://github.com/bouffalolab/bouffalo_sdk) (newer SDK)

[https://github.com/bouffalolab/bl\\_iot\\_sdk](https://github.com/bouffalolab/bl_iot_sdk) (older SDK)

Note: the older SDK may still be needed until the newer has full peripheral coverage (e.g. to use WiFi on the BL808).

<https://dev.bouffalolab.com/download>

Flashing tool (BIDevCube): [https://github.com/bouffalolab/flash\\_tools](https://github.com/bouffalolab/flash_tools)

GitHub: <https://github.com/bouffalolab>

Developer forum: <https://bbs.bouffalolab.com/t/english-forum>

Interesting information not found in BouffaloLab documents: <https://github.com/pine64/>

## **BL602**

### **Third-party development board: PineCone**

Purchase link: <https://pine64.com/product/pinecone-bl602-evaluation-board/>

Documentation: <https://wiki.pine64.org/wiki/PineCone>

Review: <https://lupyuen.github.io/articles/pinecone>

Tutorials: <https://lupyuen.github.io/articles/book>

### **Third-party development board: Ai-WB2-32S**

Purchase link: <https://www.aliexpress.com/item/1005004904931700.html>

Documentation: <https://docs.ai-thinker.com/en/wb2>

GitHub: <https://github.com/Ai-Thinker-Open/Ai-Thinker-WB2>

### **Third-party development board: DT-BL10 or XT-BL12**

Purchase links:

<https://www.aliexpress.com/item/1005005083839351.html>

<https://www.aliexpress.com/item/1005003695650307.html>

Documentation: <https://xzrnlk27j.k.topthink.com/@xgr3x6lrjy/BL602.html>

## **BL702**

### **Third-party development board: M0 Sense**

Purchase links:

<https://www.aliexpress.com/item/1005005373072135.html>

<https://www.aliexpress.com/item/1005005012406688.html>

Documentation: <https://dl.sipeed.com/shareURL/Maix-Zero/M0sense>

GitHub: [https://github.com/sipeed/M0sense\\_BL702\\_example](https://github.com/sipeed/M0sense_BL702_example)

### **Third-party development board: XT-ZB1**

Purchase links:

<https://www.aliexpress.com/item/1005004134568356.html>

<https://www.aliexpress.com/item/1005003747200098.html>

Documentation: <https://xzrnlk27j.k.topthink.com/@xgr3x6lrjy/BL702.html>

## **BL616**

### **Third-party development board: M0S Dock**

Purchase links:

<https://www.aliexpress.com/item/1005005373075939.html>

<https://www.aliexpress.com/item/1005005142466936.html>

Documentation: <https://dl.sipeed.com/shareURL/Maix-Zero/M0S>

GitHub: [https://github.com/sipeed/M0S\\_BL616\\_example](https://github.com/sipeed/M0S_BL616_example)

### **Third-party development board: Ai-M62-12F**

Purchase links:

<https://www.aliexpress.com/item/1005005407942430.html>

<https://www.aliexpress.com/item/1005005438854506.html>

Documentation: [https://docs.ai-thinker.com/ai\\_m62](https://docs.ai-thinker.com/ai_m62)

## **BL618**

### **Third-party development board: M0P Dock**

Purchase link: <https://www.aliexpress.com/item/1005005461103465.html>

Documentation: <https://dl.sipeed.com/shareURL/Maix-Zero/M0P>

GitHub: [https://github.com/sipeed/M0P\\_BL618\\_examples](https://github.com/sipeed/M0P_BL618_examples)

### **Third-party development board: Ai-M61-32S**

Purchase links:

<https://www.aliexpress.com/item/1005005407539968.html>

<https://www.aliexpress.com/item/1005005407935386.html>

Documentation: [https://docs.ai-thinker.com/ai\\_m61](https://docs.ai-thinker.com/ai_m61)

