

---

# ESC0830 芯片外设测试报告

V0.3

紫金山实验室

## 修 订 记 录

[illegible]

---

# 目 录

1. 简介 .....	4
<b>1.1. 概述 .....</b>	<b>4</b>
2. 测试工具说明 .....	5
<b>2.1. 测试板接口说明 .....</b>	<b>5</b>
<b>2.2. 芯片引脚说明 .....</b>	<b>6</b>
<b>2.3. CAN 分析仪说明 .....</b>	<b>6</b>
<b>2.4. 烧录工具说明 .....</b>	<b>8</b>
<b>2.5. 编译工具说明 .....</b>	<b>8</b>
3. 外设功能测试 .....	8
<b>3.1. UART .....</b>	<b>8</b>
<b>3.2. ADC .....</b>	<b>10</b>
<b>3.3. SPI .....</b>	<b>11</b>
<b>3.4. I2C .....</b>	<b>13</b>
<b>3.5. CAN .....</b>	<b>15</b>
<b>3.6. GPIO .....</b>	<b>15</b>
<b>3.7. TIMER .....</b>	<b>16</b>
<b>3.8. PWM .....</b>	<b>17</b>
<b>3.9. DMA .....</b>	<b>18</b>

---

## 图 目 录

图 2-1 测试板外观图 .....	6
图 2-2 CAN 分析仪外观图 .....	7
图 2-3 CANTest 通用测试软件 .....	8
图 2-4 软件烧录工具 .....	8
图 3-1 4 路 UART 实测 .....	9
图 3-2 ADC 实测 .....	11
图 3-3 PA2 时钟实测 .....	12
图 3-4 PA5 数据实测 .....	13
图 3-5 PB2 时钟实测 .....	14
图 3-6 PB3 数据实测 .....	15
图 3-7 CAN 分析仪实测 .....	15
图 3-8 GPIO 中断实测 .....	16
图 3-9 TIMER 中断实测 .....	17
图 3-10 M0 PWM3 实测 .....	18
图 3-13 DMA 实测 .....	18

---

## 表 目 录

表 3-1 UART 测试用例 .....	8
表 3-2 ADC 测试用例 .....	10
表 3-3 SPI 测试用例 .....	11
表 3-4 I2C 测试用例 .....	13
表 3-5 CAN 测试用例 .....	15
表 3-6 GPIO 测试用例 .....	16
表 3-7 TIMER 测试用例 .....	16
表 3-8 PWM 测试用例 .....	17
表 3-10 DMA 测试用例 .....	18

---

## 1. 简介

### 1.1. 概述

嵌入式内生安全控制器芯片引入异构多核 CPU 架构，紧贴嵌入式控制与物联网场景。相关理论研究表明，在异构多核 CPU 架构下能够大大降低 MCU 被攻破概率，在很多场景下很大程度提升安全等级。主要实现外部环境与控制芯片之间的安全防护功能。通过内置 3 个异构执行体（CPU）实现 MCU 的拟态化（外界看到的是 1 个 CPU，而内部是由最多 3 个异构 CPU 共同执行），使得攻击者无法有效攻击 MCU，进而保证 MCU 的安全性。

嵌入式内生安全控制器芯片具备兼容市面主流 MCU 的 IO 及功能特性，具备多个异构微处理器内核及拟态调度器，可以实现多冗余度的拟态判决。具备适用于工业控制网络及边缘计算的外设子系统：集成 UART，I2C，SPI 等 MCU 的常用外部接口，集成 CAN 总线控制器；具备内嵌 EFlash、PLL、IO 等模拟 IP，支持传感器网络等应用场景。

系统采用异构 CPU 系统架构，包含三个相互独立的 CPU 子系统，每个 CPU 子系统包含 CPU core、Flash Memory interface module(FLITF)及 Flash memory(NOR Flash)。三个异构 CPU 子系统分别采用 ARM Cortex-M3、MIPS MicroAptivUC 及 RISC-V E906 等三种不同指令集架构的 CPU，搭配各自独立的内存(NOR FLASH)，三套 CPU 子系统相互独立，保证物理隔绝，最大程度的实现安全。

