ESC0830EVBT0.1 Development Board

User Manual

Revision History

Serial No.	Modified Content	Date	Reviser	Version
1	New	2024.12	Wu Shuping	V0.1
2	Added operation results, standardized document formatting, etc.	2025.05	Hu Jingjing	V0.2

Table of Contents

List of Figures

List of Tables

1. Purpose and Scope

This document specifies the test plan for the Mimic MCU chip, outlines the usage instructions for the development board, and details the corresponding operational procedures. This document applies to the ESC0830EVBT0.1 development board.

2. Chip Introduction

The embedded endogenous security controller chip adopts a heterogeneous multi-core CPU architecture, specifically tailored for embedded control and IoT scenarios. Relevant theoretical research demonstrates that this heterogeneous multi-core CPU architecture can significantly reduce the probability of MCU compromise, substantially enhancing security levels in many application scenarios.

Its primary function is to provide security protection between the external environment and the controller chip. By integrating three heterogeneous execution units (CPUs), the chip realizes MCU mimicry (the external view shows one CPU, while internally up to three heterogeneous CPUs execute tasks concurrently). This design makes it extremely difficult for attackers to effectively compromise the MCU, thereby ensuring the MCU's security.

Figure 2-1: Mimic MCU Chip

The embedded endogenous security controller chip features input/output (I/O) and functional characteristics compatible with mainstream MCUs on the market. It incorporates multiple heterogeneous microprocessor cores and a mimic negative

feedback controller, enabling multi-redundancy mimic decision-making.

Additionally, it is equipped with peripheral subsystems suitable for industrial control networks and edge computing, integrating interfaces such as **UART, I2C, and SPI**.

The MCU features common external interfaces and an integrated CAN bus controller; equipped with built-in EFlash, PLL, IO, and other analog IPs, supporting application scenarios such as sensor networks.

The system adopts a heterogeneous CPU architecture, consisting of three mutually independent CPU subsystems. Each CPU subsystem includes a CPU core, a Flash Memory Interface Module (FLITF), and Flash memory (NOR Flash). The three heterogeneous CPU subsystems employ CPUs with three different instruction set architectures: ARM Cortex-M3, MIPS MicroAptiv UC, and RISC-V E906, respectively. Each is paired with its own independent memory (NOR Flash). These three CPU subsystems operate independently of each other, ensuring physical isolation and achieving the highest possible level of security.