## **BL808**

### **Third-party development board: Ox64**

Purchase link: <https://pine64.com/product-category/ox64/>

Documentation: <https://wiki.pine64.org/wiki/Ox64>

## Third-party development board: M1S Dock

Purchase links:

<https://www.aliexpress.com/item/1005004996668405.html>

<https://www.aliexpress.com/item/1005004970779483.html>

Documentation: <https://dl.sipeed.com/shareURL/MAIX/M1s>

GitHub:

[https://github.com/sipeed/M1s\\_BL808\\_example](https://github.com/sipeed/M1s_BL808_example)

[https://github.com/sipeed/M1s\\_BL808\\_SDK](https://github.com/sipeed/M1s_BL808_SDK)

[https://github.com/sipeed/M1s\\_BL808\\_Linux\\_SDK](https://github.com/sipeed/M1s_BL808_Linux_SDK)

# Espressif

Espressif is famous for their Xtensa-based ESP32 and ESP32-S3 modules, but they also announced their new developments would be made on RISC-V, so interesting things are to be expected from them. Like BouffaloLab, they focus on IoT-oriented MCU.

Their current RISC-V product range includes the ESP32-C2/ESP8684, ESP32-C3/ESP8685, ESP32-C6, and ESP32-H2.

The following table presents the main differences between Espressif's RISC-V SoC (not the modules):

Part	Pins	Freq.	RAM	Flash	WiFi	BLE	802.15.4	Notes
ESP8684	24	120MHz	272KB	1/2/4MB	b/g/n	yes	no	Aka. ESP32-C2
ESP32-C3	32	160MHz	400KB	0/4MB	b/g/n	yes	no	
ESP8685	28	160MHz	400KB	2/4MB	b/g/n	yes	no	
ESP32-C6	40	160MHz	512KB	0/4MB	a/x	yes	yes	Has 2 CPU (one low power)
ESP32-H2	32	96MHz	320KB	2/4MB	a/x	yes	yes	Low-power applications

GitHub: <https://github.com/espressif>

The SDK for all Espressif products is called **ESP-IDF**. It is available from GitHub, or can be downloaded from there: [https://www.espressif.com/en/support/download/all?keys=&field\\_type\\_tid%5B%5D=785](https://www.espressif.com/en/support/download/all?keys=&field_type_tid%5B%5D=785)

## ESP8684 (ESP32-C2)

### Official development board: ESP8684-DevKitM-1

Purchase link: <https://www.aliexpress.com/item/1005004436990376.html>

Documentation: <https://docs.espressif.com/projects/espressif-esp-dev-kits/en/latest/esp8684/esp8684-devkitm-1/index.html>

### Third-party development board: ESPC2-12

Purchase links:

<https://www.aliexpress.com/item/1005004861021167.html>

<https://www.aliexpress.com/item/1005004708803007.html>

Documentation: <http://bbs.doit.am/forum.php?mod=viewthread&tid=489&extra=page%3D1>

## Documentation

[https://www.espressif.com/en/support/documents/technical-documents?keys=&field\\_type\\_tid%5B%5D=956](https://www.espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=956)

Relevant documents are *ESP8684 Datasheet*, *ESP8684 Technical Reference Manual*, and *ESP32-C2 Series SoC Errata*.



## ESP32-C3 and ESP8685

### Official development board: ESP32-C3-DevKitM-1

Purchase link: <https://www.aliexpress.com/item/1005003989099547.html>

Documentation: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32c3/hw-reference/esp32c3/user-guide-devkitm-1.html>

### Third-party development board: ESP32-C3FH4

Purchase link: <https://www.aliexpress.com/item/1005004960064227.html>

Documentation: <https://github.com/WeActStudio/WeActStudio.ESP32C3CoreBoard>

### Third-party development board: nanoESP32-C3

Purchase link: <https://www.aliexpress.com/item/1005003081928629.html>

Documentation: <https://github.com/wuxx/nanoESP32-C3>

### Third-party development board: YD-ESP32-C3

Purchase links:

<https://www.aliexpress.com/item/1005004639250865.html>

<https://www.aliexpress.com/item/1005003613170790.html>

Documentation: <http://www.vcc-gnd.com/>

Breaks the ESP32-C3's USB interface out and provides a separate USB-to-serial interface.

