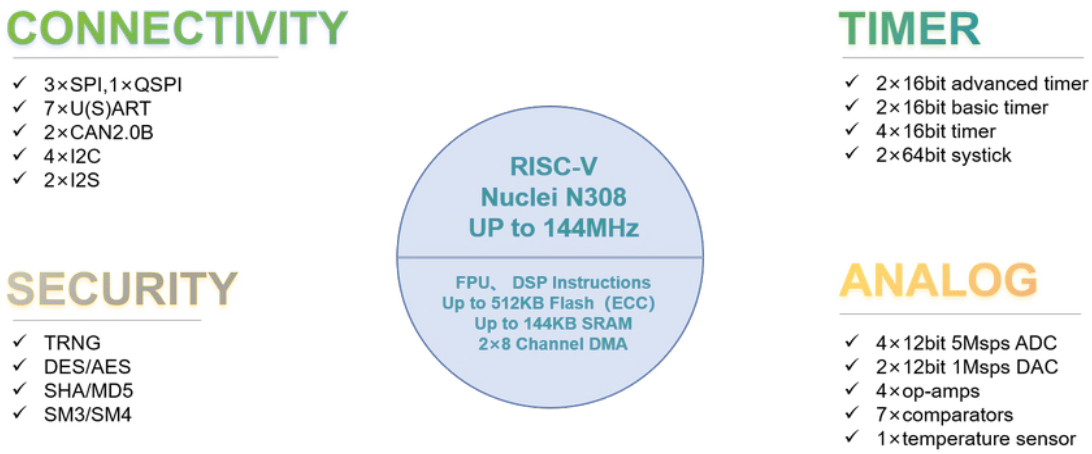
Table of contents

The CM32M433R MCU

Original post: https://www.rvmcu.com/quickstart-show-id-13.html

1. Introduction

The CM32M4xxR series chips are the first low-power and large-capacity microcontrollers based on RISC-V architecture launched by China Mobile Xinsheng Technology. This series of MCUs has the characteristics of high performance, high reliability, high security, and low power consumption, and can be widely used in smart door locks, IoT gateways, interactive panels, measurement and control terminals, student education, consumer electronics and other related fields.



- Core: Nuclei N308 (RV32IMACFP)
- Main frequency: 144MHz
- Memory: built-in 512KB Flash, 144KB SRAM
- Operating voltage: 1.8V ~ 3.6V
- Built-in cryptographic algorithm hardware acceleration engine
- Peripheral Resources: timers (General Purpose Timer, Advanced Timer, Basic Timer), SPI, I2S, QSPI, I2C, USART, UART, CAN, ADC, DAC, TSC, GPIO

2. Purchase links

Retail purchase

Taobao: https://item.taobao.com/item.htm?id=677606472384

Volume Purchase

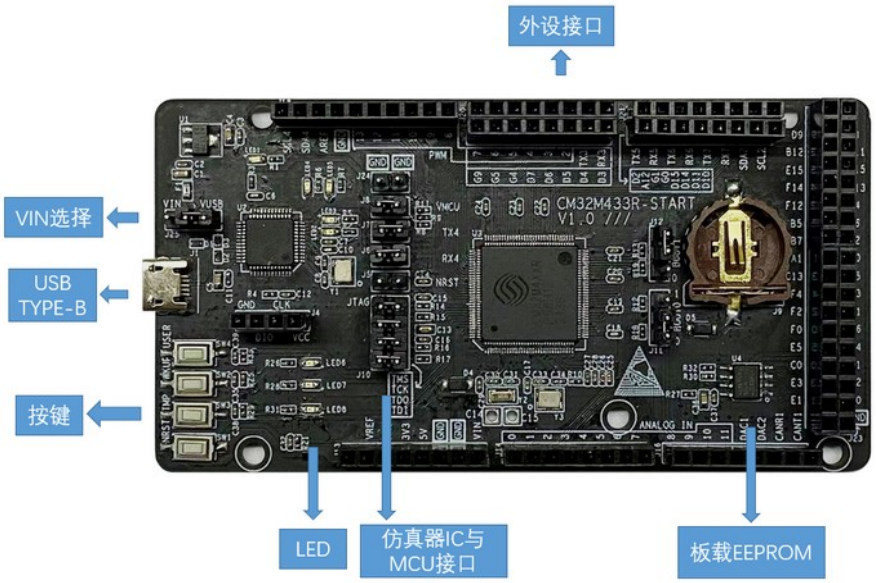
Contact: Manager Han
Tel: +86 18111213896
E-mail: hanyongchao@cmiot.chinamobile.com

The CM32M433R-START development board

Original post: https://www.rvmcu.com/quickstart-show-id-14.html

1. Introduction

CM32M433R-START is a RISC-V development board based on China Mobile's CM32M433R MCU. It features an on-board emulator and Arduino-compatible female headers.

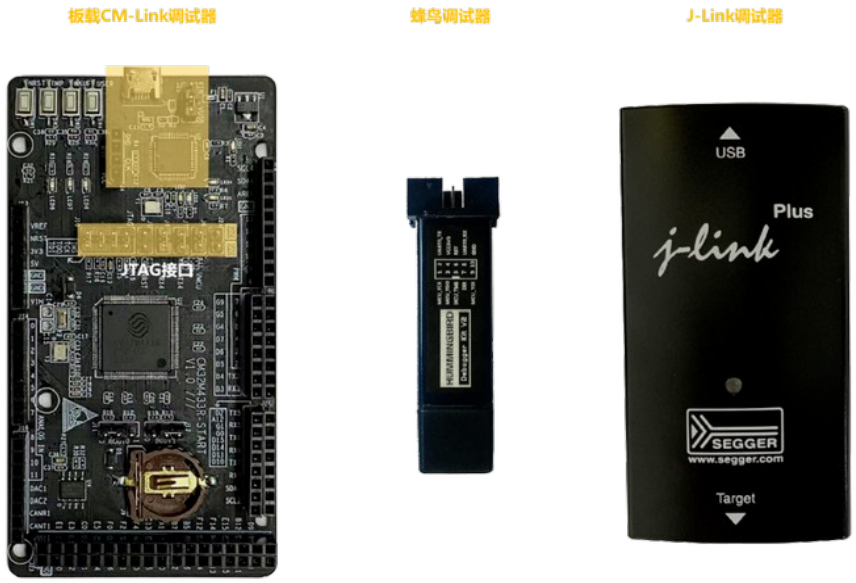


CM32M433R-START development board features:

- Microcontroller: CM32M433R
  - Core: Nuclei N308 (RV32IMACFP)
  - Main frequency: 144MHz
  - Memory: 512KB Flash, 144KB SRAM
  - Working voltage: 1.8~3.6V
  - Peripheral resources: Timers (4 general-purpose timers, 2 advanced timers, 2 basic timers) SPI3, I2S2, QSPI1, I2C4, UART4, CAN2, ADC2, DAC2
- Power supply mode: USB or 5V DC external power supply
- Size: 10.1*5.3cm
- Peripherals and interfaces:
  - USB Micro-B interface: download, debug, power supply
  - Arduino-compatible standard single row 2.54mm female headers
  - JTAG interface
  - Reset button
  - 3 user buttons
  - 3 user LED

The CM32M433R-START development board integrates an on-board CM-Link debugger, and only needs a USB cable for development. However, the on-board debugger can be disconnected from the MCU by removing jumpers to allow the use of a Hummingbird or J-Link debugger instead.

Similarly, the integrated USB-to-serial adapter can also be disconnected from the MCU by removing jumpers to allow the use of an external one, for instance the one included in the Hummingbird debugger.



