DOI:10.19651/j.cnki.emt.2416346

基于 SPI 全双工通信的 ICNI 健康管控系统

刘 维
(西南电子技术研究所 成都 610036)

摘 要：无人机通信导航识别系统(ICNI)更高程度的综合化使得 ICNI 主控模块需完成基于多种总线通信的系统管控、数据转发等功能,对核心处理器的处理性能和接口资源有着更高要求。因此对模块中的健康管控系统提出了更轻量级的设计要求。本文在主控模块内采用 CPU+MCU 分布式架构设计了一种全国产化的基于 CAN 总线的健康管理系统。由 FT-2000/4 作为核心处理器,MCU 作为协处理器实现 CAN 总线接口单元,二者通过 SPI 全双工通信完成 CAN 总线应用层数据交互,实现 CPU 对系统内各模块/单元的状态查询、命令下发,从而实现系统的健康管理。该系统可实时监测系统内各模健康状态,电路设计仅占用核心处理器 SPI 和 GPIO 接口,不占用其他接口资源和可编程逻辑(FPGA)资源,硬件设计简单,减轻电路布局和结构设计压力。健康管理部分电路器件成本较业内其他典型设计降低至少 60%,符合低成本需求。

关键词：健康管理单元;FT-2000/4 处理器;SPI 通信;CAN 总线;ICNI 主控模块
中图分类号：TN971 **文献标识码**：A **国家标准学科分类代码**：590.30

ICNI health control system based on SPI full duplex communication

Liu Wei
(Southwest China Institute of Electronic Technology, Chengdu 610036, China)

Abstract: The integration of the integrated communications/navigation/identification (ICNI) system for unmanned aerial vehicles necessitates that the master control module of the ICNI performs functions such as system control and data forwarding via multiple bus communications. This requirement imposes higher demands on the processing performance and interface resources of the core processor. Consequently, there is a need for a more lightweight design for the health control system within the module. This paper presents a fully domestic CAN bus-based health management system designed using a CPU+MCU distributed architecture within the main control module. The FT-2000/4 serves as the core processor, while the MCU acts as the coprocessor to implement the CAN bus interface unit. The two components facilitate application layer data exchange over the CAN bus through SPI full-duplex communication, enabling the CPU to query the status and issue commands to each module/unit in the system, thereby achieving effective health management. The system is capable of real-time monitoring of the health status of each module. Notably, the circuit design utilizes only the SPI and GPIO interfaces of the core processor, without occupying additional interface resources or programmable logic (FPGA) resources. This approach simplifies the hardware design and alleviates the challenges associated with circuit layout and structural design. Furthermore, the cost of the health management circuitry is at least 60% lower than that of other typical designs in the industry, addressing the need for cost-effective solutions.

Keywords: health management unit;FT-2000/4 processor;SPI communication;CAN bus;ICNI main control module

0 引 言

现代战争电子对抗技术不断发展,随着保密、抗干扰、信号隐蔽等众多功能在军用无人机上的部署,综合导航识别系统(ICNI)也不断向着综合化和一体化的方向发展[1]。对系统的故障检测、诊断能力要求也进一步提高[2-3]。主控模块作为 ICNI 系统的心脏,承担着系统内各模块状态监测及健康管理的任务。当前无人机 ICNI 系统健康管理有如下需求:1)无人机综合模块化机载系统内模块采用 ASAAC 标准设计,电路集成度和复杂度大幅提升,使得模

收稿日期:2024-07-01

块的电路布局、结构散热设计困难,对主控模块的健康管理功能电路的设计实现有更轻量级的需求[3];2)随着系统复杂度和集成度的提高,主控模块需完成基于多种总线通信的数据处理及高速转发,对核心处理器的处理资源和接口资源有更高的要求[4-5],因而对模块的健康管理功能部署有更严苛的资源占用限制要求;3)在军用机载设备元器件自主可控的要求下,国产军品级器件的高单价使得单模块成本控制的压力增大,模块电路设计有更高的降成本需求[6]。