### Third-party development board: XIAO ESP32C3

Purchase link: <https://www.aliexpress.com/item/33011482127.html>

Documentation: [https://wiki.seeedstudio.com/XIAO\\_ESP32C3\\_Getting\\_Started/](https://wiki.seeedstudio.com/XIAO_ESP32C3_Getting_Started/)

### Third-party development board: ESP32C3-MINI-DK

Purchase links:

<https://www.aliexpress.com/item/1005004994621831.html>

<https://www.aliexpress.com/i/1005004945580114.html>

<https://www.aliexpress.com/item/1005004945500567.html>

Has similar features as the [ESP32-C3-DevKitC-02](#) except uses ESP32-C3-MINI-1 instead of ESP32-C3-WROOM-02 and CH340 instead of CP2102, and costs half the price.

## Documentation

[https://www.espressif.com/en/support/documents/technical-documents?keys=&field\\_type\\_tid%5B%5D=785](https://www.espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=785)

Relevant documents are *ESP32-C3 Datasheet*, *ESP32-C3 Technical Reference Manual*, and *ESP32-C3 Series SoC Errata*.

## ESP32-C6

### Official development board: ESP32-C6-DevKitC-1

Purchase link: <https://www.aliexpress.com/item/1005005087160183.html>

Documentation: <https://docs.espressif.com/projects/espressif-esp-dev-kits/en/latest/esp32c6/esp32-c6-devkitc-1/index.html>

### Third-party development board: WeAct ESP32-C6-A

Purchase link: <https://www.aliexpress.com/item/1005005569520224.html>

No GitHub yet, but very similar in its design to WeAct's ESP32-S3-A board.

## Documentation

Data sheets & reference manuals:

[https://www.espressif.com/en/support/documents/technical-documents?keys=&field\\_type\\_tid%5B%5D=1177](https://www.espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=1177)

Relevant documents are *ESP32-C6 Datasheet*, *ESP32-C6 Technical Reference Manual*.

## ESP32-H2

### Official development board: ESP32-H2-DevKitM-1

Purchase link: <https://www.aliexpress.com/item/1005005252175587.html>

Documentation: <https://docs.espressif.com/projects/espressif-esp-dev-kits/en/latest/esp32h2/esp32-h2-devkitm-1/index.html>

## Documentation

[https://www.espressif.com/en/support/documents/technical-documents?keys=&field\\_type\\_tid%5B%5D=1211](https://www.espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=1211)

Relevant documents are *ESP32-H2 Datasheet*, and *ESP32-H3 Technical Reference Manual*.

# GigaDevice

While GigaDevice have largely developed their ARM MCU portfolio in the past years, the GD32VF103 is still their only one RISC-V MCU. It's a very interesting part, but unfortunately out-of-stock to date (Feb. 2023), and GigaDevice not being committed to RISC-V doesn't bode well of its future. Anyway, there are still development boards available for the GD32VF103 and plenty of articles and tutorials can be found on the web, so you might want to give it a try.

## Official development board: GD32VF103C-START

Purchase link: <https://www.lcsc.com/product-detail/C432220.html>

Note: this evaluation board includes GigaDevice's GDLink programmer.

## Third-party development board: Longan Nano

Purchase links:

<https://www.aliexpress.com/item/1005002542610332.html>

<https://www.aliexpress.com/item/1005003467064600.html>

Documentation: <https://github.com/sipeed/Longan-DOC>

Downloads: <https://dl.sipeed.com/shareURL/LONGAN/Nano>

Note: an additional JTAG adapter is needed to program the chip. If you don't already have one (e.g. JLink), you can buy Sipeed's USB-JTAG/TTL.