2. Purchase links

Taobao: https://item.taobao.com/item.htm?id=671656606437

Aliexpress: https://www.aliexpress.com/item/1005004333840765.html

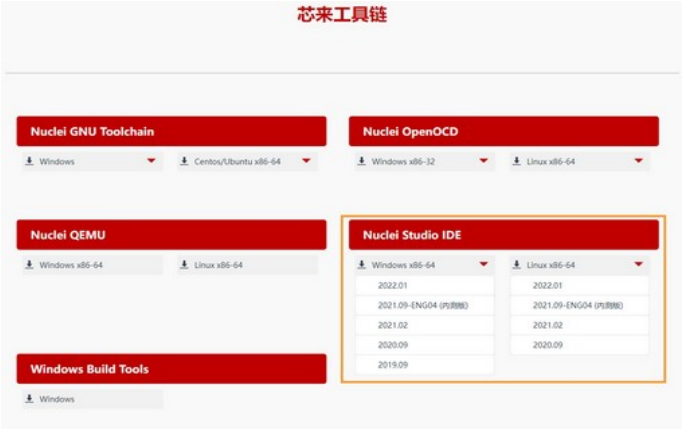
CM32M433R-START quick start guide

Original post: https://www.rvmcu.com/quickstart-show-id-15.html

1. Download and execution of Nuclei Studio

Nuclei Studio can be downloaded from the "Documents and Tools" page of Nuclei's official website (https://www.nucleisys.com/download.php).

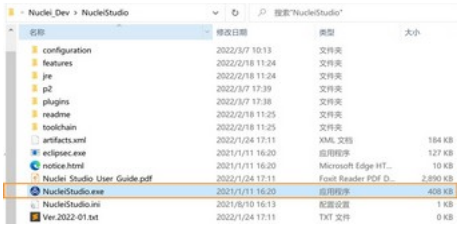
Nuclei continuously updates and maintains the tools it provides, so it is recommended to use the latest version.



Notes to Linux users:

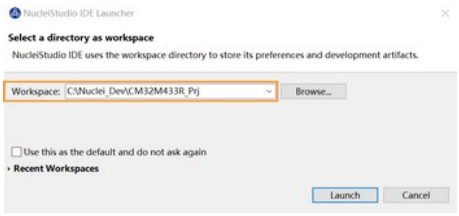
- The content type of Linux archives is not properly set on Nuclei's web server. For proper download, you must right-click the desired file version and select "Save link as..." in the context menu.
- If you have trouble directly uncompressing Linux archives, try to proceed in 2 steps: first "gzip -d the-archive.tgz", then "tar xf the-archive.tar".

Simply unzip the compressed package and run the executable file under the NucleiStudio folder to start the Nuclei Studio IDE.

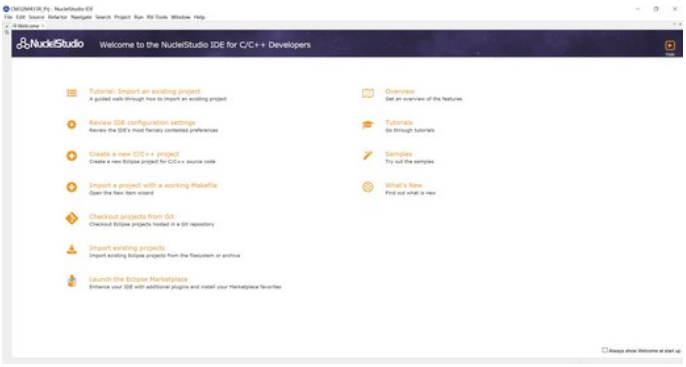


The installation procedure is similar under Linux, the executable file name is "NucleiStudio".

After starting the Nuclei Studio IDE for the first time, a dialogue box will pop up asking to set the path of the Workspace directory, which will be used to store the subsequently created project files.

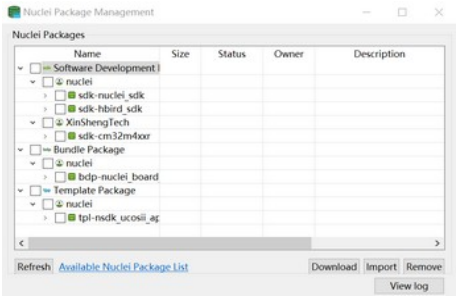


After setting the Workspace directory, click the "Launch" button to start Nuclei Studio. [If it crashes, just restart it.] The Nuclei Studio interface after the first startup is shown below.

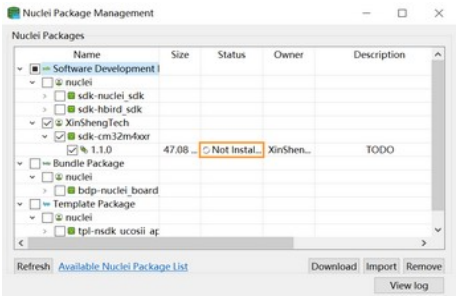


2. Import CM32M4xxR SDK

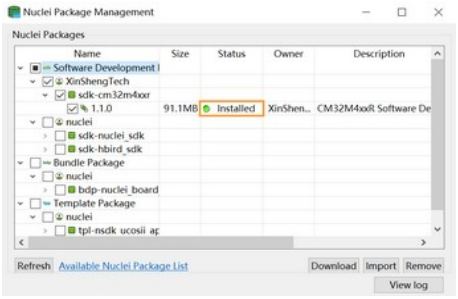
Select "RV-Tools —> Nuclei Package Management" in the Nuclei Studio menu bar, the pop-up dialogue box below will show up.



Expand "sdk-cm32m4xxr" to see that its status is "Not Installed".

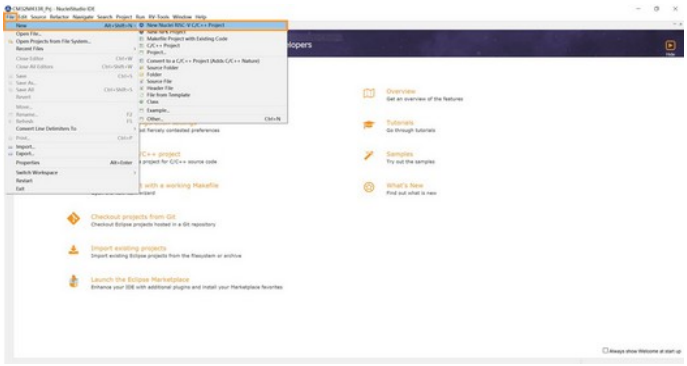


Check "sdk-cm32m4xxr", make sure the combo box says "Save to global path", and click "Download" to import the CM32M4xxR SDK. After completion, the status will be displayed as "Installed".

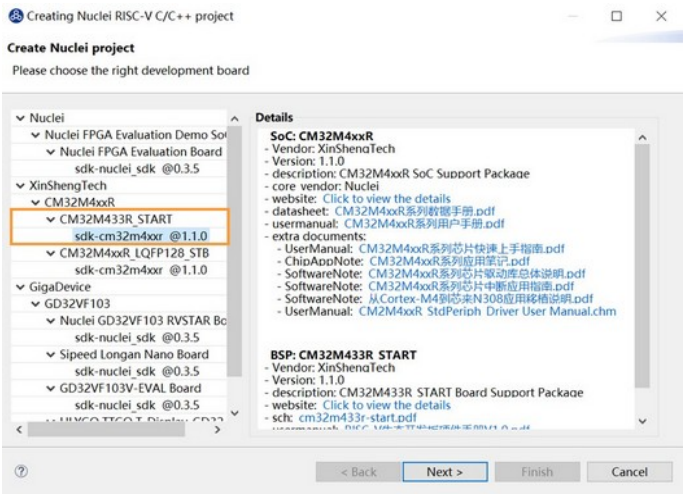


3. Create a new HelloWorld project

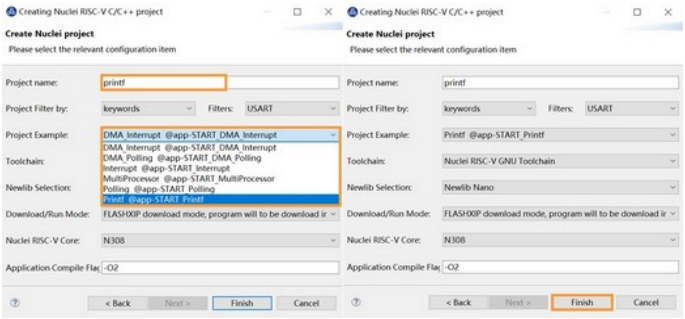
Select "File —> New —> New Nuclei RISC-V C/C++ Project" in the Nuclei Studio menu bar.