CAN (controller area network) 总线拥有网络内信息共享,通信速率高,抗干扰能力强等优点,多用于汽车电子、航空航天等电子控制系统的健康管理、状态监控,也是 ICNI 系统首选的健康管理数据总线[7-10]。目前行业内,CAN 总线接口控制通常使用 28XX 系列数字信号处理器(digital signal processing,DSP)、STM32 系列微控制器(microcontroller unit,MCU)以及片上系统(system on chip,SOC)实现[11-13]。DSP、MCU 的对外接口资源以及性能有限;SOC 性能强于前两者,但其 ARM 处理器端能直接使用的内存空间最大只有 1 GB,是其处理性能的瓶颈。上述芯片均不适合直接作为主控模块的核心处理芯片。

综上考虑,本文给出一种采用高性能 CPU+MCU 架构的全国产化健康管理系统设计。CPU 采用 FT-2000/4 四核处理器,作为核心处理器完成系统的健康管理功能;MCU 作为协处理单元实现 CAN 总线的接口管理功能,辅助 CPU 实现基于 CAN 总线的系统健康管理。二者间通过 SPI 总线全双工通信实现 CPU 对 CAN 总线协议层的数据收发,从而实现基于 CAN 总线的系统健康管理、维护升级等功能。该设计具有如下优点:1)电路设计简单,CPU 与 MCU 间仅通过 SPI 总线和 GPIO 总线相连,均为低频信号,大幅降低模块电路布局布线复杂度;2)CPU 与 MCU 间仅通过一组四线 SPI 接口和一根 GPIO 相连即可实现功能,保证了高性能 CPU 其他可用接口资源,且不占用模块的可编程逻辑资源,最大化保证了主控模块除去健康管理功能外主要业务的处理性能;3)MCU 较 DSP 和 SOPC 相比,其最小系统所需元器件的种类及数目更少,成本更低,缓解了电路布局压力的同时更符合低成本设计原则。

1　系统架构设计

ICNI 系统内,基于 CAN 总线的健康管理系统包括主控节点、从节点以及上位机。系统网络架构如图 1 所示。

系统内模块作为 CAN 总线主控节点,其他模块作为 CAN 总线从节点。主控模块会通过 CAN 总线周期下发机内自检(built-in test,BIT)指令,从节点模块收集各自模块的 BIT 要素信息,通过 CAN 总线回传给主控模块。主控模块根据收集到的 BIT 信息完成决策,并将状态上报给上位机。上位机通过以太网与主控模块连接,用户可通过上位机实时监测系统内各模块健康状态,实现状态查询,功能构建/解构,在线升级等功能。

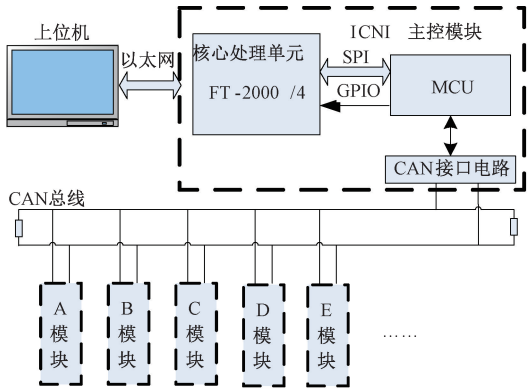

图 1　ICNI 系统内 CAN 总线网络架构

Fig. 1　CAN bus network architecture within ICNI system

在主控模块中,FT-2000/4 处理器作为核心处理器完成系统健康管理控制,MCU 作为协处理器实现 CAN 接口管理单元,将处理器发来的指令消息通过 CAN 总线转发至网络,并将收到的从节点消息上报给处理器。处理器和 MCU 之间通过 SPI 总线实现全双工通信。