Purchase link: <https://www.aliexpress.com/item/1005002714665888.html>

## Documentation, SDK and tools

Download URL: <https://www.gd32mcu.com/en/download?kw=GD32VF1>

Data sheets & reference manuals: relevant documents are *GD32VF103xx Datasheet* and *GD32VF103 User Manual*.

SDK: relevant archive is *GD32VF103 Firmware Library*.

# WCH

WCH provides a much wider range of RISC-V MCU than any other manufacturer. For this reason, each chip has its own section below.

WCH's offer is divided in 2 families, the general purpose CH32Vxxx series, which are the RISC-V equivalents of their ARM CH32Fxxx series, and the more specialised CHxxx series, which use a slightly different peripheral set.

An interesting thing to note is the CH32V003 doesn't have an ARM equivalent, which could be a hint that, like Espressif, WCH will continue to concentrate their efforts on their RISC-V products.

The following table presents the main differences between WCH's RISC-V MCU:

Part	Features	Core	Max. freq.	Voltage	Notes
<a href="#">CH32V003</a>	RV32EC	QingKe V2A	48 MHz	2.7-5.5V	Low pin count (8, 16, 20)
<a href="#">CH32V103</a>	RV32IMAC	QingKe V3A	80 MHz	2.7-5.5V	
<a href="#">CH32V203</a>	RV32IMAC	QingKe V4B	144 MHz	2.4-3.6V	LQFP-64 also has Ethernet
<a href="#">CH32V208</a>	RV32IMAC	QingKe V4C	144 MHz	2.4-3.6V	BLE + Ethernet
CH32V303	RV32IMAFc	QingKe V4F	144 MHz	2.4-3.6V	
CH32V305	RV32IMAFc	QingKe V4F	144 MHz	2.4-3.6V	
<a href="#">CH32V307</a>	RV32IMAFc	QingKe V4F	144 MHz	2.4-3.6V	Ethernet. LQFP-100 also has DVP & FSMC
<a href="#">CH565</a>	RV32IMAC	QingKe V3A	120 MHz	2.3-3.6V	USB 3.0, Gb Ethernet, EMMC, DVP
<a href="#">CH569</a>	RV32IMAC	QingKe V3A	120 MHz	2.3-3.6V	USB 3.0, Gb Ethernet, EMMC, HSPI
CH573/571	RV32IMAC	QingKe V3A	60 MHz	2.3-3.6V	Superseded by the CH582
<a href="#">CH583/582</a>	RV32IMAC	QingKe V4A	80 MHz	2.3-3.6V	BLE, ultra low power
CH592/591	RV32IMAC	QingKe V4C	80 MHz	2.3-3.6V	BLE, ultra low power, LCD controller
<a href="#">CH643</a>	RV32IMAC	QingKe V4C	48 MHz	2.0-5.5V	USB PD + type C fast charging, PIOC, RGB LED PWM
<a href="#">CH32X035</a>	RV32IMAC	QingKe V4C	48 MHz	2.0-5.5V	USB PD + type C fast charging, PIOC, OpAmp / PGA / comp.

*PIOC: programmable I/O protocol controller / PGA: programmable gain amplifier*

Note: Development boards for the CH592, CH643, and CH32X035 are not available yet as of 2023-06-11. Development boards for the CH573 have deliberately been omitted.

For further details, here is a selection table covering the whole WCH offer:

<https://special.wch.cn/en/mcu/>

*Note: all WCH MCU SDK include the schematics of the official evaluation boards as reference design.*

GitHub: <https://github.com/openwch>

## Programmer / debugger

A proprietary programmer / debugger is needed to flash WCH chips, the WCH-LinkE.

Purchase links:

<https://www.aliexpress.com/item/1005004881582037.html>

<https://www.aliexpress.com/item/1005004964197577.html>

<https://www.aliexpress.com/item/1005005244468643.html>

One is already included with the CH32V003 evaluation kit, and the CH32V003 + CH32V203 evaluation kit combo from WCH.

