W55MH32 User Manual

Version 1.0.0

http://www.wiznet.io/

Directory

List of Figures

List of Tables

1 EVB Introduction

1.1 W55MH32L-EVB

1.1.1 Introduction

The W55MH32L-EVB is a development board based on the W55MH32L chip. It features a core clock speed of 216 MHz, 1 MB of flash memory, and 96 KB of SRAM. Additionally, it is equipped with a fully integrated hardware TCP/IP offload engine, enabling Ethernet applications to be implemented with simple socket programming.

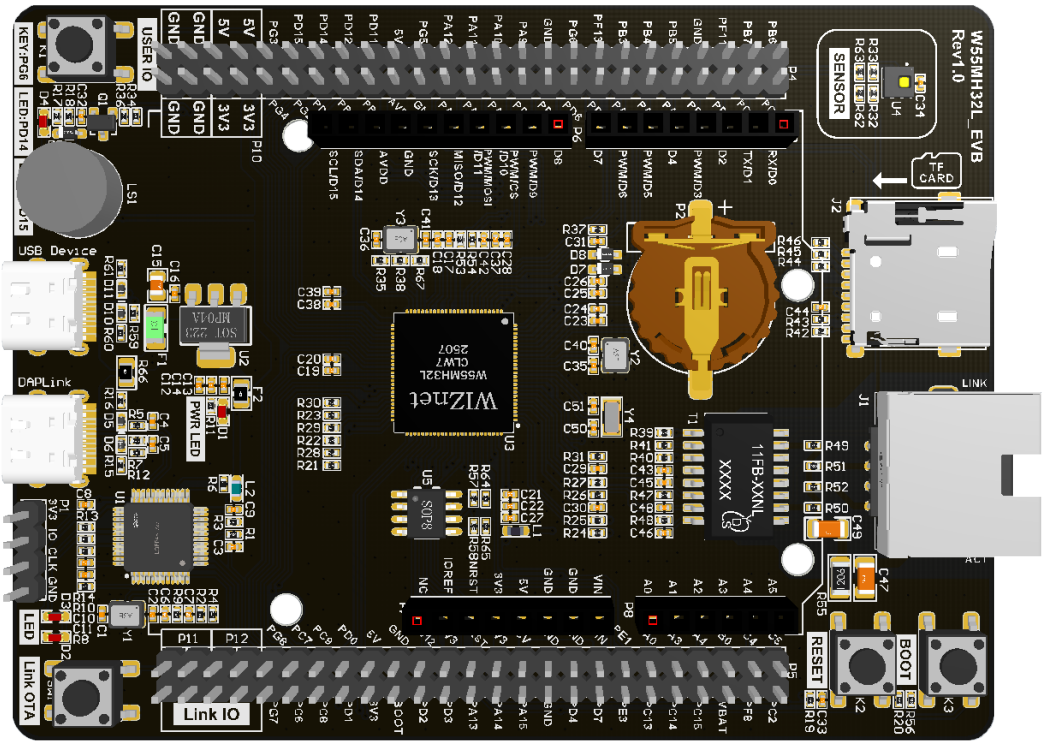

Figure 1 Top View

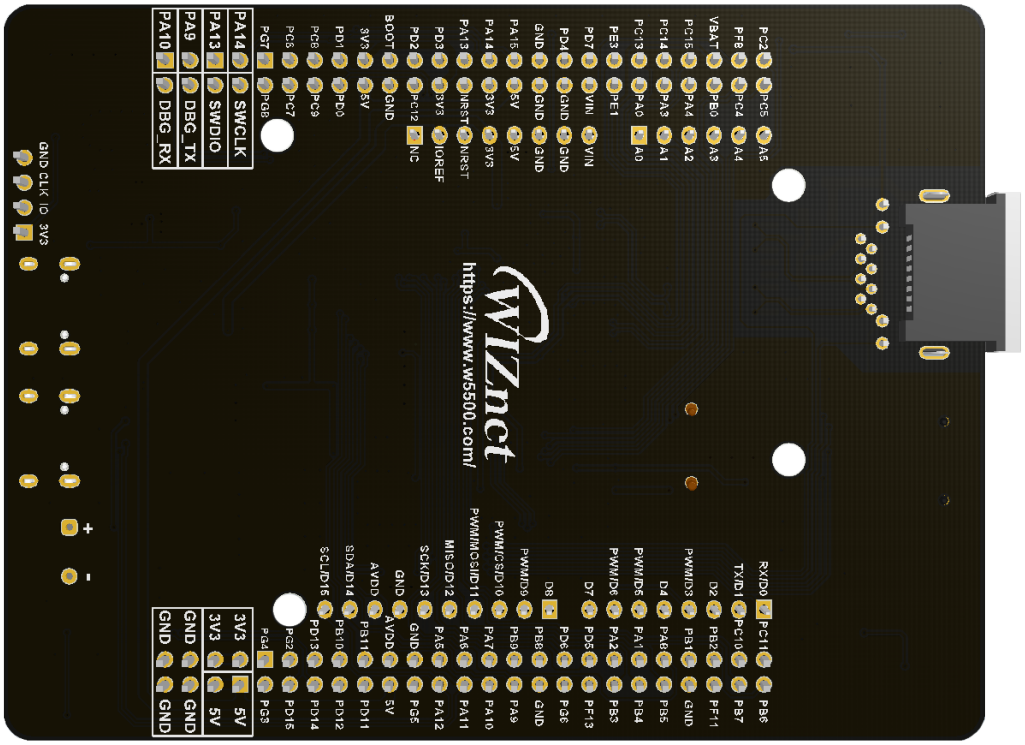

Figure 2 Bottom View

1.1.2 Onboard Resources

The development board integrates an ARDUINO header, allowing connection to ARDUINO-compatible expansion boards, offering excellent compatibility and scalability.
Additionally, the board features a built-in WIZ-Link emulator, supporting debugging, program downloading, and virtual serial port functions. This eliminates the need for external emulators and serial debugging tools, enabling effortless learning of microcontroller and Ethernet applications with just the development board.
Other onboard peripherals include a TF card slot, a temperature and humidity sensor, EEPROM, user buttons, user LEDs, and more, providing abundant resources.