2.1 Development Board Functional Block Diagram

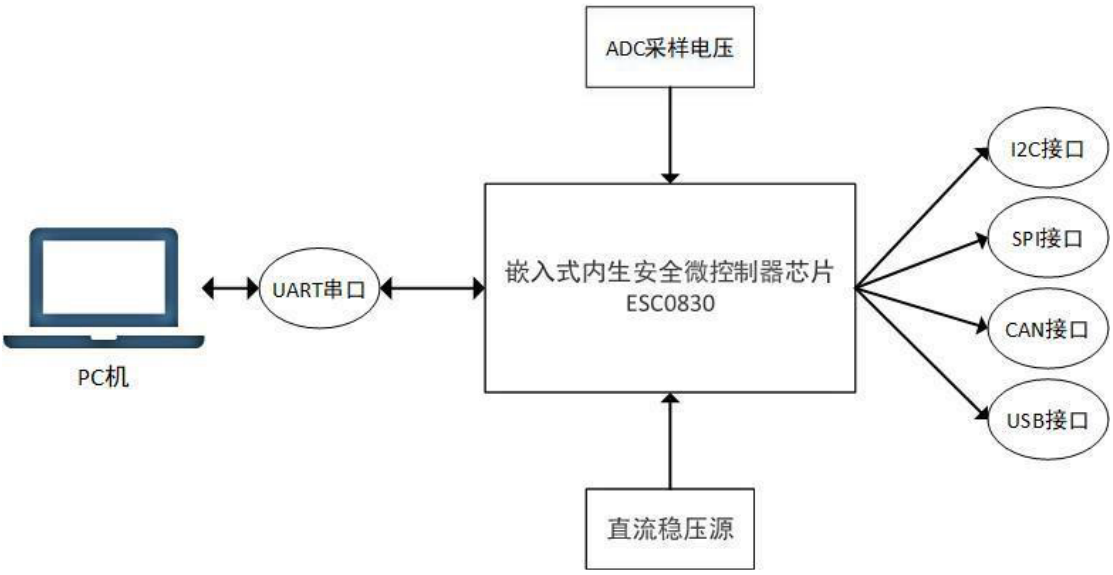

Figure 2-2 Development Board System Block Diagram

2.2 Physical picture of the development board

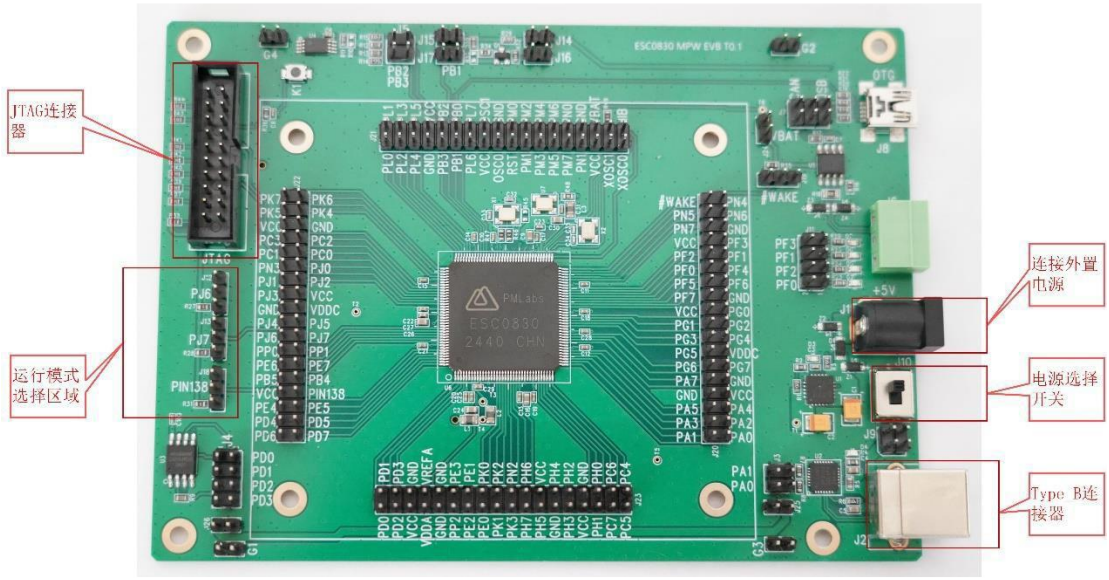

Figure 2-3 Mimic MCU Development Board

3. Technical Solution

The Mimic MCU development board is a test board developed for the purpose of realizing the Mimic MCU chip. It provides power, serial port peripherals, and corresponding control circuits for the Mimic MCU chip. After being programmed with the corresponding software, the development board can implement mimic defense functionalities and enable interactive applications with the Mimic MCU chip.

Table 3-1 Function Switching Pins for the Mimic MCU Chip

Signal	Chip Multiplexed I/O	Development Board Location
test_mode	Pin138	J18
test_sel0	Pin129	J12
test_sel1	Pin130	J13
PM4	Pin85	/

3.1 Mimic MCU Development Board Startup

The development board is compatible with two different power supply modes: external power supply and USB power supply. Setting the switch J10 to the USB position enables USB power supply. In this mode, only a USB serial cable needs to be connected. Setting the switch J10 to the power connector position enables external power supply. In this mode, an external power source must be connected, while the USB serial cable is used solely for data transmission. The development board supports various peripherals such as CAN, USB, I2C, JTAG, etc., with the relevant interfaces connected

via pin headers. Serial communication with a PC is achieved through the Type_B connector at position J2.

3.2 Mimic MCU Chip Operating Modes

The development board provides pin headers for selecting different boot modes. By configuring the jumper caps at three locations—J18, J12, and J13—to switch the voltage levels of the chip's configuration pins, three operating modes are available.

Flash Programming Mode

Set chip Pin138 (corresponding to J18 on the board) to high level (3.3V), Pin129 (corresponding to PJ6 on J12 of the board) to high level (3.3V), Pin130 (corresponding to PJ7 on J13 of the board) to high level (3.3V), and PM4 to high level (3.3V) to configure the chip for Flash programming mode. Insert the USB_UART serial cable and connect PA0 and PA1 to U0Rx and U0Tx, respectively. Power on or reset the chip, and use the chip programming tool to burn the binary program.

Normal Operation Mode

Set chip Pin138 (corresponding to J18 on the board) to low level (GND) to configure the chip for normal operation mode. Insert the USB_UART serial cable and connect PA0 and PA1 to U0Rx and U0Tx, respectively. Power on or reset the chip to run the internal program.

IP-TEST Mode

Set chip Pin138 (corresponding to J18 on the board) to high level (3.3V), Pin129 (corresponding to PJ6 on J12 of the board) to low level (GND), and Pin130 (corresponding to PJ7 on J13 of the board) to high level (3.3V) to configure the chip for test mode, enabling pin detection of the internal clock status.

3.3. Programming Method

Use the ESC08x0 chip programming tool for burning. The programming baud rate is 230400. For the first programming, click the " **Flash 配置**" button. For subsequent programming, simply select the target core, specify the binary file to be programmed, and click "一键烧写"

Figure 3-1 Programming Software Diagram

3.4. Triple-Mode Boot

The scheduler application is configured to operate in the default triple-mode boot state. This configuration enables the three heterogeneous CPUs (ARM, MIPS, and RISC-V) to output messages via UART within the application code. A binary file for the scheduler CPU is compiled, along with separate binary files for the ARM, MIPS, and RISC-V CPUs, each designed to output the message: "Hello, ESC0830. Welcome to the mimic world!".

Set the test board to programming mode and sequentially program the scheduler CPU binary file, followed by the binary files for the ARM, MIPS, and RISC-V CPUs, into the Flash memory. Then, switch the test board to run mode, reset it to start execution, set the serial port baud rate to 115200, and the serial port will output the corresponding information.

Figure 3-2 Programming Software Interface Diagram