*Note: you may also come across the WCH-Link (without final E) on AliExpress. It is the predecessor of the WCH-LinkE and does not support the CH32V003.*

The flashing software, WCHISPTool, is unfortunately Windows-only and doesn't work under Wine. If you run Linux, you can flash your MCU with OpenOCD, but you must first unlock the flash write protection using the MounRiver IDE (see below).

Download link: [http://wch-ic.com/downloads/WCHISPTool\\_Setup\\_exe.html](http://wch-ic.com/downloads/WCHISPTool_Setup_exe.html)

There's also a portable command-line flashing tool, but it uses configuration files generated by WCHISPTool... :/

## MounRiver IDE

MounRiver Community Edition is an Eclipse-based IDE supporting all RISC-V and ARM WCH MCU, available for Windows, Linux and Mac. It includes a portable flashing tool, which is convenient if you don't use Windows (see above), particularly to unlock the flash write protection to be able to program your chip.

WCH provides modified versions of OpenOCD and GCC with MounRiver, but these toolchains can also be downloaded separately, e.g. for installation on a continuous integration server.

Download link: <http://www.mounriver.com/>

## CH32V003

### Official development board

Purchase link: <https://www.aliexpress.com/item/1005004895791296.html>

### Third-party development board: nanoCH32V003

Purchase links:

<https://www.aliexpress.com/item/1005005221751705.html>

<https://www.aliexpress.com/item/1005005222228477.html>

Documentation: <https://github.com/wuxx/nanoCH32V003>

## **Third-party development board: QSZNTEC CH32V003**

Purchase links:

<https://www.aliexpress.com/item/1005004964355080.html>

<https://www.aliexpress.com/item/1005005137124754.html>

## **Third-party development board: Twen CH32V003**

Purchase link: <https://www.aliexpress.com/item/1005005343785386.html>

Note: doesn't have a crystal, so all GPIO pins are available.

## **Documentation and SDK**

Data sheet: [http://wch-ic.com/downloads/CH32V003DS0\\_PDF.html](http://wch-ic.com/downloads/CH32V003DS0_PDF.html)

Reference manual: [http://wch-ic.com/downloads/CH32V003RM\\_PDF.html](http://wch-ic.com/downloads/CH32V003RM_PDF.html)

Processor manual: [http://wch-ic.com/downloads/QingKeV2\\_Processor\\_Manual\\_PDF.html](http://wch-ic.com/downloads/QingKeV2_Processor_Manual_PDF.html)

SDK: [https://www.wch.cn/downloads/CH32V003EVT\\_ZIP.html](https://www.wch.cn/downloads/CH32V003EVT_ZIP.html)

## **CH32V103**

### **Official development board**

Purchase links:

<https://www.lcsc.com/product-detail/C2943983.html>

<https://www.lcsc.com/product-detail/C2943982.html>

<https://www.aliexpress.com/item/1005004607642695.html>

## **Documentation and SDK**

Data sheet: [http://wch-ic.com/downloads/CH32V103DS0\\_PDF.html](http://wch-ic.com/downloads/CH32V103DS0_PDF.html)

Reference manual: [http://wch-ic.com/downloads/CH32xRM\\_PDF.html](http://wch-ic.com/downloads/CH32xRM_PDF.html)

Processor manual: [http://wch-ic.com/downloads/QingKeV3\\_Processor\\_Manual\\_PDF.html](http://wch-ic.com/downloads/QingKeV3_Processor_Manual_PDF.html)

SDK: [https://www.wch.cn/downloads/CH32V103EVT\\_ZIP.html](https://www.wch.cn/downloads/CH32V103EVT_ZIP.html)

## **CH32V203**

### **Official development board**

Purchase link: <https://www.aliexpress.com/item/1005004493040662.html>

### **Third-party development board: BluePill+ CH32V203**

Purchase link: <https://www.aliexpress.com/item/1005001474741936.html>

Documentation: <https://github.com/WeActStudio/WeActStudio.BluePill-Plus-CH32>

*Note: this board initially shipped with a CH32V103C8T6, but now uses a CH32V203C8T6.*