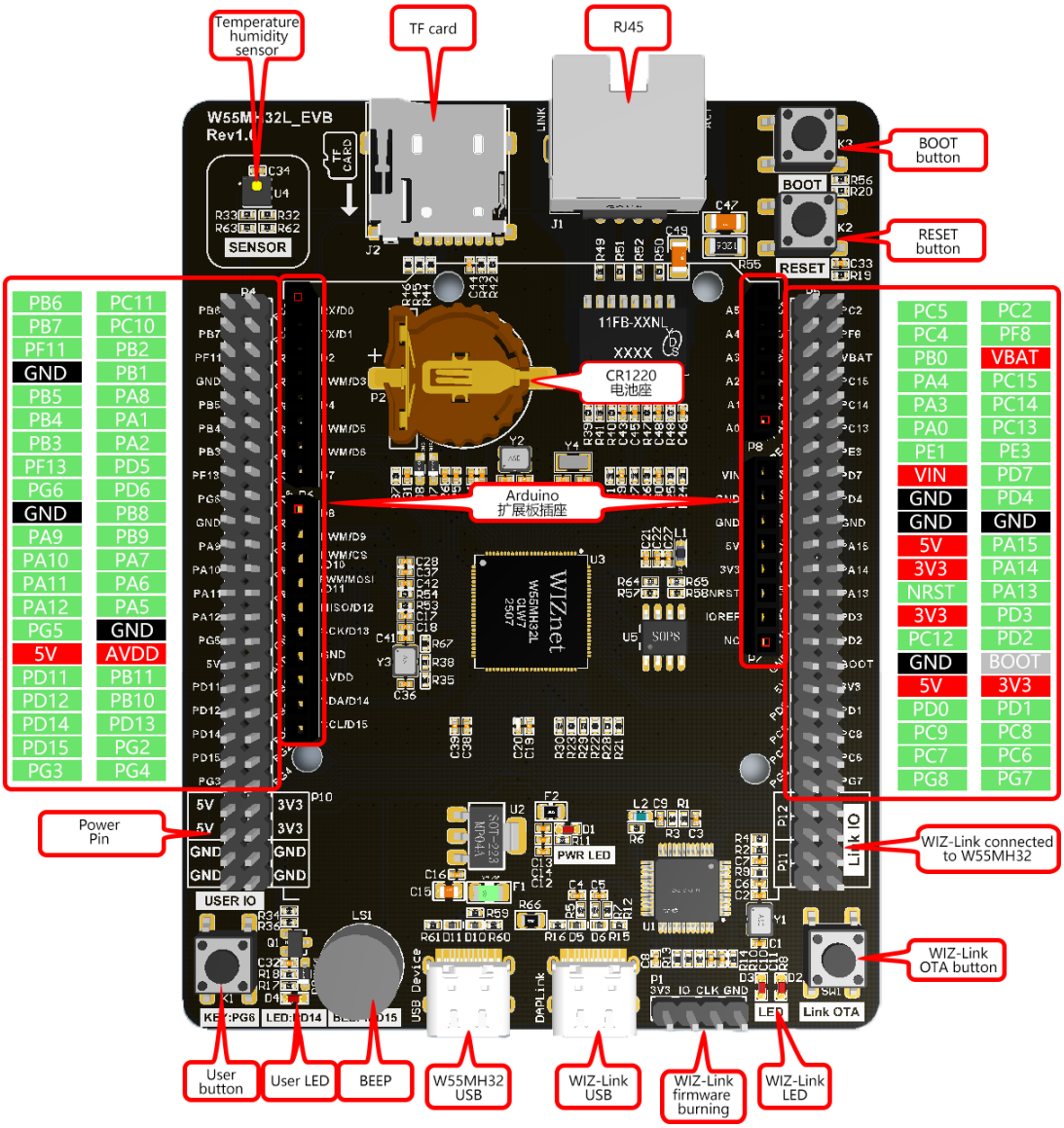

Figure 3 Hardware Resource Configuration Diagram

Table 1 Hardware Resource Configuration Table

Resources	Description
MCU	W55MH32L,LQFP100,1MB FLASH,96KB RAM
Clock Frequency	216MHz
GPIO	*61
USART	*5
SPI	*2
I2C	*2
USB	*1

CAN	*1
SDIO	*1
Ethernet	*1
12 bit ADC	*3 (12 channel)
12 bit DAC	*2 (2 channel)
TRNG	Support
Hardware encryption algorithm unit	Support
RTC	CR1220 Battery Socket *1
Buzzer	Active Buzzer*1
Button	RESET*1, BOOT*1, USER*1, WIZ-Link OTA*1
LED	PWR*1, USER*1, WIZ-Link*2
Emulator	WIZ-Link*1
Sensor	AHT20*1
EEPROM	AT24C16C-SSHM-T*1
ARDUINO Header Pin	Support
USB	W55MH32*1,WIZ-Link*1
Ethernet Interface	Sinking RJ45*1

1.2 W55MH32Q-EVB

1.2.1 Introduction

W55MH32Q-EVB is a minimal development board based on the W55MH32Q chip, with a clock speed of 216MHz, 1MB of flash memory, and 96KB of SRAM. It also has a complete hardware TCP/IP offloading engine, and can be used for Ethernet applications with simple socket programming.

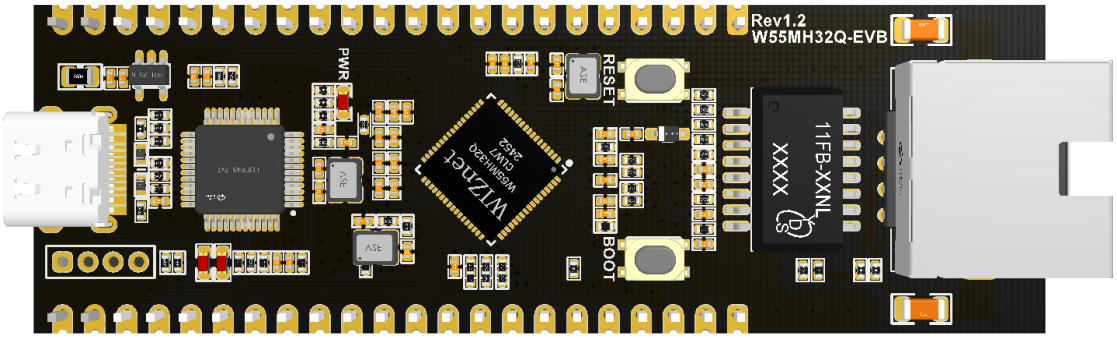

Figure 4 Top View

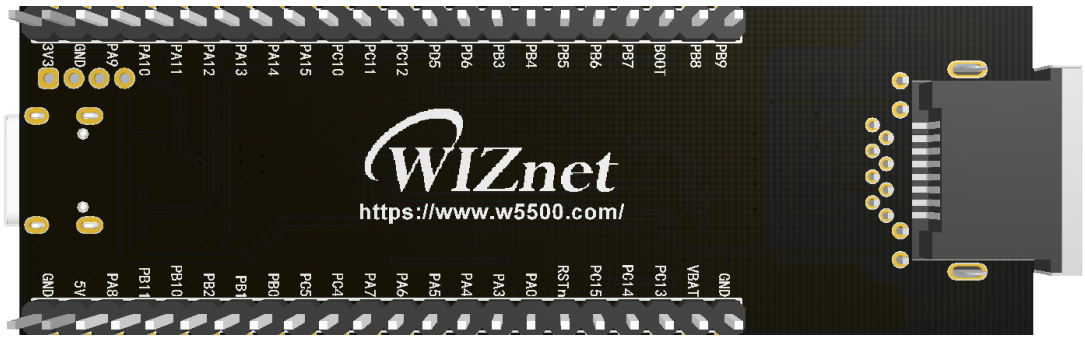

Figure 5 Bottom View

1.2.2 Onboard Resources

All IO of the chip are led out to the development board pin, and the chip's functions can meet the vast majority of embedded application scenarios and development needs. Its small size makes it easy to test the chip and evaluate its functions. It also integrates a WIZ Link emulator, which supports debugging, program downloading, and virtual serial port functionality. No need for external simulators and serial debugging tools anymore, just one development board can easily handle the learning content of microcontroller+Ethernet.