2　主控模块硬件设计

2.1　CPU 核心处理单元设计

主控模块的核心 CPU 采用 FT-2000/4 处理器。FT-2000/4 是我国自主研制的一款高性能四核处理器。集成了 4 个 64 位高性能核,主频 2.6 GHz。其包含丰富的外围功能部件和外部接口,可应用于高性能嵌入式应用场景[14]。本项目 FT-2000/4 处理器采用 4 片 512 MB DDR3 组成 2 GB 内存,4 片 QSPI FLASH 组成 128 MB FLASH 用于存储系统和应用程序。主控模块电源外部输入 12 V,经过 EMI 滤波电路滤波后,经过 DC/DC 电源芯片转换分别生成处理器所需各类型电源。FT-2000/4 的系统输入时钟由 48 MHz 晶振提供。CPU 核心处理单元功能框图如图 2 所示。

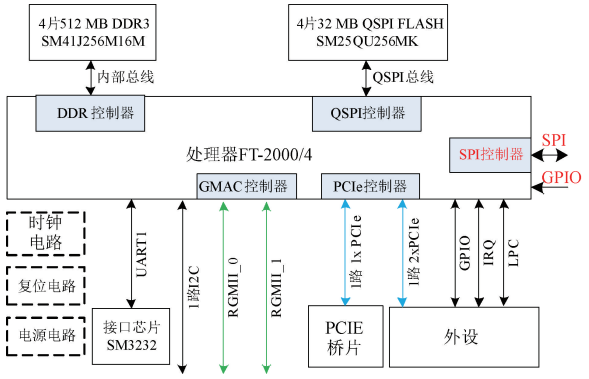

图 2　CPU 核心处理单元功能框图

Fig. 2　Functional block diagram of CPU processing unit

2.2　SPI 通信设计

1) SPI 总线硬件设计

串行外设接口(serial peripheral interface,SPI)是由

Motorola 公司推出的一种高速、全双工、同步串行总线，被广泛应用于板内电路设计中[15]。SPI 通信以主/从模式工作，总线中包含 1 个主机设备，可包含多个从机设备。总线使用的信号线分别为 SCK、MOSI、MISO 和 CS。SCK 为串行移位时钟线，用来同步数据传输，由主机提供；MOSI 为主机输出/从机输入信号线，MISO 为主机输入/从机输出信号线；CS 为片选线(从设备选择)。SPI 通信过程中数据的发送和接收同步进行，通信由主设备产生串行移位时钟来发起。SPI 总线连接如图 3 所示。

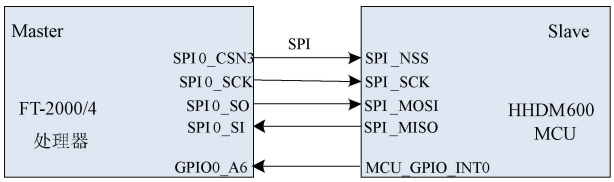

图 3　SPI 硬件连接框图

Fig. 3　SPI hardware connection diagram

本设计中 FT-2000/4 作为 SPI 通信主机端，MCU 作为从机端。SPI 总线通过时钟极性和时钟相位两个控制位来配置数据传输方式。本设计中，通信双方均遵循时钟极性高电平有效(空闲态为低电平)，串行时钟相位为 SCK 从空闲状态跳变的第一个边沿(上升沿)采样。数据按照 8 bits 位宽进行传输。总线数据传输为履带式结构。在片选信号有效的时间内，伴随着时钟信号，SI 数据输入线和 SO 数据输出线上同时有数据比特流流入和流出，如图 4 所示。

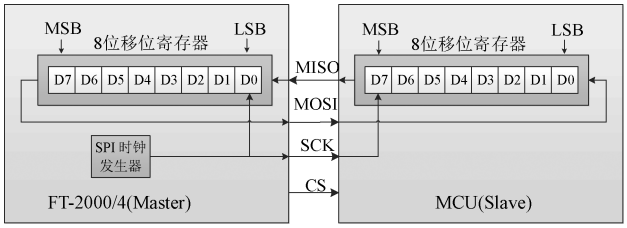

图 4　SPI 总线传输示意图

Fig. 4　Schematic diagram of SPI bus transmission

2) CPU 外部中断设计

MCU 作为从机无法主动产生 SCK 时钟，但需要将收到的 CAN 总线数据主动上报给主机。因此本设计中采用 MCU 的一根通用输出信号 GPIO 连接到 CPU 的 GPIO 接口。当有从机数据上报请求时，MCU 的 GPIO 会产生周期的下降沿。CPU 一旦检测到 GPIO 信号线的下降沿，便立即触发外部中断，发起 SPI 读动作将 MCU 端的数据取走。