### **Third-party development board: nanoCH32V203**

Purchase link: <https://www.aliexpress.com/item/1005004908206775.html>

Documentation: <https://github.com/wuxx/nanoCH32V203>

### **Documentation and SDK**

Data sheet: [http://wch-ic.com/downloads/CH32V203DS0\\_PDF.html](http://wch-ic.com/downloads/CH32V203DS0_PDF.html)

Reference manual: [http://wch-ic.com/downloads/CH32FV2x\\_V3xRM\\_PDF.html](http://wch-ic.com/downloads/CH32FV2x_V3xRM_PDF.html)

Processor manual: [http://wch-ic.com/downloads/QingKeV4\\_Processor\\_Manual\\_PDF.html](http://wch-ic.com/downloads/QingKeV4_Processor_Manual_PDF.html)

SDK: [https://www.wch.cn/downloads/CH32V20XEVT\\_ZIP.html](https://www.wch.cn/downloads/CH32V20XEVT_ZIP.html)

## **CH32V208**

### **Official development board**

Purchase link: <https://www.aliexpress.com/item/1005004924242063.html>

### **Documentation and SDK**

Data sheet: [http://wch-ic.com/downloads/CH32V208DS0\\_PDF.html](http://wch-ic.com/downloads/CH32V208DS0_PDF.html)

Reference manual: [http://wch-ic.com/downloads/CH32FV2x\\_V3xRM\\_PDF.html](http://wch-ic.com/downloads/CH32FV2x_V3xRM_PDF.html)

Processor manual: [http://wch-ic.com/downloads/QingKeV4\\_Processor\\_Manual\\_PDF.html](http://wch-ic.com/downloads/QingKeV4_Processor_Manual_PDF.html)

SDK: [https://www.wch.cn/downloads/CH32V20XEVT\\_ZIP.html](https://www.wch.cn/downloads/CH32V20XEVT_ZIP.html)

## **CH32V303**

### **Official development board**

Purchase link: <https://www.aliexpress.com/item/1005005444077007.html>

### **Documentation and SDK**

Data sheet: [http://wch-ic.com/downloads/CH32V307DS0\\_PDF.html](http://wch-ic.com/downloads/CH32V307DS0_PDF.html)

Reference manual: [http://wch-ic.com/downloads/CH32FV2x\\_V3xRM\\_PDF.html](http://wch-ic.com/downloads/CH32FV2x_V3xRM_PDF.html)

Processor manual: [http://wch-ic.com/downloads/QingKeV4\\_Processor\\_Manual\\_PDF.html](http://wch-ic.com/downloads/QingKeV4_Processor_Manual_PDF.html)

SDK: [https://www.wch.cn/downloads/CH32V307EVT\\_ZIP.html](https://www.wch.cn/downloads/CH32V307EVT_ZIP.html)

## CH32V305

### Third-party development board: nanoCH32V305

Purchase link: <https://www.aliexpress.com/item/1005005033298927.html>

Documentation: <https://github.com/wuxx/nanoCH32V305>

### Documentation and SDK

Data sheet: [http://wch-ic.com/downloads/CH32V307DS0\\_PDF.html](http://wch-ic.com/downloads/CH32V307DS0_PDF.html)

Reference manual: [http://wch-ic.com/downloads/CH32FV2x\\_V3xRM\\_PDF.html](http://wch-ic.com/downloads/CH32FV2x_V3xRM_PDF.html)

Processor manual: [http://wch-ic.com/downloads/QingKeV4\\_Processor\\_Manual\\_PDF.html](http://wch-ic.com/downloads/QingKeV4_Processor_Manual_PDF.html)

SDK: [https://www.wch.cn/downloads/CH32V307EVT\\_ZIP.html](https://www.wch.cn/downloads/CH32V307EVT_ZIP.html)