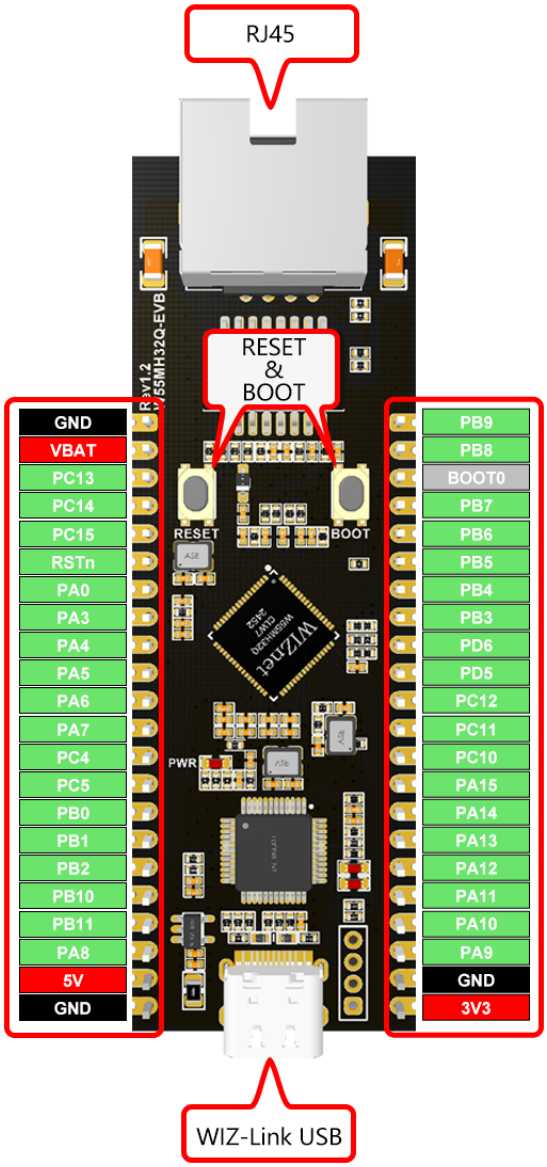

Figure 6 Hardware Resource Configuration Diagram

Table 2 Hardware Resource Configuration Table

Resources	Description
MCU	W55MH32Q,QFN68,1MB FLASH,96KB RAM
Clock Frequency	216MHz
GPIO	*36
USART	*3
SPI	*2
I2C	*2
USB	*1
CAN	*1

Ethernet	*1
12 bit ADC	*3 (12 channel)
12 bit DAC	*2 (2 channel)
TRNG	Support
Hardware encryption algorithm unit	Support
Button	RESET*1, BOOT*1
LED	PWR*1
USB	W55MH32*1
Ethernet Interface	Sinking RJ45*1

2 Install IDE

2.1 Friendly Reminder

1. The installation path must not contain Chinese characters; it must be an English-only path.
2. The installation directory must not conflict with the directories of 51's KEIL or KEIL4. All three directories must be separate.
3. The installation of KEIL5 requires an additional step compared to KEIL4: you must add the MCU library; otherwise, it cannot be used.

2.2 Getting the KEIL5 Installation Package

The W55MH32 development board supports development with the KEIL5 MDK (Microcontroller Development Kit). You can visit the official website at www.keil.com to download the installation package.

Figure 7 KEIL Official Website Installation Package Illustration

2.3 Start Installation KEIL5

Double-click the downloaded installation package and click the "Next" button to begin the installation.

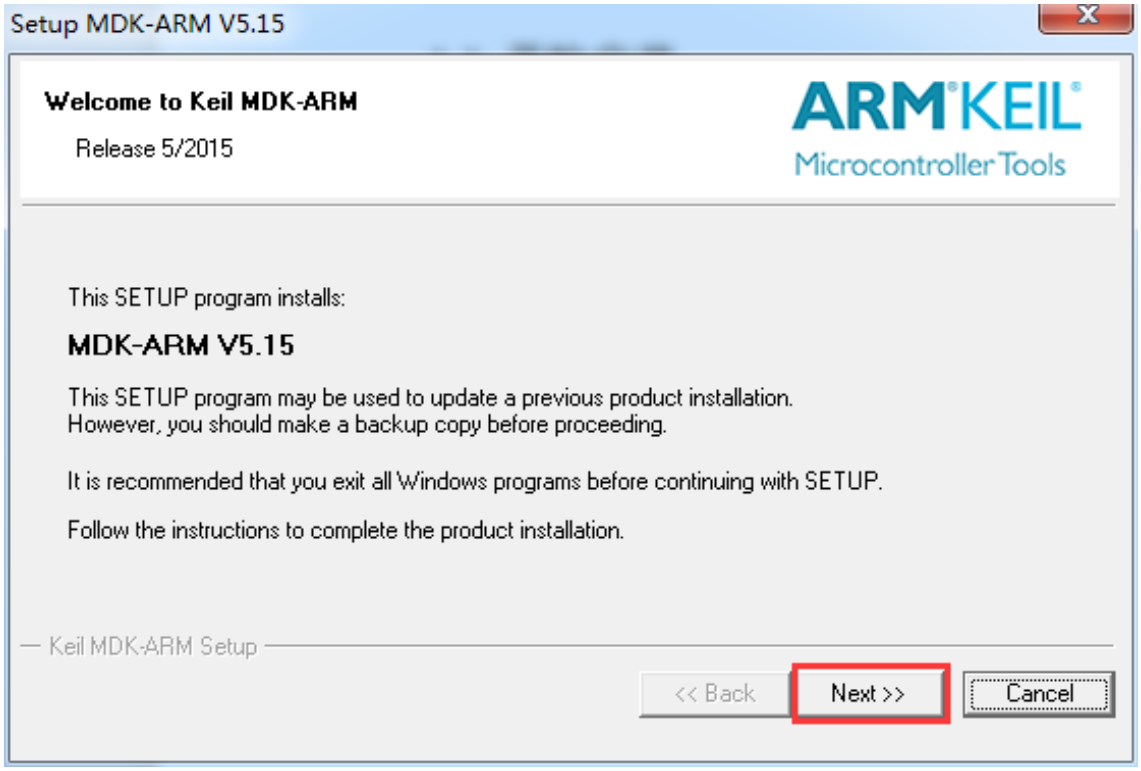

Figure 8 Start Installation KEIL5

Follow the on-screen instructions step by step for the installation. Once the installation is complete, it should appear as shown in the image below.