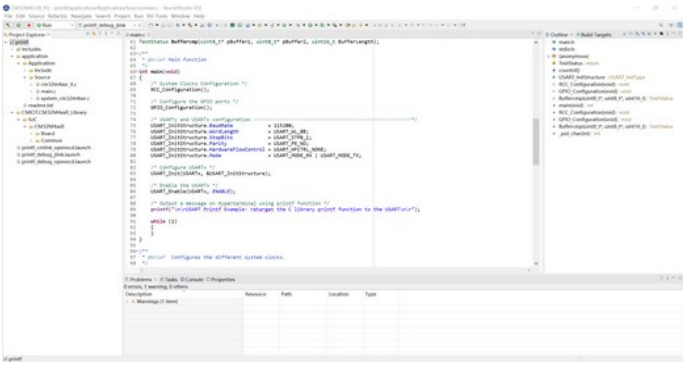
In the pop-up dialogue box, select "sdk-cm32m4xxr" under the "CM32M433R_START" column, and click "Next".



Enter the project name "printf" in the "Project Name" column, select "Printf @app-START Printf" in the "Project Example" drop-down column, and leave other settings unchanged. Click "Finish" to complete the creation of the project.



Once the project is created, the user interface looks as shown below. From the Project Explorer column on the left, you can see that the created project already includes support for the CM32M4xxR SDK.



4. Project introduction

Looking at the main function, we can see:

- Serial port initialization settings
- Print "USART Printf Example: retarget the C library printf function to the USART"

The printf function of the Newlib C runtime library calls the _write function, which outputs characters one by one by using the _put_char function, whose default implementation does nothing, as shown in the excerpt from CMIOT.CM32M4xxR_Library/SoC/CM32M4xxR/Common/Env/Stubs/write.c shown below.

```
__attribute__((weak)) int _put_char(int ch)
{
    return ch;
}


ssize_t _write(int fd, const void* ptr, size_t len)
{
    if (!isatty(fd)) {
        return -1;
    }

    const uint8_t * writebuf = (const uint8_t *)ptr;

    for (size_t i = 0; i < len; i++) {
        if (writebuf[i] == '\n') {
            _put_char('\r');
        }

        _put_char(writebuf[i]);
    }

    return len;
}
```

The _put_char function is overloaded in main.c as follows.

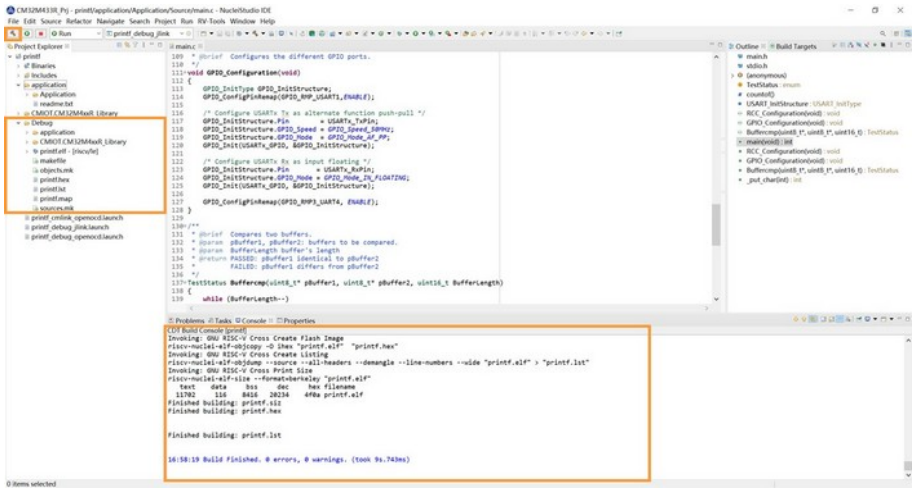
```c
/**
  * @brief  Redefine printf function.
  */
int _put_char(int ch)
{
    USART_SendData(USARTx, (uint8_t)ch);

    /* Loop until the end of transmission */
    while (USART_GetFlagStatus(USARTx, USART_FLAG_TXDE) == RESET)
    {

    }

    return ch;
}
```

As you can see, _put_char outputs characters to the serial port by calling the USART_SendData function of the USART module, and they finally get displayed on the screen of the host PC with the help of the serial port debugging assistant.

5. Project compilation

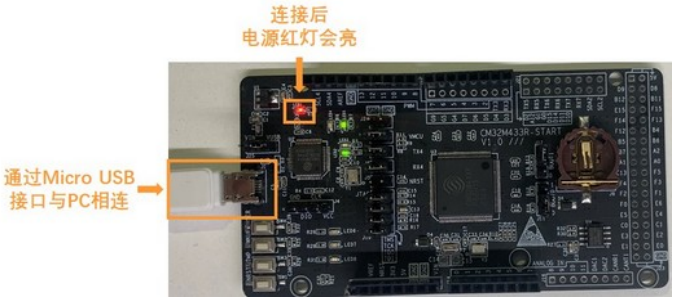
The newly created project has already set the configuration options related to the compilation link, and you can click the "hammer" button on the Nuclei Studio IDE menu to compile directly.

If the compilation is successful, it will be displayed as shown below. You can see the code size of the generated executable file, including the text segment, data segment, and bss segment, as well as the decimal and hexadecimal values of the total size. The compiled files are stored in the Debug directory.



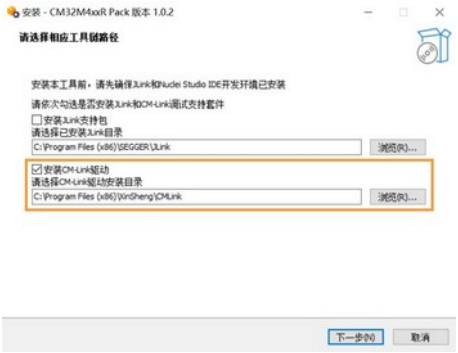
6. Execution

First, you need to connect the CM32M433R-START development board to the PC through a Micro USB cable. Once connected, the red power light on the board will light up.



The first time you use the CM32M433R-START development board, you need to install the driver for it.

For Windows, download, unzip, and execute: https://www.rvmcu.com/app/quickstart/skins/default/software/CM32M4xxR-Support-Pack-v1.0.2-win32-x32.zip



Don't check the installation of the J-Link support package for now.