## CH32V307

### Official development board

Purchase link: <https://www.lcsc.com/product-detail/C2943980.html>

### Third-party development board: YD-CH32V307VC

Purchase link: <https://www.aliexpress.com/item/1005004367173443.html>

Documentation: <http://www.vcc-gnd.com/>

### Third-party development board: CH32V307RC-MINI

Purchase link: <https://www.aliexpress.com/item/1005005175678285.html>

### Documentation and SDK

Data sheet: [http://wch-ic.com/downloads/CH32V307DS0\\_PDF.html](http://wch-ic.com/downloads/CH32V307DS0_PDF.html)

Reference manual: [http://wch-ic.com/downloads/CH32FV2x\\_V3xRM\\_PDF.html](http://wch-ic.com/downloads/CH32FV2x_V3xRM_PDF.html)

Processor manual: [http://wch-ic.com/downloads/QingKeV4\\_Processor\\_Manual\\_PDF.html](http://wch-ic.com/downloads/QingKeV4_Processor_Manual_PDF.html)

SDK: [https://www.wch.cn/downloads/CH32V307EVT\\_ZIP.html](https://www.wch.cn/downloads/CH32V307EVT_ZIP.html)

## CH565

### Official development board

Purchase link: <https://www.aliexpress.com/item/1005004346104186.html>

### Documentation and SDK

Same as CH569.



## CH569

### Official development board

Purchase links:

<https://www.lcsc.com/product-detail/C3001176.html>  
<https://www.aliexpress.com/item/1005004328816871.html>

### Documentation and SDK

Data sheet: [http://wch-ic.com/downloads/CH569DS1\\_PDF.html](http://wch-ic.com/downloads/CH569DS1_PDF.html)

Processor manual: [http://wch-ic.com/downloads/QingKeV3\\_Processor\\_Manual\\_PDF.html](http://wch-ic.com/downloads/QingKeV3_Processor_Manual_PDF.html)

SDK: [https://www.wch.cn/downloads/CH569EVT\\_ZIP.html](https://www.wch.cn/downloads/CH569EVT_ZIP.html)

## CH582

### Official development board

Purchase links:

<https://www.lcsc.com/product-detail/C2943981.html>  
<https://www.aliexpress.com/item/1005005060737000.html> (CH582M/CH583M)  
<https://www.aliexpress.com/item/1005004346585597.html> (CH582F)

### Third-party development board: YD-CH58x

Purchase links:

<https://www.aliexpress.com/item/1005005305938011.html>  
<https://www.aliexpress.com/item/1005004787513484.html>  
<https://www.aliexpress.com/item/1005004794466027.html>

Documentation: <http://www.vcc-gnd.com/>

Can be programmed and debugged using OpenOCD and a WCH-Link or WCH-LinkE programmer, so is suitable for use with Linux.

### Third-party development board: 303CH582M01

Purchase links:

<https://www.aliexpress.com/item/1005005458836770.html>  
<https://www.aliexpress.com/item/1005005467071580.html>  
<https://www.aliexpress.com/item/1005005456987838.html>

Similar to YD-CH58x. After ordering, ask the seller for schematic diagram. Uses the USB C connector for programming, which at the moment only works under Windows.

## Documentation and SDK

Data sheet: [http://wch-ic.com/downloads/CH583DS1\\_PDF.html](http://wch-ic.com/downloads/CH583DS1_PDF.html)

Processor manual: [http://wch-ic.com/downloads/QingKeV4\\_Processor\\_Manual\\_PDF.html](http://wch-ic.com/downloads/QingKeV4_Processor_Manual_PDF.html)

SDK: [https://www.wch.cn/downloads/CH583EVT\\_ZIP.html](https://www.wch.cn/downloads/CH583EVT_ZIP.html)

## Alternatives to WCH tools

If you would like to use mainstream GCC instead of WCH's, or to try an open-source flashing tool, the following Reddit thread will be of interest to you:

<https://www.reddit.com/r/RISCV/comments/115u6i9/comment/j94xvpq/>