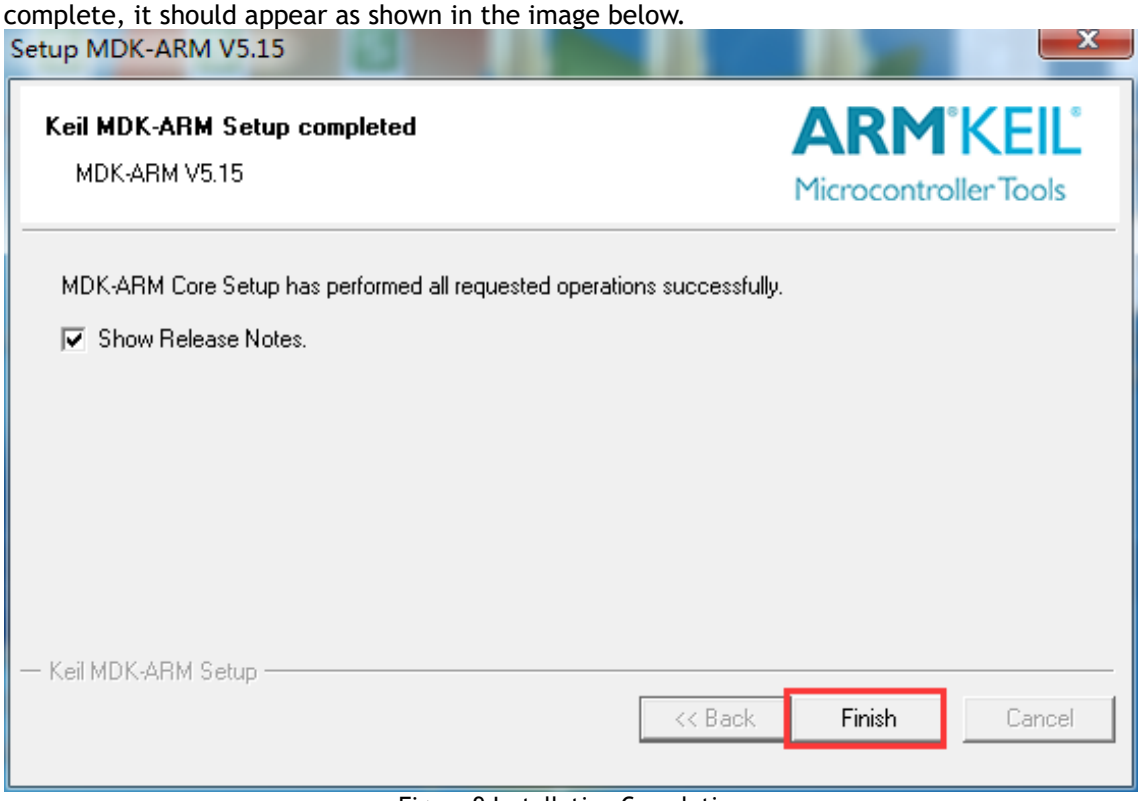

Figure 9 Installation Completion

2.4 Install the W55MH32 Chip Pack

Open the W55MH32_DFP.1.1.0.pack file in the software package directory, then click "Next" to complete the installation.

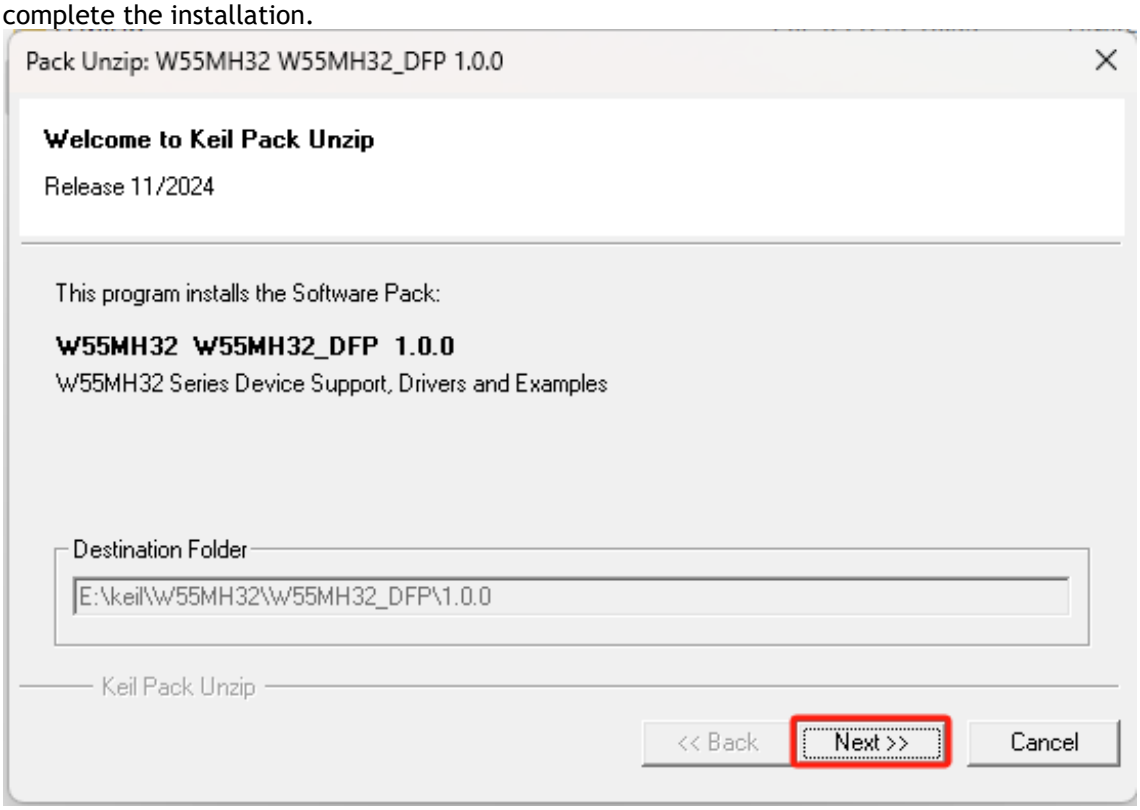

Figure 10 Install pack

3 Open the Example Project

3.1 Peripheral Example Directory

Open the "SDK\ModuleDemo" folder in the documentation package. It contains most of the peripheral examples as well as TOE engine examples.

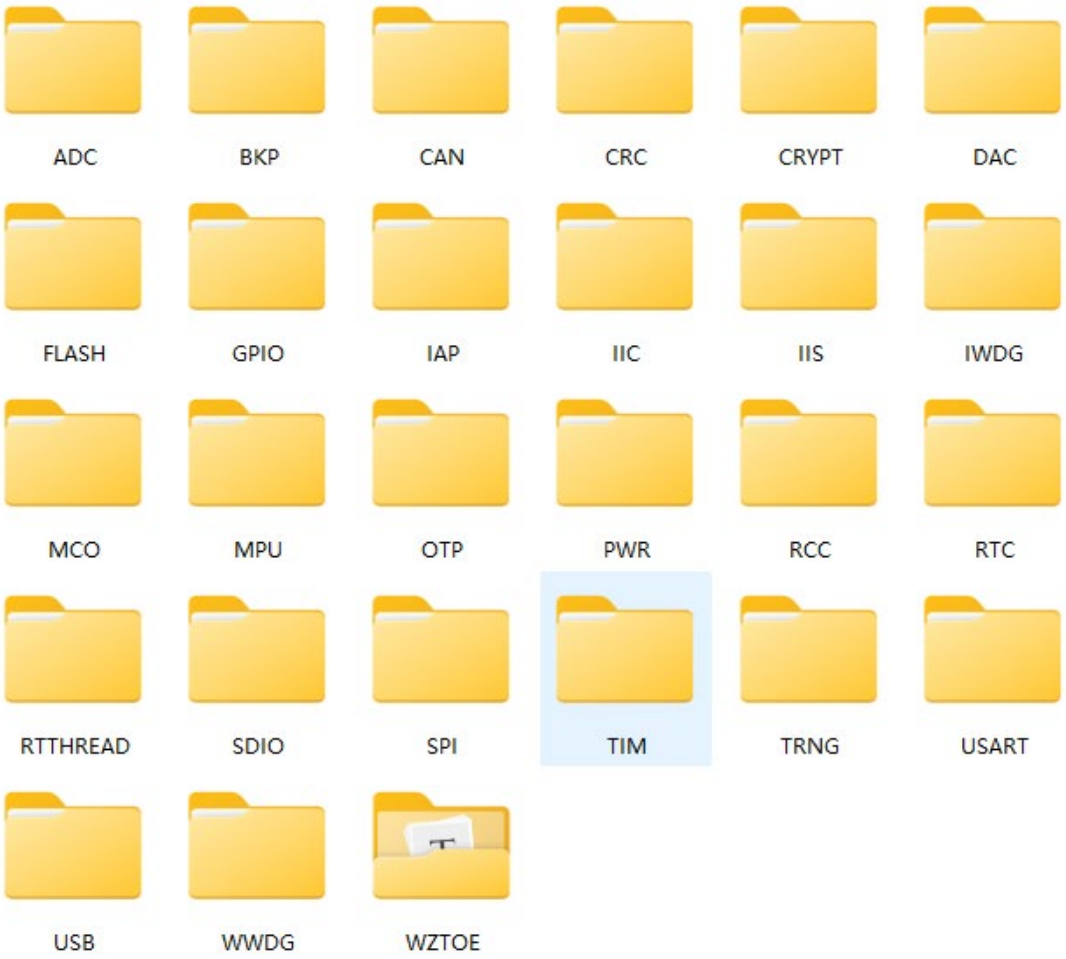

Figure 11 Example Program

3.2 TOE Engine Example Directory

The TOE engine example directory includes various network protocol applications.