---

## 2. 测试工具说明

### 2.1. 测试板接口说明

本次测试使用 4C123GH6PGE\_JEVE\_V1.1 测试板(使用 ESC0830EVBT0.1 测试板类似, 参考手册《ESC0830 拟态 MCU 芯片开发板使用手册》), 包括三种模式:

#### ➤ Flash 烧写模式

将芯片 Pin138 接高电平 (3.3V), Pin129 (对应板子上 J22 的 PJ6) 接高电平 (3.3V), Pin130 (对应板子上 J21 的 PJ7) 接高电平 (3.3V), PM4 接高电平 (3.3V), 配置芯片为 Flash 烧写模式, 插入 USB\_UART 串口线, 并将 PA0,PA1 接 U0Rx, U0Tx。上电或者复位芯片, 使用芯片烧写工具烧录 bin 程序。

#### ➤ 正常运行模式

将芯片 Pin138 接低电平 (GND), 配置为正常工作模式, 插入 USB\_UART 串口线, 并将 PA0,PA1 接 U0Rx, U0Tx。上电或者复位芯片, 运行芯片内部程序。

#### ➤ IP-TEST 模式

将芯片 Pin138 接高电平 (3.3V), Pin129 对应板子上 J22 的 PJ6) 接低电平 (GND), Pin130 (对应板子上 J21 的 PJ7) 接高电平 (3.3V), 配置芯片进入测试模式, 对内部时钟状态进行引脚检测。