2.3　CAN 总线设计

MCU 作为主控模块 CAN 接口单元接入 CAN 总线网络。本设计采用国产 MCU HHDM600 作为 CAN 接口管理控制芯片。MCU 采用高性能 32 位 ARM Cortex-M4 内核，最高主频 144 MHz，内部集成 144 KB 内存 SRAM，可外扩 FLASH 和 SRAM。具有 2 个 CAN 总线外部接口。模块 12 V 输入电压经过 EMI 滤波电路后，由 DC/DC 电源芯片(SM4630)转成 3.3 V，为 HHDM600 供电。芯片的主时钟由 25 MHz 外部晶振提供。HHDM600 支持两路 CAN 总线接口，接口兼容规范 2.0 A 和 2.0 B，位速率达 1 Mbit/s。采用 SM65HVD230D 作为 CAN 总线收发器。该收发器满足 ISO 11898 CAN 总线标准，提供差分发送和差分接收能力，传输速率达到 1 Mbps。CAN 总线接口电路如图 5 所示。

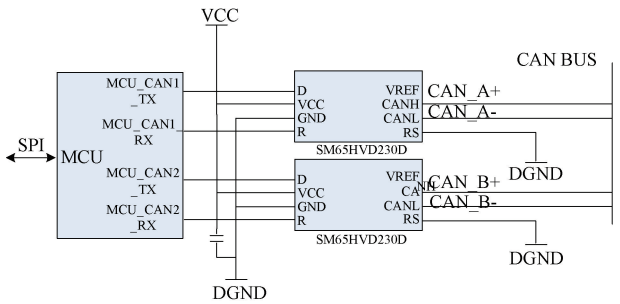

图 5　CAN 总线接口电路设计

Fig. 5　Circuit design of CAN bus interface

3　软件设计

3.1　CPU SPI 驱动设计

1) SPI 接口配置

FT-2000/4 处理器的 IO 管脚采用了复用设计，用户需要通过配置复用配置寄存器来设置引脚的当前功能。处理器使用四根 SPI 信号线为 SPI0_SCK、SPI0_CSN3、SPI0_SO、SPI0_SI 四根管脚线，分别与 LPC_LAD3、GPIO1_A6、GPIO1_A7、GPIO1_B0 功能复用。对处理器的引脚复用寄存器进行配置，寄存器的基地址为：0x28180000。每个复用引脚的控制信号有 4 位。其中 bit[1:0]表示功能选择，00:func0，01:func1，10:func2，11:func3。管脚复用配置如表 1 所示。

表 1　SPI 管脚复用配置

Table 1　Multiplex configuration of SPI pins

SPI 功能管脚控制域	偏移[有效位]	功能选择
spi0_sck_pad	0x0208[15:12]	00(func0)
spi0_cs3_pad	0x0218[3:0]	01(func1)
spi0_s0_pad	0x0208[11:8]	00(func0)
spi0_si_pad	0x0208[7:4]	00(func0)

由引脚复用寄存器的基地址加上偏移地址可得到各 SPI 功能管脚复用配置寄存器的地址，向各寄存器的有效位中写入相应的功能选择值，即可分别使能 4 根 SPI 管脚功能。

SPI 时钟线的时钟是由 FT-2000/4 的 SPI 时钟和 SPI 分频寄存器 BAUDR 的 SCKDV 域的值决定。SPI 时钟的

计算公式如下:

$$spi\_sck = \frac{ssi\_sck}{SCKDV} \tag{1}$$

系统时钟 ssi_clk 为 48 MHz,系统使用的 spi_sck 时钟为 8 MHz,因此在驱动中配置分频系数 SCKDV 为 6。根据应用场景配置 SPI0 的控制寄存器 CTRLR0 为收发工作模式,配置传输时的时钟极性和时钟相位,以及相应片选有效。