The on-board debugger also includes a UART-to-USB adapter. Therefore, after connecting the CM32M433R-START development board to the PC, it will be recognized by the host as 2 COM serial port, which can be viewed through the device manager of the PC:

For Linux, execute the following operations:

- Use the lsusb command to view the status of the USB device:

```
lsusb
# After running this command, the following information will be displayed.
...
Bus 001 Device 003: ID 7777:5740 CMIOT CM-LINK
```

- Use the following command to find the tty device corresponding to the debugging interface:

```
ls -al /dev/serial/by-id | grep usb-CMIOT.*
# After running this command, the following information will be displayed.
...
...usb-CMIOT_CM-LINK_3602115033413635331441 5E-if00 -> ../../ttyACM0
...usb-CMIOT_CM-LINK_3602115033413635331441 5E-if02 -> ../../ttyACM1
```

- Add the host user to the dialout group:

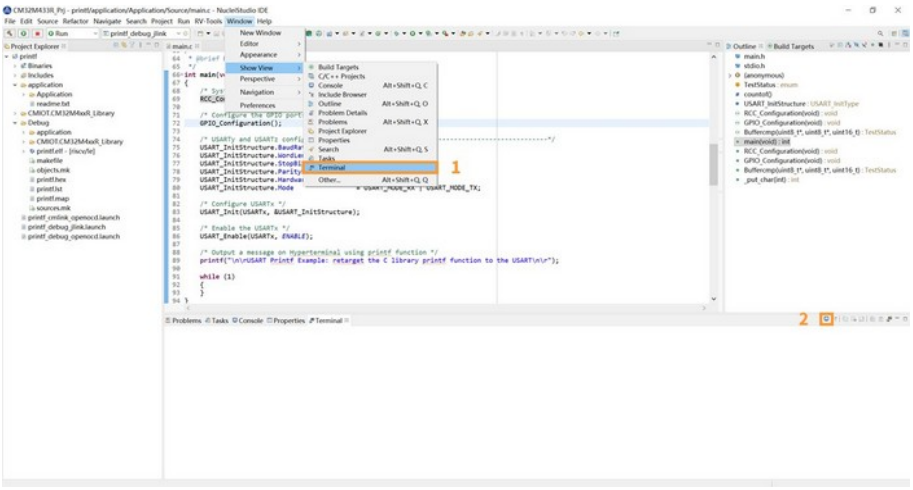
```
# Run this command to display the host user name
whoami
# Assuming the host user name is displayed as your_user_name,
# run the following command to add your_user_name to the dialout group
sudo usermod -a -G dialout your_user_name
```

- Confirm whether the host user belongs to the dialout group

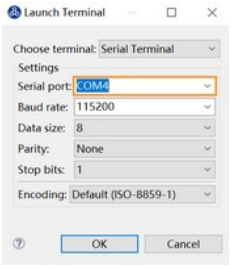
```
groups
# As long as dialout is seen from the displayed groups, it means that the host user
# belongs to this group, indicating that the setting is successful
```

- You need to log out and then back in for this change to take effect

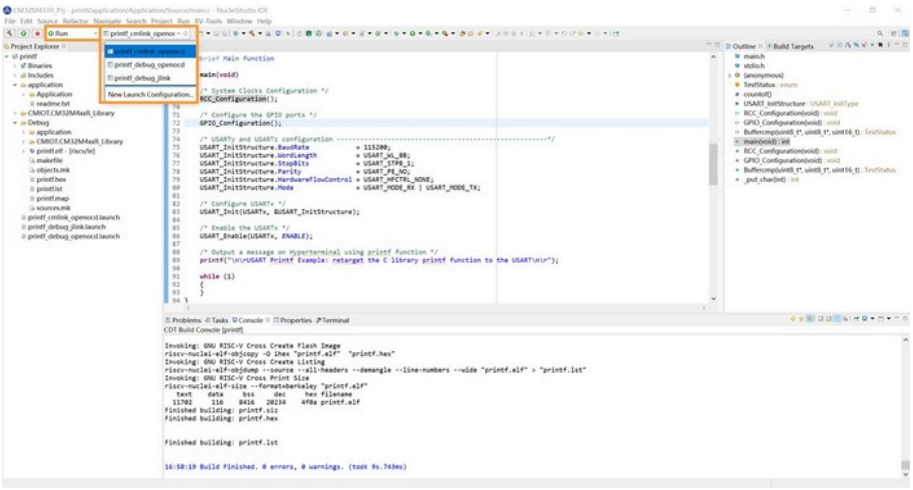
The serial port debugging tool is integrated in the Nuclei Studio IDE. Select "Window->Show View->Terminal" in the menu bar, then click the "Display" icon button in the "Terminal" window to open the serial port tool configuration interface.



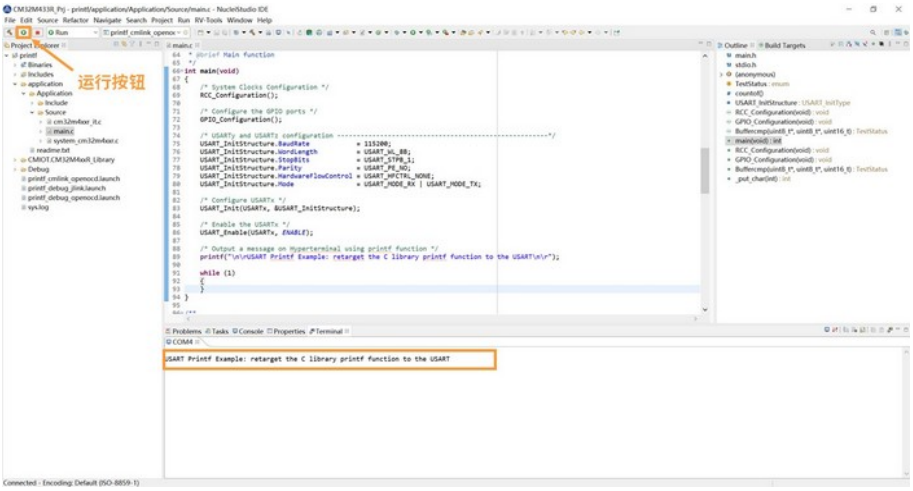
In the pop-up window, set the parameters related to serial communication: 115200 baud, 8 data bits, no parity, 1 stop bit (115200-8-N-1). Select the serial port to match the device determined earlier.



In the "Launch Bar" under the Nuclei Studio IDE menu bar, set the mode to "Run" and configure it to "printf_cmlink_openocd", as shown below.

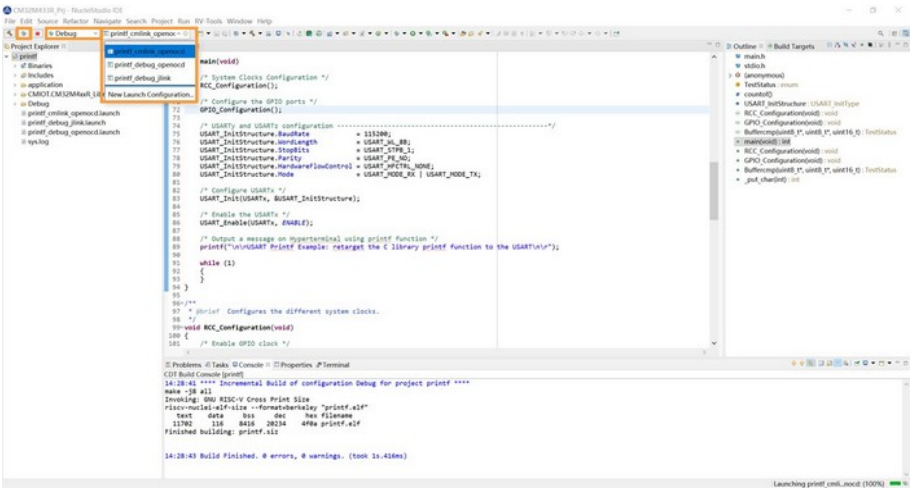


Once configured, click the green "Run" button to start downloading the program to the CM32M433R-START development board and run the sample project. The following message should display in the Terminal window.

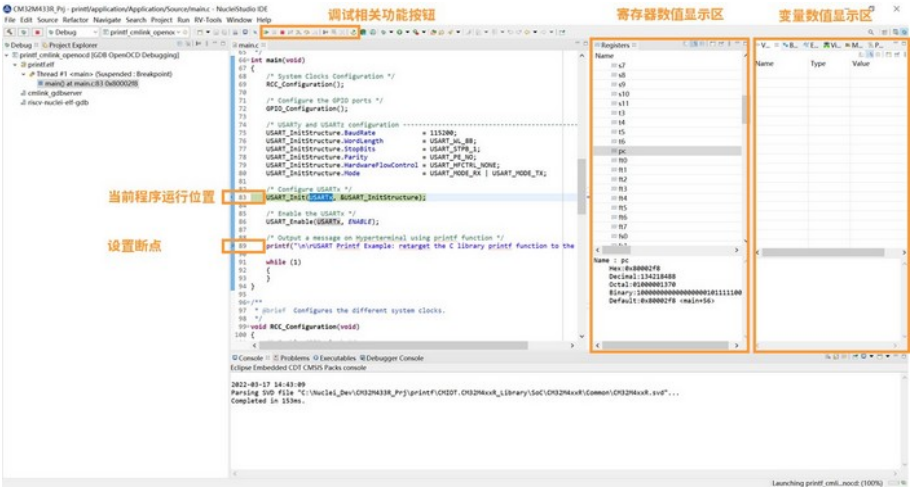


7. Debugging

Similarly to the previous step, set the mode to "Debug" in the "Launch Bar" under the Nuclei Studio IDE menu bar, and configure it to "printf_cmlink_openocd", as shown below.