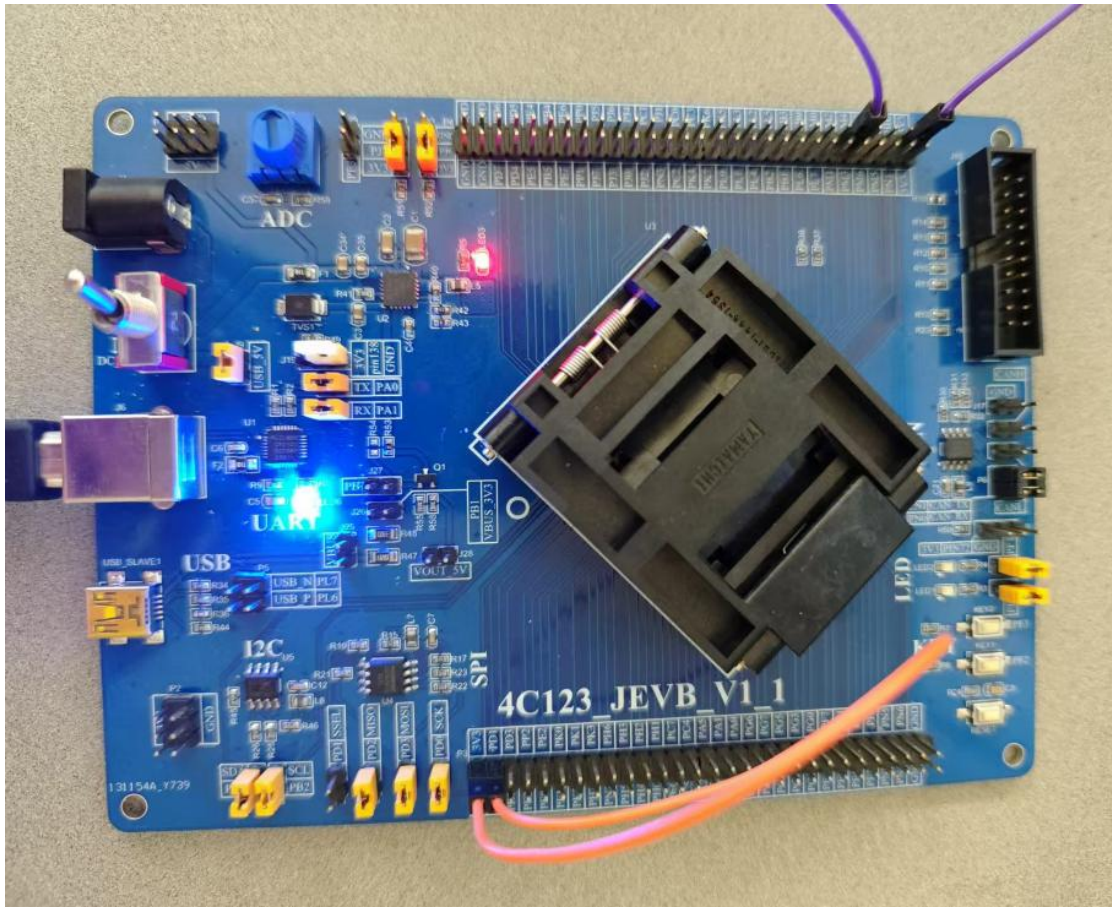


图 2-1 测试板外观图

## 2.2. 芯片引脚说明

芯片管脚总数为 144，其中 105 个管脚为功能复用管脚，具体参考文档《ESC0830 芯片用户手册》。

## 2.3. CAN 分析仪说明

本次 CAN 总线相关测试使用 USBCAN-2E-U 高性能 CAN 接口卡作为接收方。





图 2-2 CAN 分析仪外观图

本次 CAN 总线相关测试使用 CANTest 通用测试软件进行接收到的 CAN 信号的解析。

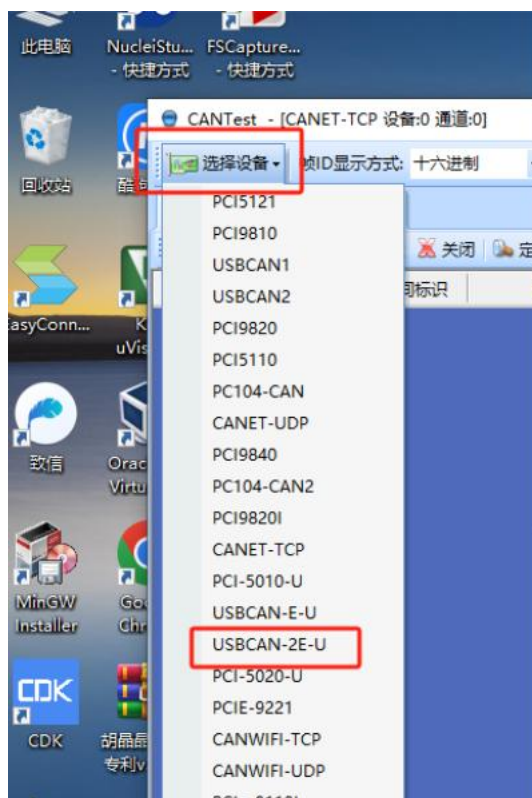


图 2-3 CANTest 通用测试软件

2.4. 烧录工具说明

本次测试使用定制的芯片烧录工具，支持指定最多 4 块 CPU 的 FLASH 并行烧录。



图 2-4 软件烧录工具

2.5. 编译工具说明

本次测试使用 VSCode 作为开发图形界面，3 个功能 CPU 的应用代码共用一份，并在 Windows 操作系统使用 GCC 为交叉编译工具链，编译生成适用于不同指令集架构的 3 份 bin。