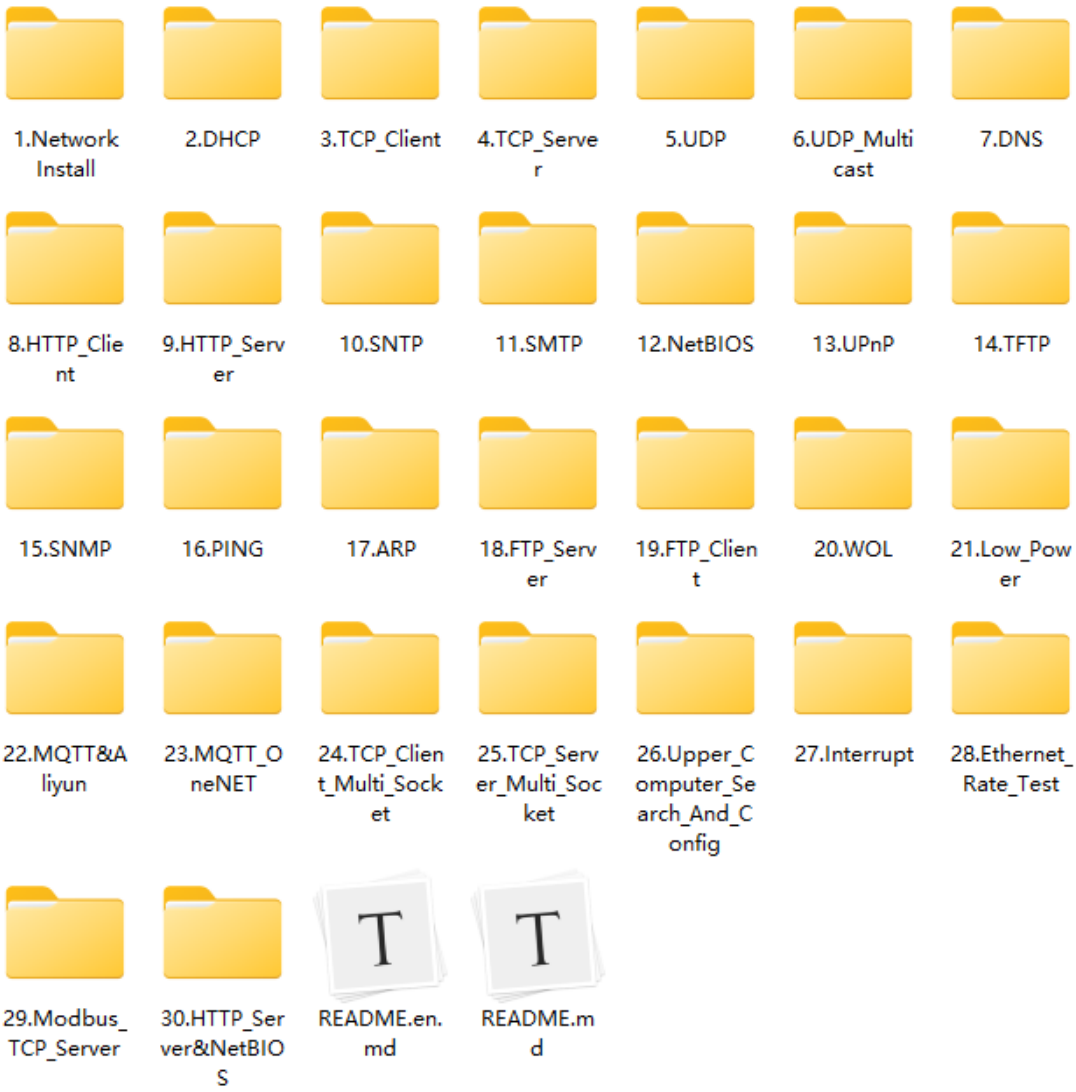

Figure 12 TOE Engine Example Directory

3.3 Introduction to the KEIL Main Interface

The KEIL project file has the extension .uvprojx. The KEIL project files for peripheral examples are located in the corresponding peripheral example's USER directory, while the KEIL project files for TOE examples are found in the Project directory.
Double-click to open the project, and the main KEIL5 window will appear, as shown in the image below.

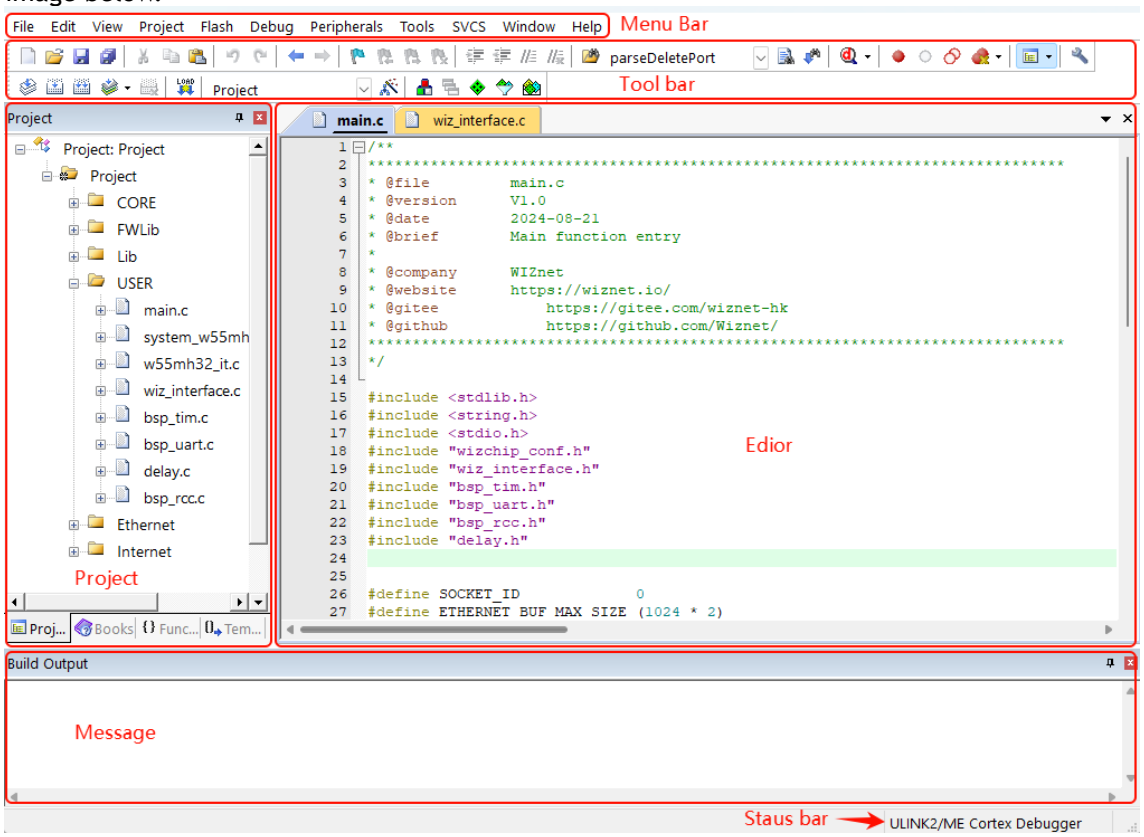

Figure 13 KEIL5 Main Interface

- **Menu Bar**: Contains options such as File, Edit, View, Project, Help, and more.
- **Tool bar**: Provides quick access to commonly used tools; some key tools will be highlighted below.
- **Project** Window: Displays the project contents, including file groups, source files, header files, etc.
- **Editor** Window: The area where you write and edit your code.
- **Message** Window: Displays compilation messages, programming information, and other feedback.
- **Status Bar**: Shows the cursor's position (line and column), character encoding, keyboard Num Lock status, and other system information.