Then, click the "Beetle" button to start downloading the program to the CM32M433R-START development board, and enter the debugging mode, the interface is shown below.



CM32M433R-START on-board peripherals use

Original post: https://www.rvmcu.com/quickstart-show-id-16.html

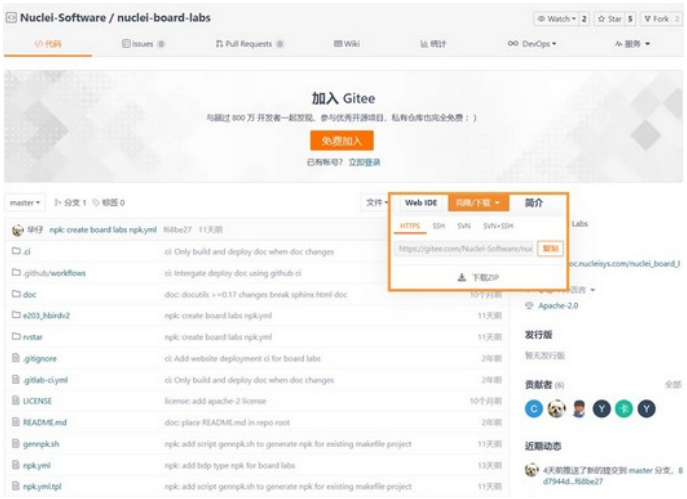
This document guides users on how to import NPK that contains more routines for the CM32M433R-START development board, and create related application projects for onboard peripherals. It is intended for users who already have a certain background. For a detailed introduction to the process of downloading and installing Nuclei Studio IDE, compiling, debugging and running applications, please refer to CM32M433R-START quick start guide.

1. Import Nuclei Board Labs

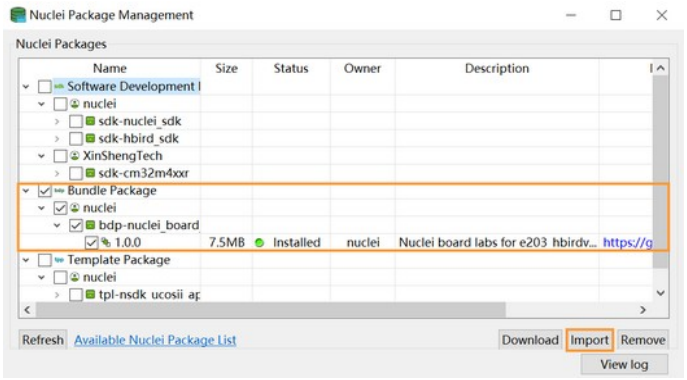
Nuclei Board Labs is a package provided by Nuclei for its hardware platforms (including the CM32M433R-START development board) and used in the courses of the Nuclei University Program. It is maintained on Github (https://github.com/Nuclei-Software/nuclei-board-labs) and NPK can be directly imported and used in the Nuclei Studio IDE.

Nuclei Board Labs only provides application layer code, which needs to be used with the underlying SDK. It is thus necessary to import the CM32M4xxR SDK beforehand.

- Download the zip package directly from the Nuclei Board Labs Github repository as shown below.

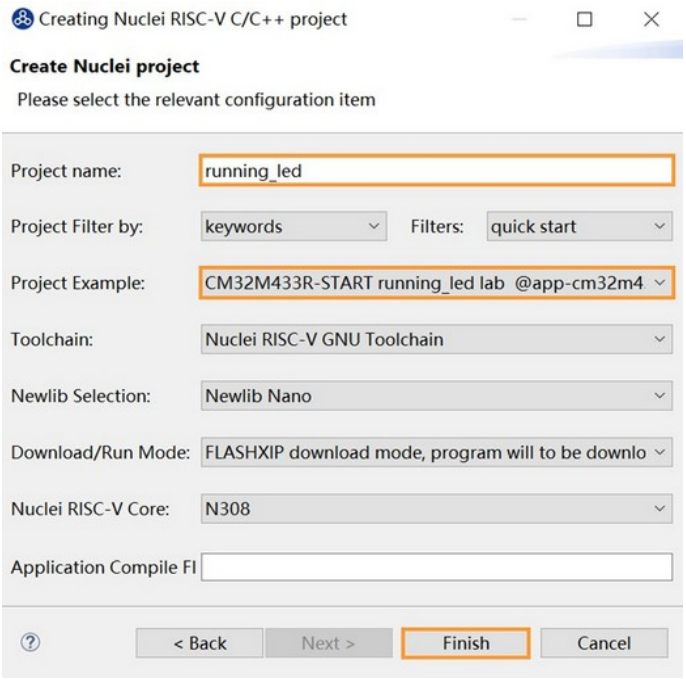


- Start Nuclei Studio IDE, open "Nuclei Package Management", click the "Import" button in the pop-up dialogue box, and add the downloaded Nuclei Board Labs zip package.



2. New project

Create a new project and select "CM32M433R-START running_led lab" as template. Leave other settings unchanged.



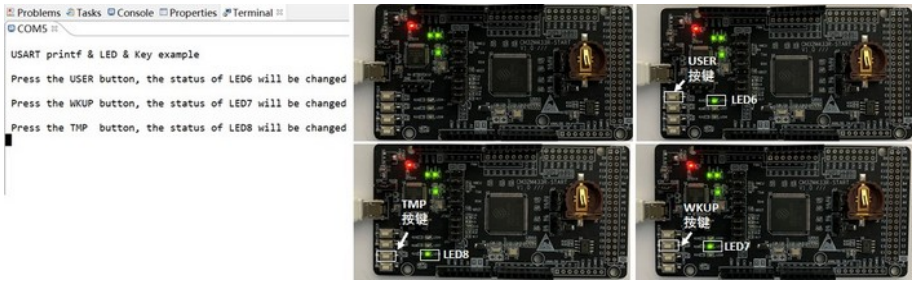
3. Project introduction

The main function implements the following steps:

- Initialize the serial port for printing
- Initialize the GPIO connected to the onboard LED
- Initialize the GPIO connected to the onboard button
- Print the usage instructions
- Loop: detect which button is pressed and toggle the corresponding LED

4. Execution

The picture below shows serial port output on the left, and the outcome of the different key presses on the right.