3. 外设功能测试

3.1. UART

表 3-1 UART 测试用例

测试内容	4 路 UART 打印
------	-------------

测试步骤	1. 调度器应用程序置于默认的二模启动状态；功能 CPU 在应用程序代码中向 UART0（默认）、UART1、UART2、UART3 中发送数据；编译生成调度 CPU 和 3 个功能 CPU 的 bin 文件 2. 测试板置于烧录模式，依次烧录调度 CPU 和 ARM、MIPS 及 RISC-V 三个不同类型 CPU 的 bin 文件到 flash 3. 测试板置于运行模式，复位运行 4. 通过杜邦线分别将 J1 TX 连接 PA0、PC4、PD6、PC6；J2 RX 连接 PA1、PC5、PD7、PC7，观察串口打印
技术要求	能够得到 ARM、MIPS 及 RISC-V 三个 CPU 输出内容经过裁决之后通过 UART 打印输出，满足拟态设备架构需具备三个执行体的要求。
预期结果	1. 烧录少于 3 个功能 CPU 的 bin，无法启动； 2. J1 TX 连接 PA0、J2 RX 连接 PA1，运行模式，复位，串口打印 UART0 的“Hello, ZJU! A”。 3. J1 TX 连接 PC4、J2 RX 连接 PC5，运行模式，复位，串口打印 UART1 的“B”。 4. J1 TX 连接 PD6、J2 RX 连接 PD7，运行模式，复位，串口打印 UART2 的“C”。 5. J1 TX 连接 PC6、J2 RX 连接 PC7，运行模式，复位，串口打印 UART3 的“D”。
标准依据	ESC0830 嵌入式内生安全微控制器芯片设计手册

测试结果：与“预期结果”相符，结果 1 说明设置三冗余度调度有效；结果 2 说明输出裁决器正常；结果 3、4、5 说明 4 路 UART 正常运行。



图 3-1 4 路 UART 实测

3.2. ADC

表 3-2 ADC 测试用例

测试内容	ADC 采集电压
测试步骤	1. 调度器应用程序置于默认的二模启动状态；功能 CPU 在应用程序代码中以 3.3V 为参考电压循环检测 ADC 的模拟输入并打印转换后的数值；编译生成调度 CPU 和 3 个功能 CPU 的 bin 文件 2. 测试板置于烧录模式，依次烧录调度 CPU 和 3 个功能 CPU 的 bin 文件到 flash 3. 测试板置于运行模式，复位运行 4. 通过杜邦线将 PD1 接 J29 的无标注接口(非 PE3)，调节滑动变阻器 R59 改变电压，观察串口 ADC 模块转换数值的变化 5. 通过杜邦线将 PD1 分别接 GND 和 3.3V, 观察串口 ADC 模块转换数值的变化
技术要求	支持 ADC 采样，芯片采集到测试板上所设定的电压值
预期结果	1. 调节滑动变阻器，ADC 数据随之变化，数据在 0~4095 之间 2. PD1 接 GND, ADC 数据为 0 左右；PD1 接 3.3V，ADC 数据为 4095 左右
标准依据	ESC0830 嵌入式内生安全微控制器芯片设计手册

测试结果：与“预期结果”相符，说明 ADC 模块正常。



图 3-2 ADC 实测

3.3. SPI

表 3-3 SPI 测试用例

测试内容	SSI 主模式发送数据
------	-------------

测试步骤	1. 调度器应用程序置于默认的三模启动状态；功能 CPU 在应用程序代码中以 10KHz 时钟循环发送 SPI 数据 0x55；编译生成调度 CPU 和 3 个功能 CPU 的 bin 文件 2. 测试板置于烧录模式，依次烧录调度 CPU 和 3 个功能 CPU 的 bin 文件到 flash 3. 测试板置于运行模式，复位运行 4. 示波器测试芯片 SPI 引脚 PA2、PA5，观测 SPI 时钟串行时钟线 CLK 和串行数据线 Tx 波形
技术要求	支持 I2C 协议，芯片输出 SPI 协议波形
预期结果	1. 示波器实测 PA2 时钟频率 10KHz； 2. 示波器实测 PA5 数据波形 “01010101”；
标准依据	ESC0830 嵌入式内生安全微控制器芯片设计手册