3.4 Common Shortcut Keys

Three Types of Common Tool Shortcut Keys

Figure 14 Common Shortcut Keys

1. **Simulation Tool Shortcut Buttons**: Used during simulation, these buttons serve the purpose of marking power-off states.

Table 3 Simulation Tool Shortcut Buttons

Name	Shortcut Keys
Start/Stop Debug Session	Ctrl+F5
Insert/Remove Breakpoint	F9
Enable/Disable Breakpoint	Ctrl+F9
Disable All Breakpoints	-
Kill All Breakpoints	Ctrl+Shift+F9

2. **Compilation Tool Shortcut Buttons**: Used for compiling and downloading code.

Table 4 Compilation Tool Shortcut Buttons

Name	Shortcut Keys
Translate	Ctrl+F7
Build	F7
Rebuild	-
Batch Build	-
Stop Build	-
Download	F8

3. **Project Target Options (also known as the Magic Wand)**: This refers to the configuration settings for the project target, such as selecting the chip device, C/C++ options, simulation settings, and more.

4 Introduction to the Emulator

4.1 Introduction

The development board integrates a WIZ-Link emulator, as shown in Figure 12. WIZ-Link follows ARM's CMSIS-DAP standard and supports both program downloading and online simulation. It is driver-free and compatible with four operating systems: XP, WIN7, WIN8, and WIN10. It also supports direct downloading with KEIL and IAR.
Additionally, WIZ-Link supports a virtual serial port, which is internally connected to the PA9 and PA10 pins of the W55MH32.
Downloading, logging, and simulation can all be performed through WIZ-Link, making it very convenient.

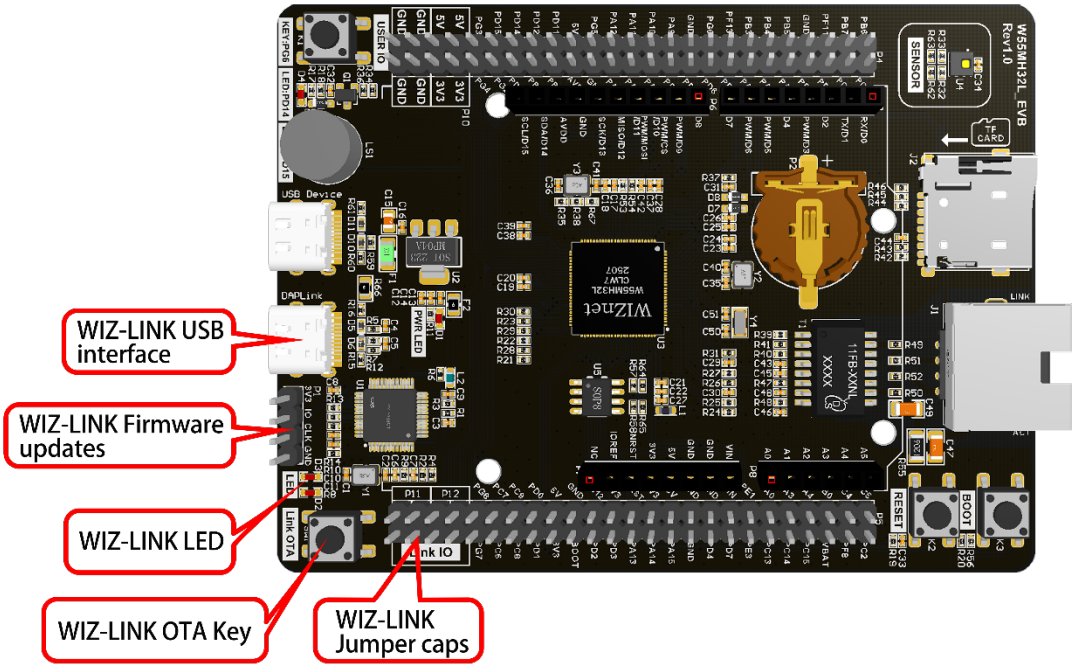

Figure 15 WIZ-Link

4.2 Emulator Configuration

After connecting the development board to the computer via WIZ-Link, open the Device Manager on the computer. You will be able to see the WIZ-Link device and the virtual serial port. As shown in the figure below.

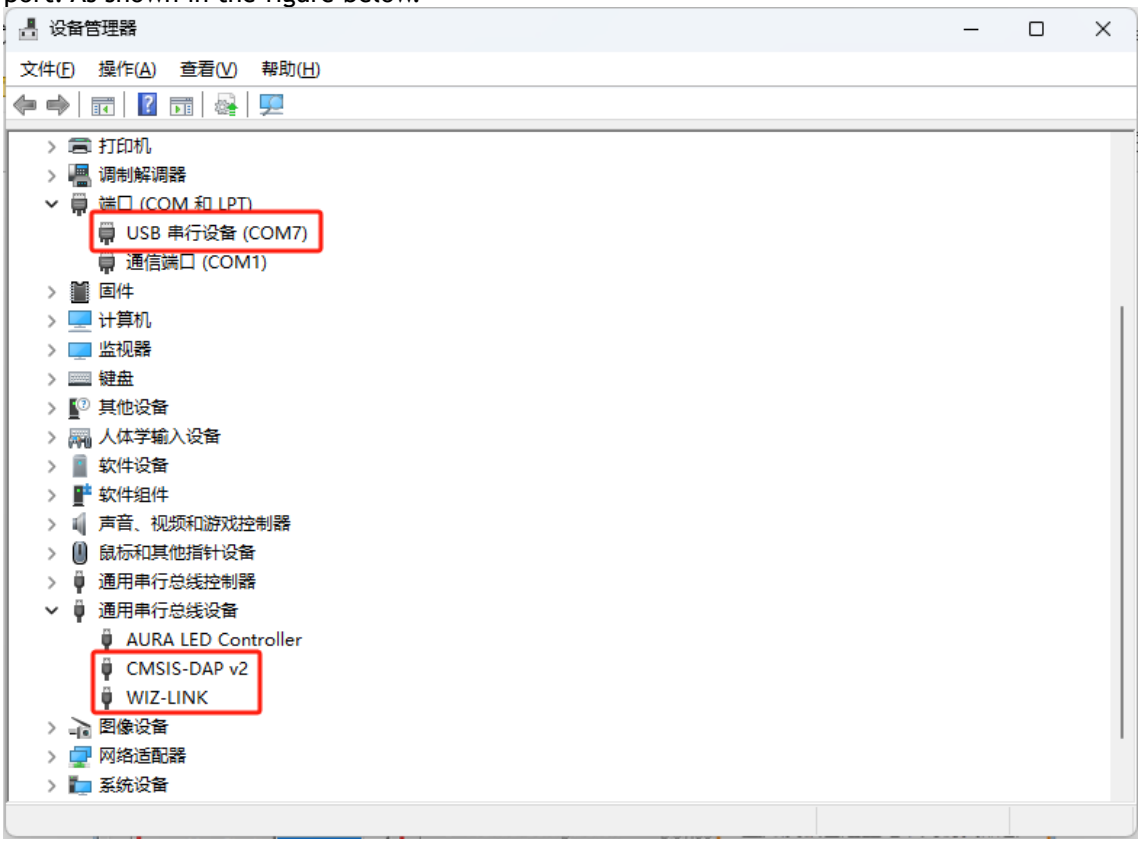

Figure 16 WIZ-LinkDevice Manager Illustration

Open KEIL, and in the Target Options tab (also known as the Magic Wand), select the CMSIS-DAP Emulator as shown in the figure below.