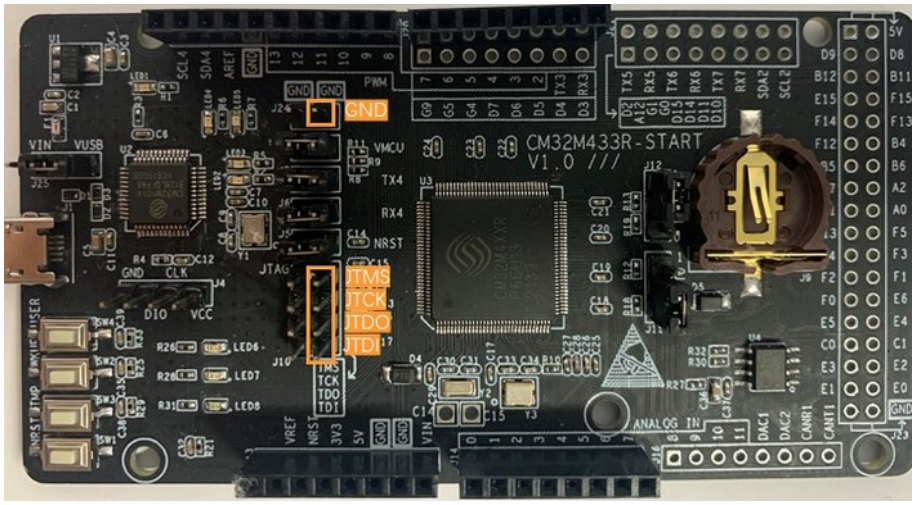
Debugging the CM32M433R-START with the Hummingbird debugger

Original post: https://www.rvmcu.com/quickstart-show-id-18.html

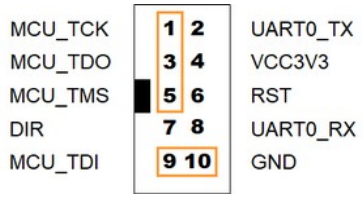
This document guides users on how to connect the Hummingbird debugger and the CM32M433R-START development board to run and debug applications with Nuclei Studio. It is intended for users who already have a certain background. For a detailed introduction to the process of downloading and installing Nuclei Studio IDE, compiling, debugging and running applications, please refer to CM32M433R-START quick start guide.

1. Connect the Hummingbird debugger to the CM32M433R-START

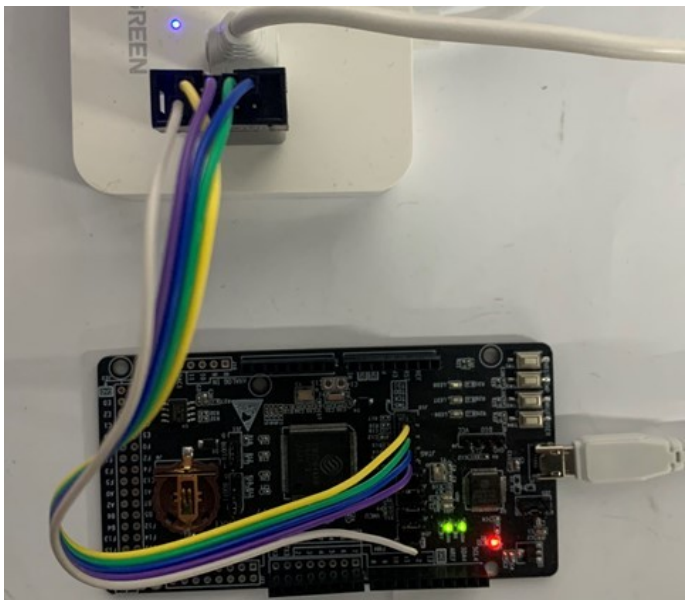
To connect the Hummingbird debugger, you need to remove the five jumpers shown on the picture below.



The pinout of the Hummingbird debugger connector is shown in the diagram below. The pins in the orange boxes are to be connected to the pins of the development board with matching names. Note that the connections to the development board must be made on the pins closest to the CM32M433R, as indicated by the orange boxes on the picture above.



When in use, the Micro USB cable also needs to be connected to provide power supply and serial port function as illustrated below.



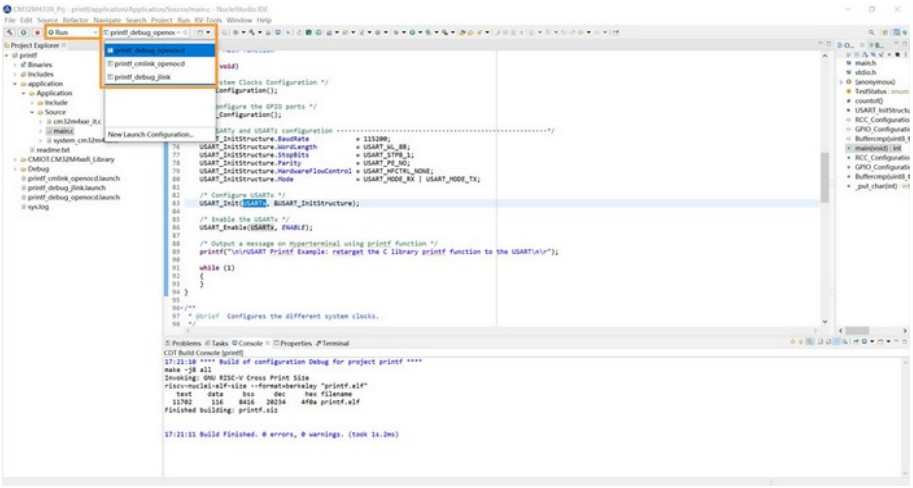
Notes:

- The above connection method only uses the JTAG debugging function of the Hummingbird debugger. If you want to also use the serial port function of the Hummingbird debugger, you can remove the two jumpers corresponding to TX4 and RX4 on the development board, and then connect the UART0_TX and UART0_RX to the corresponding pins of the development board closest to the CM32M433R (TX4—UART0_TX, RX4—UART0_RX).
- In addition, the CM32M433R chip can also be powered through the Hummingbird debugger. First remove the shorting cap corresponding to the VMCU on the development board, and then connect the VCC3V3 of the Hummingbird debugger to the corresponding pin of the development board closest to the CM32M433R, so no longer need to use the Micro USB cable.

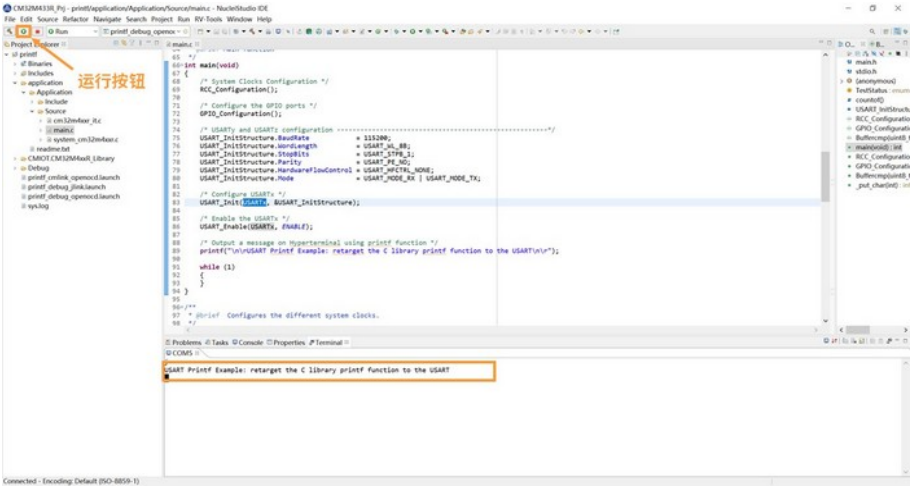
2. Run the printf application

Please refer to CM32M433R-START quick start guide, for a detailed explanation of how to create and build the printf project.

In the "Launch Bar" under the Nuclei Studio IDE menu bar, set the mode to "Run" and configure it to "printf_debug_openocd", as shown below.

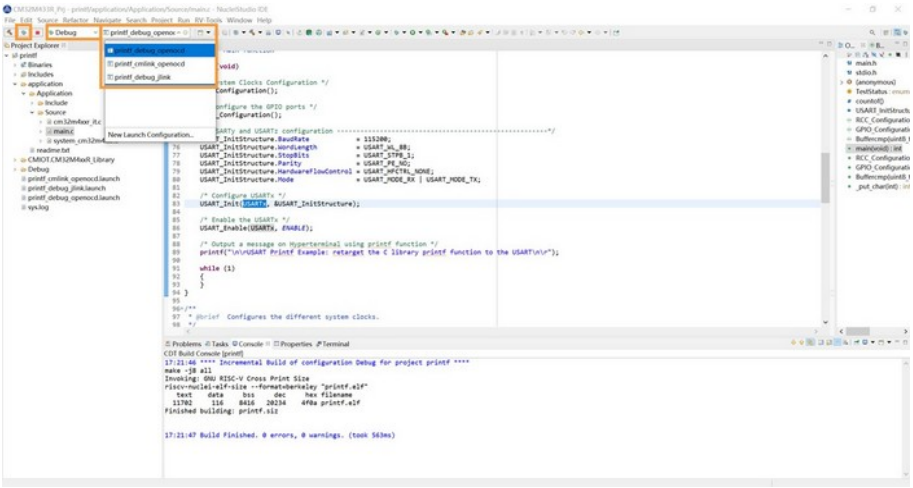


Then, click the green "Run" button to start downloading the program to the CM32M433R-START development board, and run the sample project. You should see the output shown below.