测试结果：与“预期结果”相符，PA2 实测时钟 10.00KHz 说明 SPI 时钟工作正常，PA5 实测数据波形 0b01010101=0x55 说明 SPI 发送数据正确

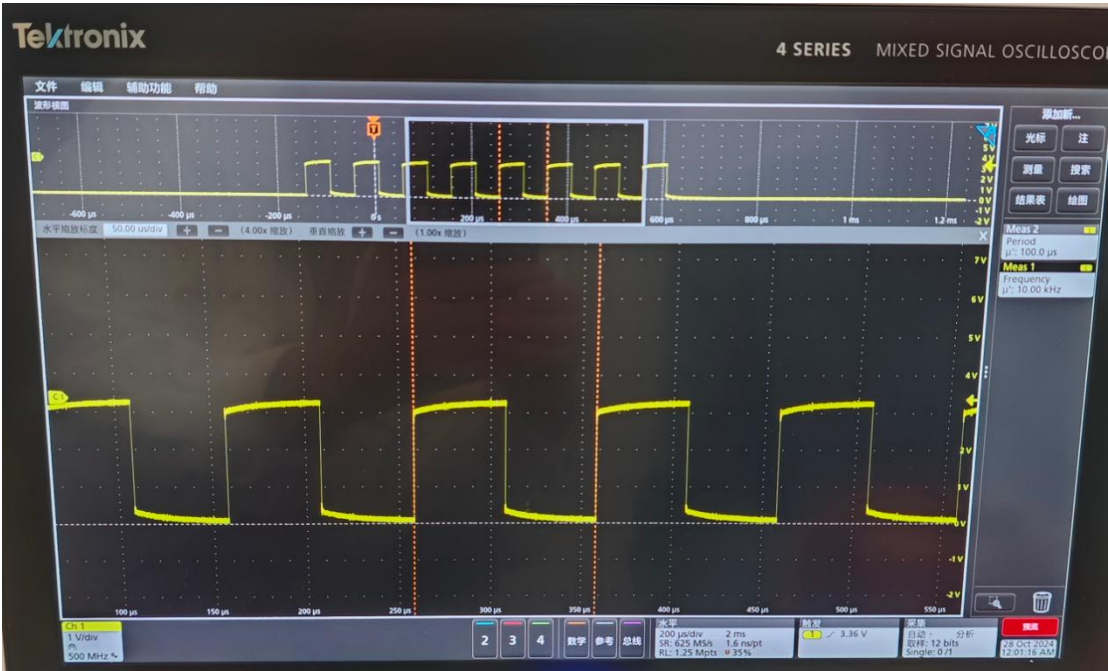


图 3-3 PA2 时钟实测



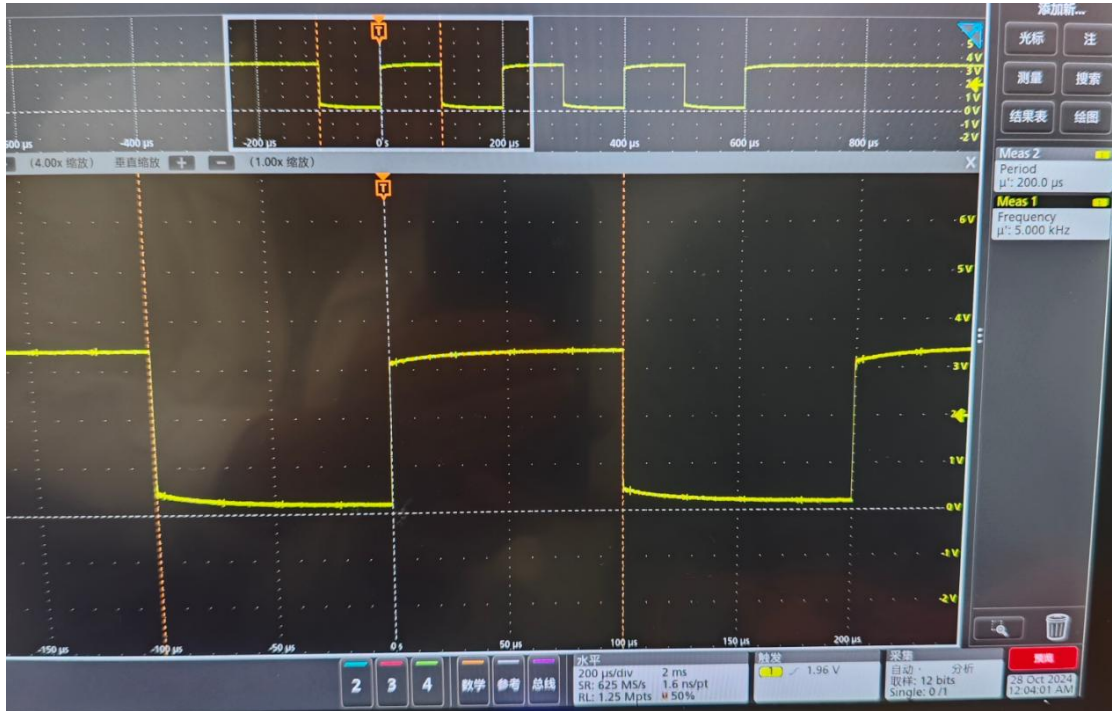


图 3-4 PA5 数据实测

3.4. I2C

表 3-4 I2C 测试用例

测试内容	I2C 作为主机发送数据
测试步骤	<div>1. 调度器应用程序置于默认的三模启动状态；功能 CPU 在应用程序代码中以 10KHz 时钟循环向目标地址 0x50 发送 I2C 数据 0x45；编译生成调度 CPU 和 3 个功能 CPU 的 bin 文件</div> <div>2. 测试板置于烧录模式，依次烧录调度 CPU 和 3 个功能 CPU 的 bin 文件到 flash</div> <div>3. 通过跳帽将测试板 J23 的 PB2 口、J24 的 PB3 口分别接 I2C 串行时钟线 SCL 和串行数据线 SDA</div> <div>4. 测试板置于运行模式，复位运行</div> <div>5. 通过示波器测试芯片输出引脚 PB2, PB3，观测 I2C 时钟和数据的波形；</div>
技术要求	支持 I2C 协议，芯片输出 I2C 协议波形
预期结果	<div>1. 示波器实测 PB2 的 SCL 时钟频率 10KHz；</div> <div>2. 示波器实测 PB3 的 SDA 数据波形“101000000010001010”，从左往右分别是 7 位地</div>

	址位 0b1010000=0x50、一位写标志 0、一位应答标志 0、8 位数据位 0b0100101=0x45、一位应答标志 0；
标准依据	ESC0830 嵌入式内生安全微控制器芯片设计手册

测试结果：与“预期结果”相符，PB2 实测时钟 9.956KHz 说明 I2C 时钟工作正常，PB3 实测数据波形“101000000010001010”说明 I2C 发送数据正确。