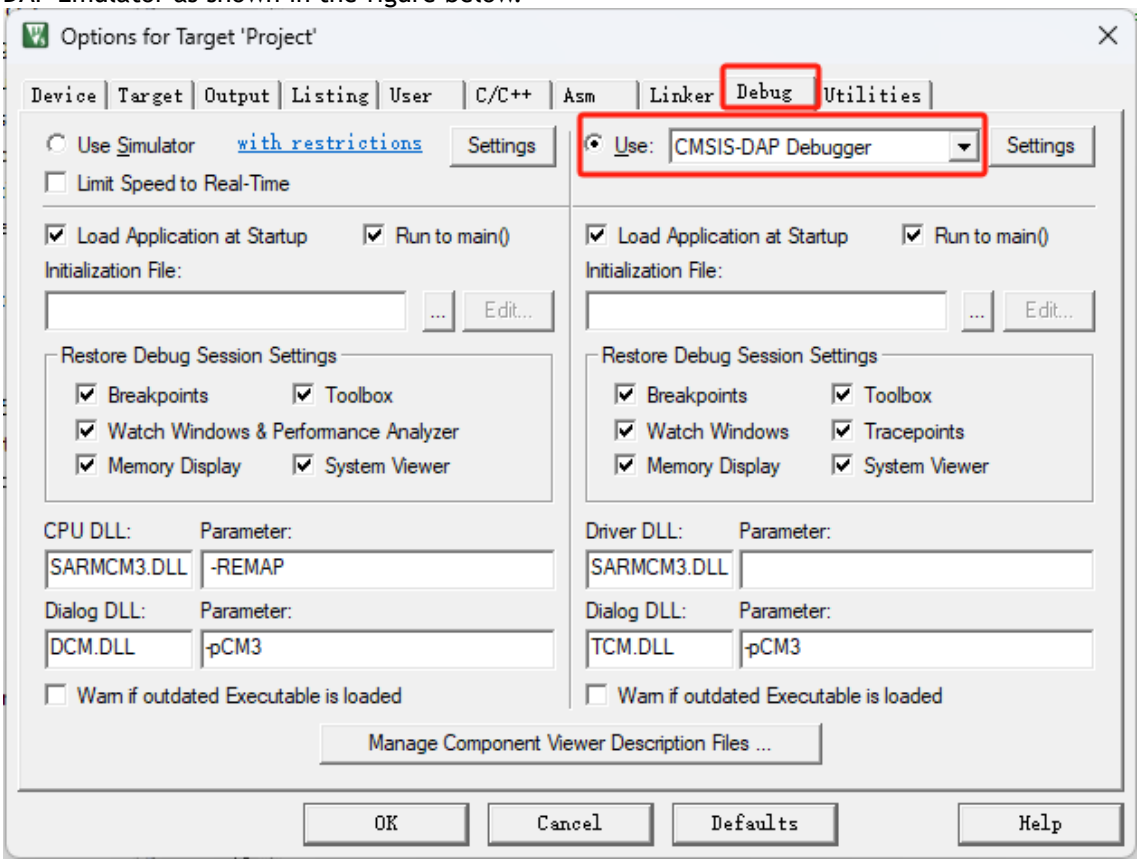

Figure 17 Debug Option Configuration

After clicking the Settings button, you will enter the CMSIS-DAP Debugger Configuration window.

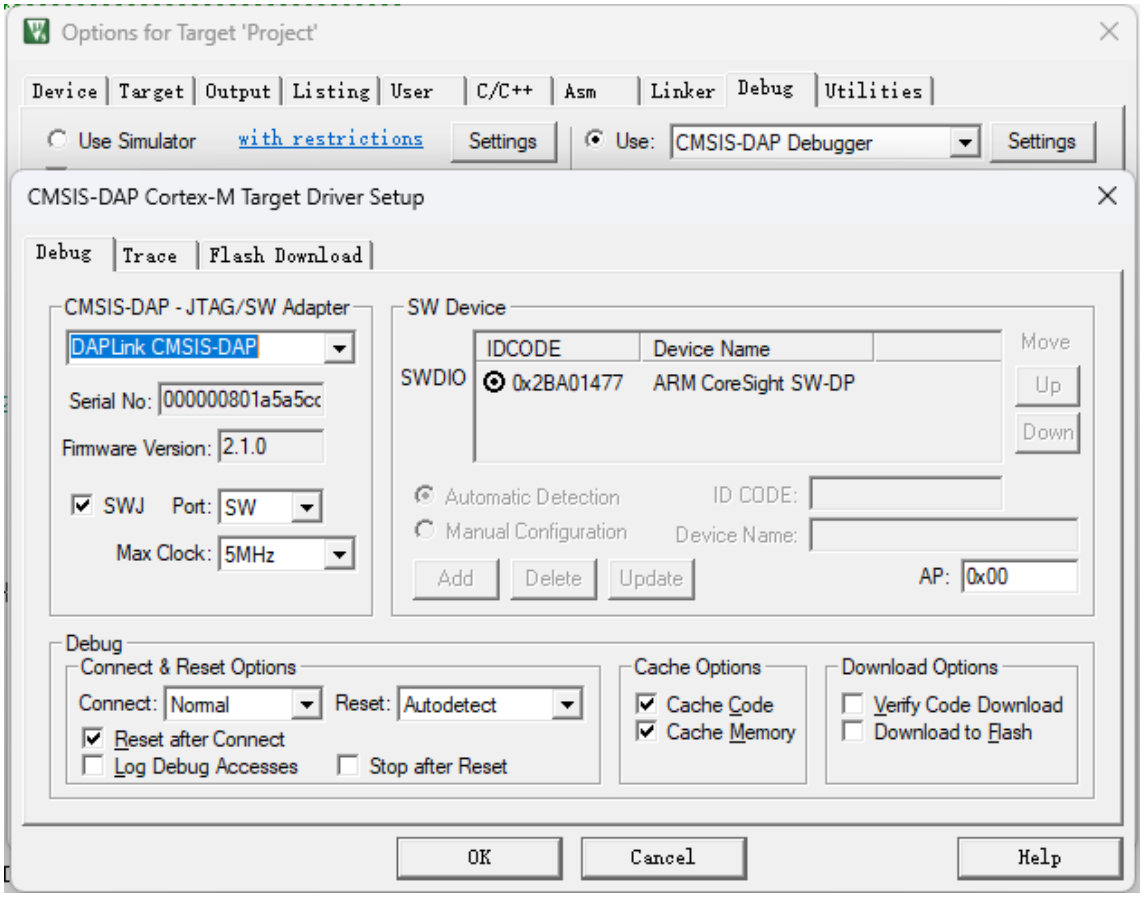

Figure 18 CMSIS-DAP Debugger Configuration

Note: If the Device Manager recognizes the WIZ-Link, but the development board is not detected in the CMSIS-DAP Debugger settings window, try upgrading KEIL5 to the latest version.

4.3 Using Other Emulators and Serial Debugging Tools

The W55MH32 supports most emulators on the market, such as ST-LINK V2, JLINK, and ARM Emulator.

Note: It does not support ST-LINK V3.

To use a different serial debugging tool or emulator, the jumpers connecting the W55MH32 to the WIZ-Link serial and debugging ports on the development board can be removed. This will prevent simultaneous connections, as shown in Figure 16.