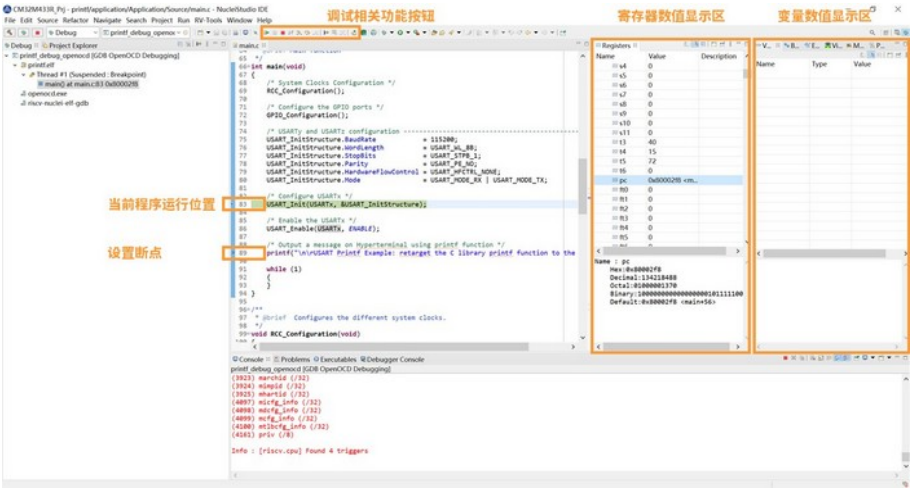


3. Debug the printf application

In the "Launch Bar" under the Nuclei Studio IDE menu bar, set the mode to "Debug" and configure it to "printf_debug_openocd" as shown below.



Then, click the "Beetle" button to start downloading the program to the CM32M433R-START development board and enter the debugging mode, as shown below.



Debugging the CM32M433R-START with a J-Link debugger

Original post: https://www.rvmcu.com/quickstart-show-id-18.html

This document guides users on how to connect the J-Link debugger and the CM32M433R-START development board to run and debug applications with Nuclei Studio. It is intended for users who already have a certain background. For a detailed introduction to the process of downloading and installing Nuclei Studio IDE, compiling, debugging and running applications, please refer to CM32M433R-START quick start guide.

1. J-Link driver installation

To use J-Link, you need to install the driver first. Visit the J-Link page on the SEGGER official website and download the latest version of the appropriate driver as shown below.

It is also necessary to install the J-Link support package for the CM32M433R-START development board:

- Windows version: https://github.com/CMIOT-XinShengTech/CM32M4xxR-Tools/blob/main/CM32M4xxR%20Support%20Pack/CM32M4xxR-Support-Pack-v1.0.2-win32-x32.exe
- Linux version: https://github.com/CMIOT-XinShengTech/CM32M4xxR-Tools/blob/main/CM32M4xxR%20Support%20Pack/CM32M4xxR-Support-Pack-v1.0.2-linux-x64.run

Under Linux, run the following commands:

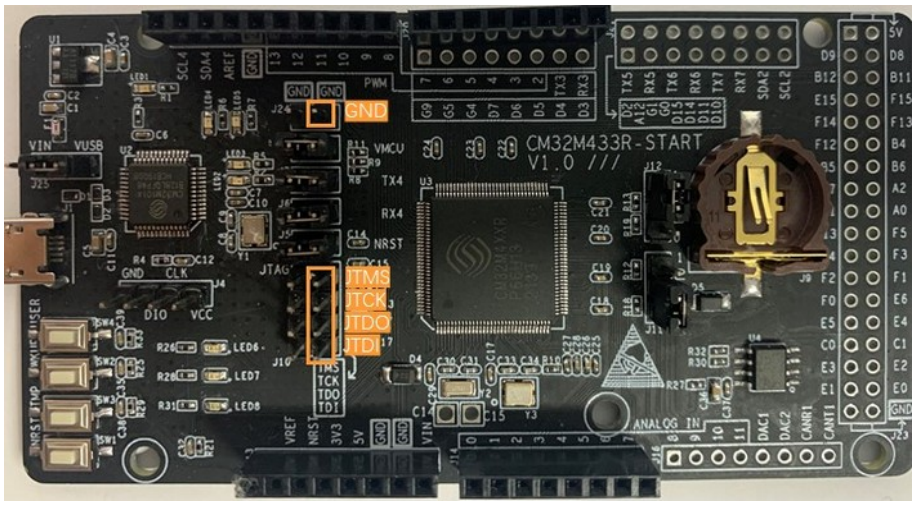
```
chmod +x CM32M4xxR-Support-Pack-v1.0.2-linux-x64.run
sudo ./CM32M4xxR-Support-Pack-v1.0.2-linux-x64.run
```

Under Windows, start the installer, select J-Link and proceed with the installation.

Note: the path here needs to be consistent with the J-Link driver installation path.

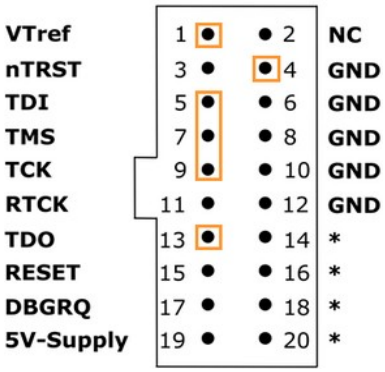
2. Connect the J-Link debugger to the CM32M433R-START

To connect the J-Link debugger, you need to remove the five jumpers shown on the picture below.

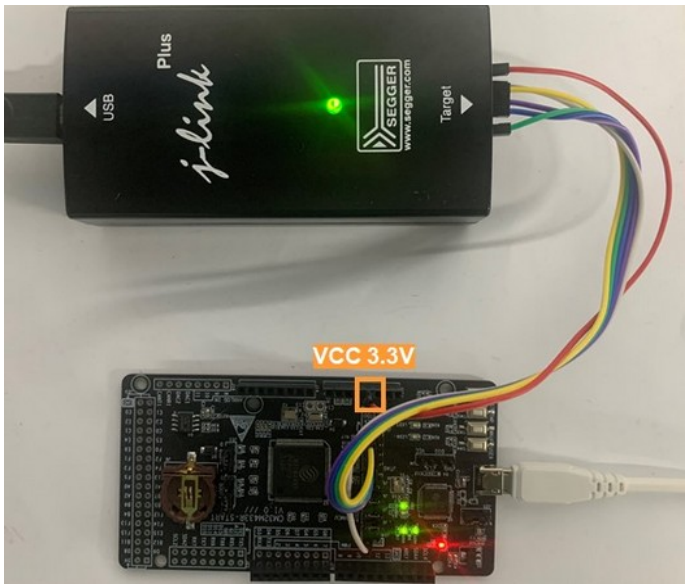


The pinout of the J-Link connector is shown in the diagram below. The pins in the orange boxes are to be connected to the pins of the development board with matching names. Note that the connections to the development board must be made on the pins closest to the CM32M433R, as indicated by the orange boxes on the picture above.

The VTref pin should be connected to the 3V3 header of the CM32M433R-START development board.