图 3-5 PB2 时钟实测

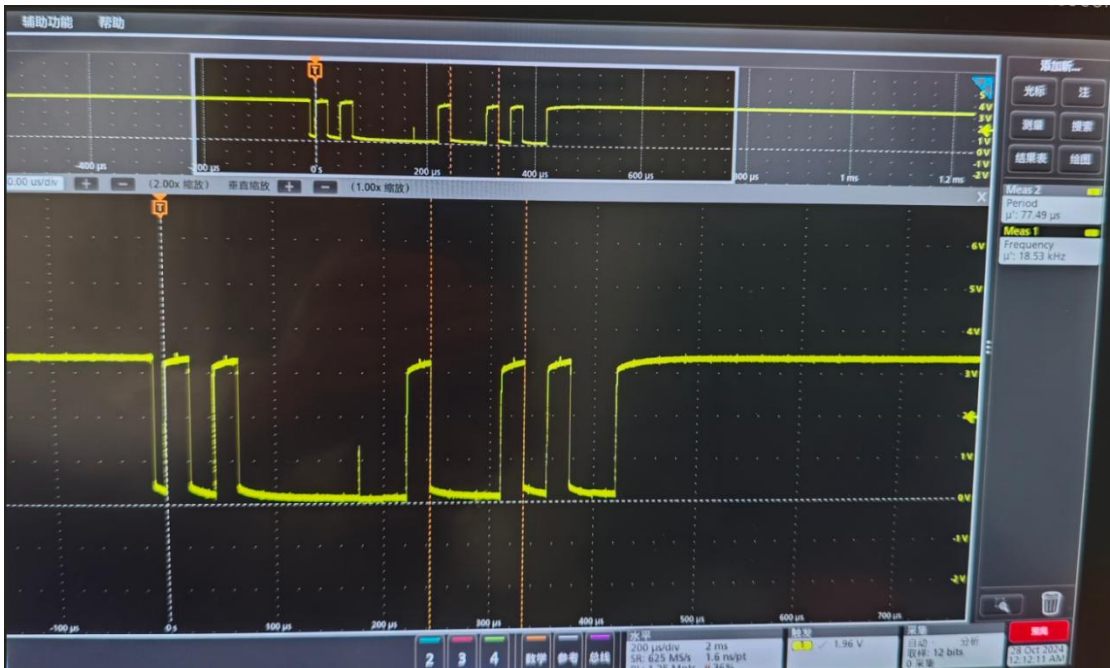




图 3-6 PB3 数据实测

3.5. CAN

表 3-5 CAN 测试用例

测试内容	CAN 发送数据
测试步骤	1. 调度器应用程序置于默认的二模启动状态；功能 CPU 在应用程序代码中以 1MHz 波特率循环发送帧号 0x26 数据 0x12345678 和帧号 0x27 数据 0x87654321；编译生成调度 CPU 和 3 个功能 CPU 的 bin 文件 2. 测试板置于烧录模式，依次烧录调度 CPU 和 3 个功能 CPU 的 bin 文件到 flash 3. 通过跳帽将 P6 的 PN1 接 CAN_TX, PN0 接 CAN_RX, 通过杜邦线将 J15, J16 的 CANH, CANL 接 CAN 报文分析仪 CAN1 的 CANH, CANL 接口 4. 测试板置于运行模式，复位运行 5. 通过逻辑分析仪查看 CAN 发送数据是否正确
技术要求	支持 CAN 协议，芯片输出报文数据
预期结果	1. CAN 分析仪以 1MHz 波特率接收数据，循环收到帧号 0x26 数据 0x12345678 和帧号 0x27 数据 0x87654321；
标准依据	ESC0830 嵌入式内生安全微控制器芯片设计手册

测试结果：与“预期结果”相符，说明 CAN 发送数据正常。



序号	传输方向	时间标识	帧ID	帧格式	帧类型	数据长度	数据(HEX)
00000000	接收	11:06:24.7...	0x00000026	数据帧	标准帧	0x04	78 56 34 12
00000001	接收	11:06:24.7...	0x00000027	数据帧	标准帧	0x04	21 43 65 87
00000002	接收	11:06:24.7...	0x00000026	数据帧	标准帧	0x04	78 56 34 12
00000003	接收	11:06:24.7...	0x00000027	数据帧	标准帧	0x04	21 43 65 87
00000004	接收	11:06:24.7...	0x00000026	数据帧	标准帧	0x04	78 56 34 12
00000005	接收	11:06:24.7...	0x00000027	数据帧	标准帧	0x04	21 43 65 87