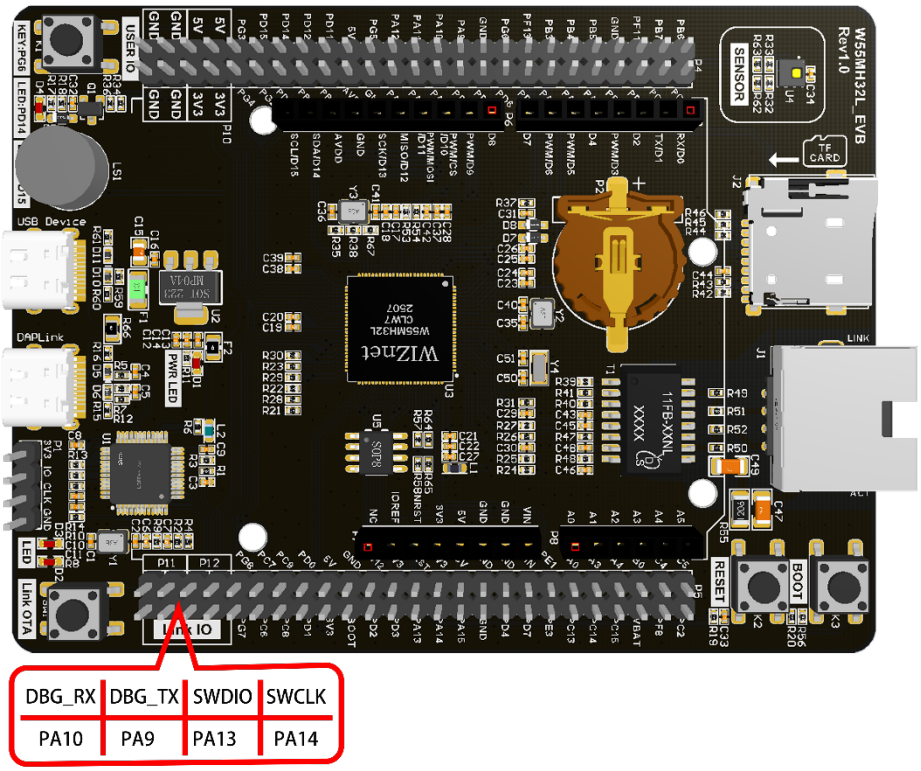

Figure 19 W55MH32 Jumper Cap Connection to WIZ-Link

When using JLINK, there may be a situation where the chip is not recognized in the JLINK configuration interface. In this case, the following changes need to be made.

1. Modify the JLinkDevices.xml file in the JLINK installation directory by adding the following content.

```
<Device>
    <ChipInfo Vendor="WIZnet" Name="W55MH32-100LQFP" Core="JLINK_CORE_CORTEX_M3"
    WorkRAMAddr="0x20000000" WorkRAMSize="0x00018000"/>
    <FlashBankInfo Loader="Devices/WIZnet/W55MH32/W55MH32.FLM"
    LoaderType="FLASH_ALGO_TYPE_OPEN" MaxSize="0x00100000" Name="Internal Flash"
    BaseAddr="0x08000000" AlwaysPresent="1"/>
</Device>
```

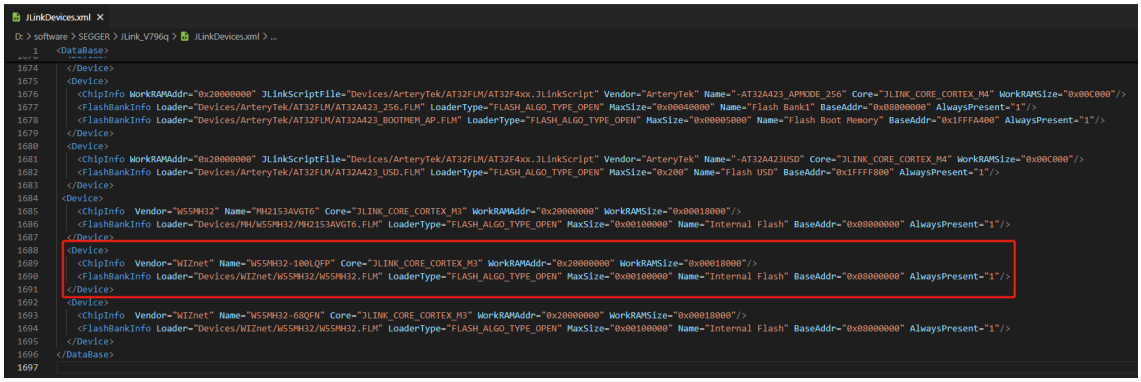

Figure 20 JLINK Modification Content Illustration

2. Add the W55MH32.FLM file under the path Devices/WIZnet/W55MH32/.

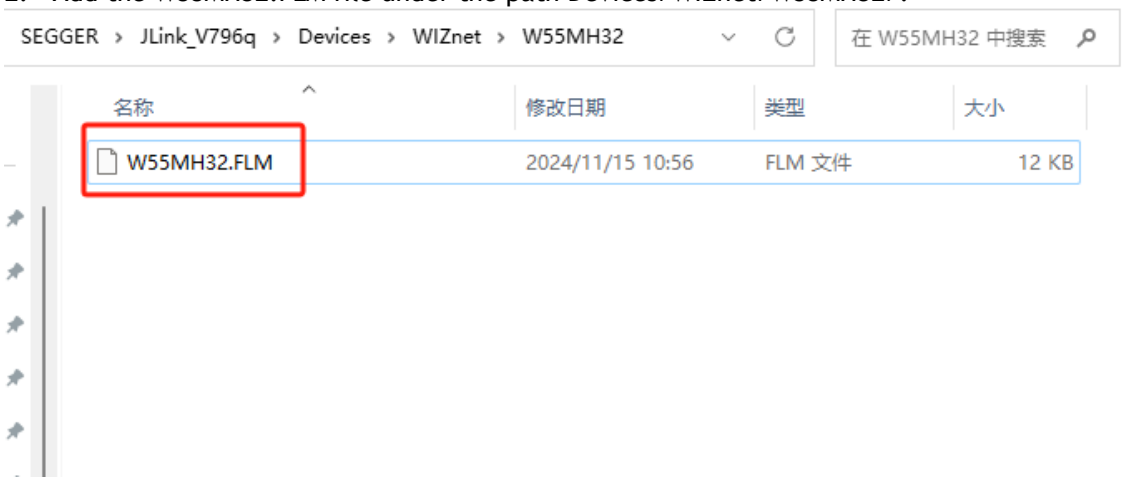

Figure 21 Add the W55MH32.FLM file

The W55MH32.FLM file is located in the KEIL installation directory under the path W55MH32 > W55MH32_DFP > 1.0.0 > CMSIS > Flash, as shown in the image below.

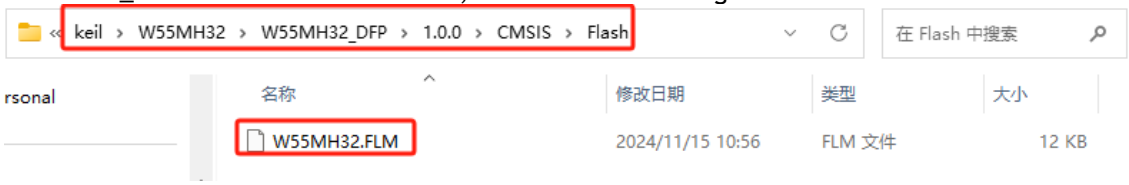

Figure 22 Location of the W55MH32.FLM file