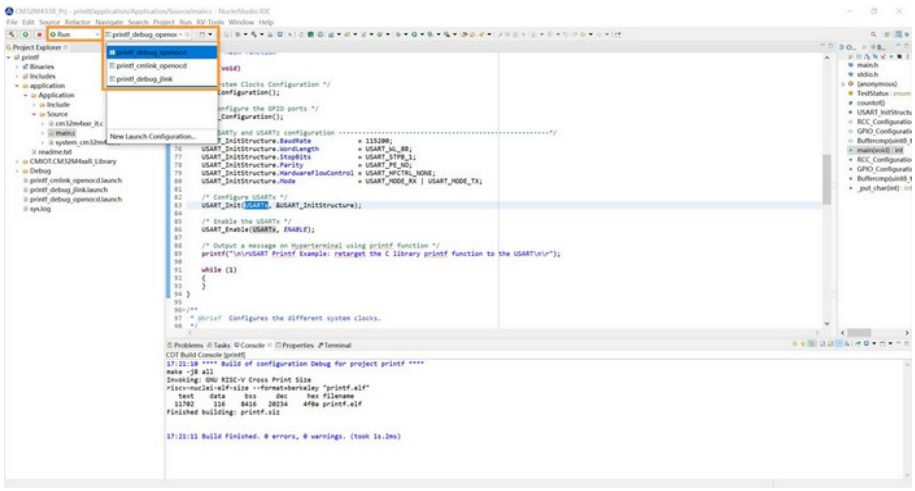
When in use, the Micro USB cable also needs to be connected to provide power supply and serial port function as illustrated below.



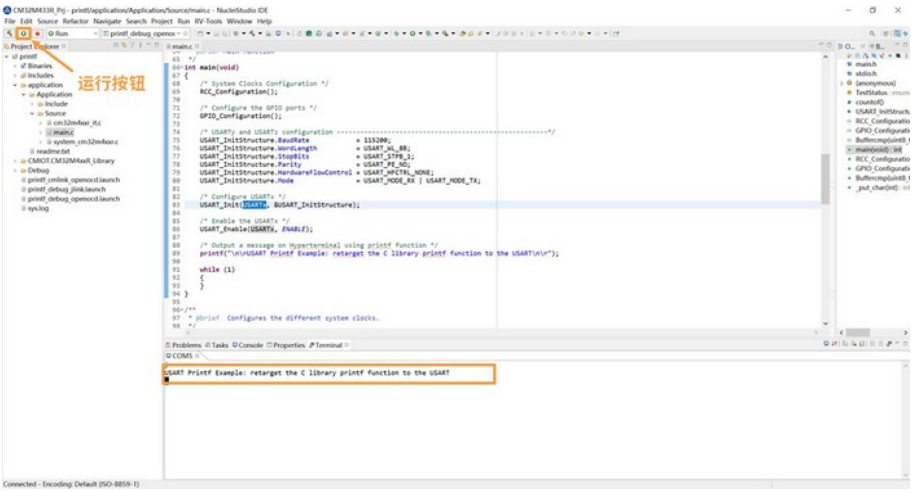
3. Run the printf application

Please refer to CM32M433R-START quick start guide, for a detailed explanation of how to create and build the printf project.

In the "Launch Bar" under the Nuclei Studio IDE menu bar, set the mode to "Run" and configure it to "printf_debug_openocd", as shown below.

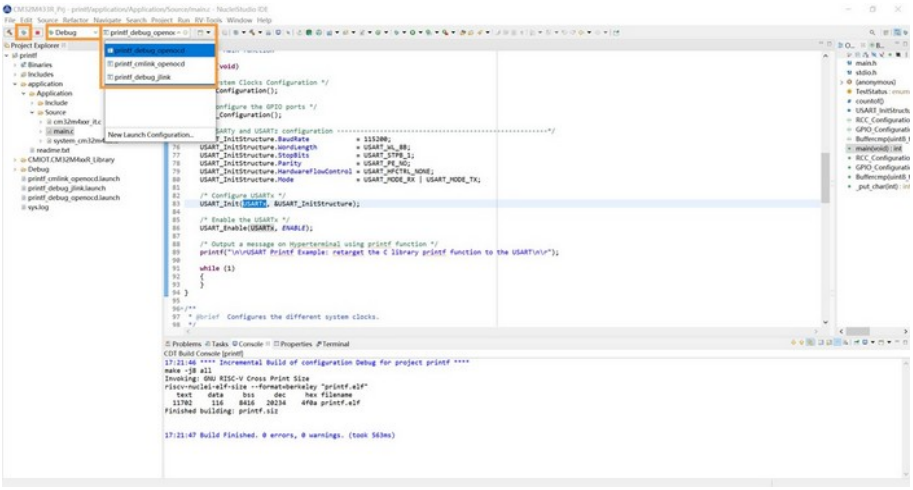


Then, click the green "Run" button to start downloading the program to the CM32M433R-START development board, and run the sample project. You should see the output shown below.

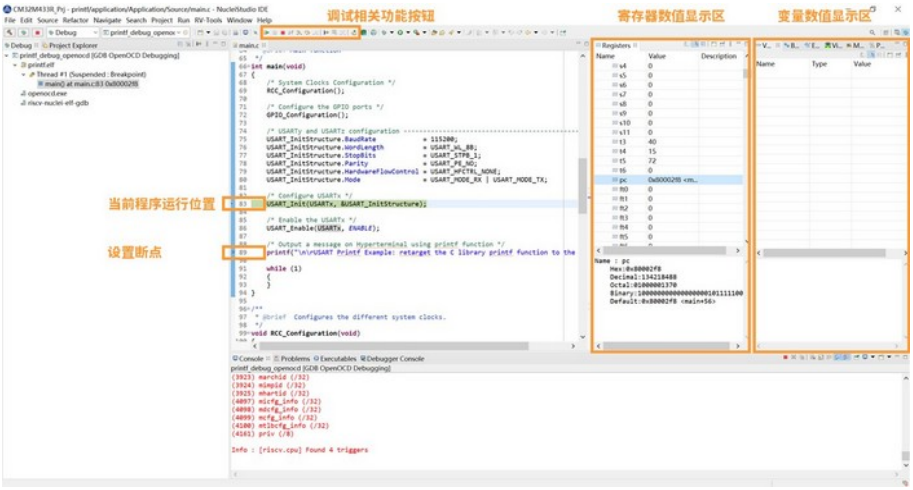


4. Debug the printf application

In the "Launch Bar" under the Nuclei Studio IDE menu bar, set the mode to "Debug" and configure it to "printf_debug_openocd" as shown below.



Then, click the "Beetle" button to start downloading the program to the CM32M433R-START development board and enter the debugging mode, as shown below.


调试相关功能按钮
寄存器数值显示区
变量数值显示区
当前程序运行位置
设置断点

Reference documents

- CM32M433R reference manual: https://www.rvmcu.com/app/quickstart/skins/default/doc/CM32M4xxR-user-guide-V1.4.pdf
- CM32M433R data sheet: https://www.rvmcu.com/app/quickstart/skins/default/doc/CM32M4xxR-datasheet-V1.4.pdf
- Other documents (e.g. application notes, libraries) from Xinsheng: https://www.xinshengcmiot.cn/service/download
- Nuclei Instruction Set Architecture Manual: https://doc.nucleisys.com/nuclei_spec/
- Nuclei N300 series processor core data book: https://www.nucleisys.com/upload/file/2020/02/1582893657-2424.pdf
- Other documents available at: https://user.nucleisys.com/ (registration required)
- Development tools: https://nucleisys.com/download.php
- Nuclei Studio user manual: https://www.rvmcu.com/nucleistudio-userguide-id-25.html
- CM32M433R-START quick start guide: https://www.rvmcu.com/quickstart-show-id-15.html (also translated in this document)
- CM32M433R-START development board schematic: https://www.rvmcu.com/app/quickstart/skins/default/doc/cm32m433r-start-sch.pdf