图 3-7 CAN 分析仪实测

3.6. GPIO

表 3-6 GPIO 测试用例	
测试内容	GPIOF3 中断功能
测试步骤	1. 调度器应用程序置于默认的二模启动状态；功能 CPU 在应用程序代码中以配置 GPIOF 的 Pin3 口上升沿触发中断；编译生成调度 CPU 和 3 个功能 CPU 的 bin 文件 2. 测试板置于烧录模式，依次烧录调度 CPU 和 3 个功能 CPU 的 bin 文件到 flash 3. 测试板置于运行模式，复位运行 4. 按压测试板上的按键 Key2
技术要求	支持 GPIO 中断
预期结果	1. 芯片输出 GPIOF3 中断处理函数中的告警信息
标准依据	ESC0830 嵌入式内生安全微控制器芯片设计手册

测试结果：与“预期结果”相符，说明 GPIO 中断正常运行。

```
test_gpio-----
gpiof3 start:...
GPIO_INT_PIN_3, cnt = 1
GPIO_INT_PIN_3, cnt = 2
GPIO_INT_PIN_3, cnt = 3
GPIO_INT_PIN_3, cnt = 4
GPIO_INT_PIN_3, cnt = 5
GPIO_INT_PIN_3, cnt = 6
GPIO_INT_PIN_3, cnt = 7
GPIO_INT_PIN_3, cnt = 8
GPIO_INT_PIN_3, cnt = 9
```

图 3-8 GPIO 中断实测

3.7. TIMER

表 3-7 TIMER 测试用例	
测试内容	Timer0A 中断功能
测试步骤	1. 调度器应用程序置于默认的二模启动状态；功能 CPU 在应用程序代码中已配置 Timer0A 倒计时 0x01000000 触发中断；编译生成调度 CPU 和 3 个功能 CPU 的 bin 文件 2. 测试板置于烧录模式，依次烧录调度 CPU 和 3 个功能 CPU 的 bin 文件到 flash 3. 测试板置于运行模式，复位运行

技术要求	支持定时器中断
预期结果	1. 芯片输出 timer0A 中断处理函数中的告警信息
标准依据	ESC0830 嵌入式内生安全微控制器芯片设计手册

测试结果：与“预期结果”相符，说明 TIMER 正常运行。



图 3-9 TIMER 中断实测

### 3.8. PWM

表 3-8 PWM 测试用例

测试内容	M0 PWM3 功能
测试步骤	<ol style="list-style-type: none"><li>1. 调度器应用程序置于默认的三模启动状态；功能 CPU 在应用程序代码中已配置 MOPWM3 以占空比 75%在 GPIOB 的 PIN5 口生成调制波形；编译生成调度 CPU 和 3 个功能 CPU 的 bin 文件</li><li>2. 测试板置于烧录模式，依次烧录调度 CPU 和 3 个功能 CPU 的 bin 文件到 flash</li><li>3. 测试板置于运行模式，复位运行</li></ol>
技术要求	支持定时器中断
预期结果	1. GPIOB 的 PIN5 口产生占空比 75%的 PWM 波形
标准依据	ESC0830 嵌入式内生安全微控制器芯片设计手册

测试结果：与“预期结果”相符，说明 M0 PWM3 正常运行。

图 3-10 M0 PWM3 实测

3.9. DMA

表 3-9 DMA 测试用例

测试内容	DMA 内存到外设功能
测试步骤	1. 调度器应用程序置于默认的三模启动状态;功能 CPU 在应用程序代码中已配置 UART0 启用 DMA，DMA 开启传输后奖 5 个 ‘U’ 传递到 UART0 tx 口；编译生成调度 CPU 和 3 个功能 CPU 的 bin 文件 2. 测试板置于烧录模式，依次烧录调度 CPU 和 3 个功能 CPU 的 bin 文件到 flash 3. 测试板置于运行模式，复位运行
技术要求	支持 DMA
预期结果	1. 芯片通过 UART0 输出 DMA 传递的 5 个 ‘U’
标准依据	ESC0830 嵌入式内生安全微控制器芯片设计手册

测试结果：与“预期结果”相符，说明 DMA 正常运行。



图 3-11 DMA 实测