

# Step-by-Step Guide to Program the CH32V003 Using MounRiver Studio

## 1. Download and Install MounRiver Studio

### For Windows:

- **Step 1:** Visit the [MounRiver Studio website](#) and navigate to the download section. Choose the latest version suitable for your operating system (Windows).
- **Step 2:** Run the installer and follow the prompts to complete the installation.
- **Step 3:** Open MounRiver Studio after installation to ensure it runs correctly.

### For Linux:

- **Step 1:** Visit the [MounRiver Studio website](#) and navigate to the download section. Choose the latest version suitable for your operating system (Linux).
- **Step 2:** Extract the contents of the '.tar.xz' file into a known directory. And open the 'beforeinstall' folder and execute the 'start.sh' as administrator.
- **Step 3:** Now open the 'MRS\_Community' folder and run/double-click the 'MounRiver Studio\_Community' file to launch the application.
- **Step 4:** Make sure this folder is kept in a reachable location and not in the root directory (because **step 3** is how you will open the application every time you want to launch MounRiver Studio).

## 2. Set Up Your Workspace

- **Create a New Workspace:**
  - When you first open MounRiver Studio after installation, it will prompt you to create or select a workspace. This directory will store all your projects and settings for easier access.
- **Choose a Workspace Location:**
  - Select a location on your computer where you'd like to save this workspace (e.g., [Documents/MounRiver\\_Workspace](#)).
- **Switch Workspaces (if needed):**
  - If you want to switch between different workspaces, go to **File > Switch Workspace > Other...** and choose a new or existing workspace directory.

## 3. Set Up the WCH-LinkE Debugger

- **Step 1:** Connect the **WCH-LinkE** debugger to your computer.

- **Step 2:** Install any necessary drivers for the WCH-LinkE (these should be available on the WCH website if not pre-installed).
- **Step 3:** Verify that **CH32V003** support is available in MounRiver Studio by selecting CH32V003 as the target microcontroller when creating your project.
- **Step 4:** If required, download any additional toolchains or files from WCH's official resources to support development with the CH32V003.

#### 4. Connect Your CH32V003 Microcontroller with WCH-LinkE

- **Step 1:** Connect your CH32V003 board to your computer using the **WCH-LinkE** programmer, specifically designed for this microcontroller. The CH32V003 uses a **proprietary 1-wire protocol** for programming, requiring the WCH-LinkE or other compatible WCH-Link debuggers for communication.
- **Step 2:** Ensure the programming pins (SWDIO and VCC, GND) are connected properly.

#### 5. Set Up a New Project in MounRiver Studio

- **Step 1:** Open MounRiver Studio, and go to **File > New > MounRiver Project**.
- **Step 2:** Select **CH32V003** from the list of supported devices, and select the project template suitable for your CH32V003 microcontroller.
- **Step 3:** Give the appropriate **Project Name** and click **Finish**.

#### 6. Configure the Project Settings in MounRiver Studio(if necessary)

- **Step 1:** Open the project settings by navigating to **Project > Properties**.
- **Step 2:** Under **C/C++ Build > Settings**, confirm the correct **Toolchain** (should be for RISC-V) and select optimization settings if needed.
- **Step 3:** Set up debugging options in Debug Configurations, such as enabling the WCH-LinkE debugger.

*Note:* For CH32V003, no specific memory map or detailed clock configuration is required, as these are preset for basic projects.

#### 7. Write or Import Your Code

- **Step 1:** Create or import a simple example, like an LED blink. Open the **main.c** file generated in your project folder.
- **Step 2:** Write your code, beginning with the necessary initializations.
- **Step 3:** Save your code. To verify, click **Build** or press **Ctrl+B** to compile the code and check for any errors.

## 8. Configure the Debugger/Programmer

- **Step 1:** Ensure the **WCH-LinkE** debugger is properly connected to your CH32V003 development board as described in **Stage 3**.
- **Step 2:** In **MounRiver Studio**, go to **Run > Debug Configurations**.
- **Step 3:** Under the **Debug Configurations** window, locate and select the **CH32V003** target configuration, or create a new configuration if one doesn't already exist.
- **Step 4:** Set the **Debugger** option to **WCH-LinkE**.
- **Step 5:** Ensure that **SWD** (Serial Wire Debug) is selected as the connection interface, as this is essential for communicating with CH32V003.
- **Step 6:** Confirm that the target microcontroller is set to **CH32V003**.

## 9. Flash the Program onto the CH32V003

- **Step 1:** Go to **Flash > Download** or press **F8** to program the microcontroller.
- **Step 2:** MounRiver Studio initiates communication with WCH-LinkE, which uploads the code. Once the upload is complete, you should see a **"Programming successful"** message in the console.

## 10. Verify and Debug the Program

- **Step 1:** To test your program, connect LED to the any other configured GPIO pin or use the onboard LED on PD4.
- **Step 2:** In MounRiver Studio, select **Run > Debug As > OpenOCD MRS Debugging** (or any other debugger set in **Stage 7**) to execute your program in debug mode.
- **Step 3:** Observe the LED on your board. If the program runs as expected, the LED should blink according to the delay loop.
- **Debugging Options:** Use MounRiver Studio's tools to:
  - Set breakpoints by clicking on the left side of the line number in your code.
  - Monitor registers and variables in the **Debug** panel, using the Debugger Perspective.
  - Step through the code line by line for deeper inspection.

*Note:* This stage changes/switches the programming perspective to the debugging perspective. To get back to the programming perspective again simply click on the **MRS develop** option(next to debin option) in the top right corner.

## 11. Testing Other Features

After setting up the basics, you can expand to additional functionalities:

- **PWM Control:** Set up PWM outputs to control RGB LEDs or servo motors.
- **ADC Configuration:** Experiment with analog readings using the ADC module.
- **Communication Protocols:** Try I2C, SPI, or USART communication to interface with other peripherals.
- **Low Power Modes:** Configure sleep and standby modes to test power-saving functionalities.

For further example codes related to these features, visit the official GitHub page for the CH32V003 by WCH: [WCH GitHub CH32V003 Examples](#).

## 12. Accessing Documentation

Keep the following documentation handy:

- **Datasheets:** Overview of specifications and hardware features.
- **Reference Manual:** Details on registers, peripherals, and control mechanisms.
- **Schematic/Board Layout:** Reference your board's layout for easy prototyping.

These documents can also be found in the Hardware Files, or visit the official GitHub page for the CH32V003 by WCH: [WCH GitHub CH32V003](#).

## Creating C++ Project in MounRiver Studio:

WCH has done all the work for you [here](#) and made a clear document on creating a C++ project, You can import the project file directly from their repository or you can find their '[Use MRS Create C++ project](#)' file to guide you on creating it on your own.