2) SPI 中断配置

SPI 通信作为一种串行通信模式,在数据传输过程中需要保证时钟和数据的实时同步以完成数据的无差错传输,对确定性具有较高的要求[16]。为减少任务调度对总线时序的影响,本设计中 SPI 采用中断模式进行数据通信。FT-2000/4 处理器采用 ARM V8 架构,提供单独的 SPI 中断。本设计中使用 SPI0 控制器,其中断向量 ID 为 50。当处理器收到 SPI 控制器置位的中断信号时,处理器会停止当前工作优先处理中断信号代表的工作,完成后再继续之前的工作。驱动程序中将中断 ID 与相应的中断服务程序挂接,并将该中断单独绑定至处理器的 3 号核。3 核只用于处理 SPI 中断以及 SPI 数据通信的相关业务,与其他核没有耦合,避免其他核的任务及中断与其相互影响[17]。

3)底层驱动接口设计

SPI 总线上的数据收发操作在同时进行,需要在一次中断服务过程中同时完成当前 SPI 数据接收和发送。CPU 接收 MCU 发来的数据也是通过发送无效数据来驱动实现。设计 SPI 数据收发函数,接口格式为:

FError FSpimTransferByInterrupt(FSpim * instance_p, const void * tx_buf, void * rx_buf, fsize_t len)

其中,instance_p 为 SPI 控制器配置句柄,tx_buf 为发送数据指针,rx_buf 为接收数据指针,len 为此次通信数据字节数。

CPU 发送数据时,将待发数据缓存地址填入收发函数的 tx_buf 参数中。CPU 通过使能并置位中断位来触发中断服务,将相关发送参数传入中断服务函数 FSpimInterruptHandler 中。中断服务函数中,根据当前发送 FIFO 和接收 FIFO 中的状态循环按字节读取和发送 SPI 总线数据,循环触发中断直到已处理数据字节数达到此次通信需求待发送字节数 len,则认为此次有效数据已全部处理发送,屏蔽相应中断位,结束中断服务。CPU 发起数据接收时,通过发送 len 个无效符字节 0x0 到总线,总线上回读到的 len 个字节即为接收到对端的有效数据,保存至 rx_buf 中。收发驱动工作流程如图 6 所示。

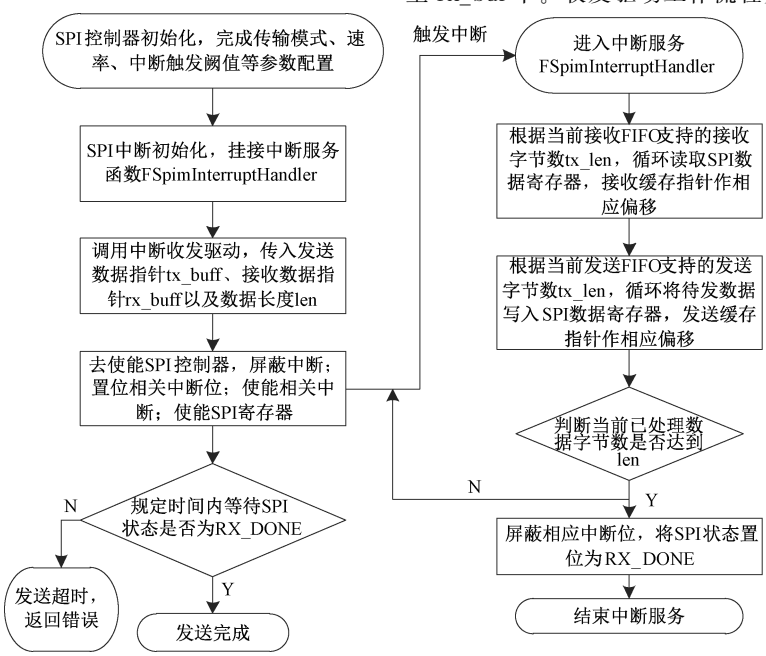

图 6　SPI 底层收发驱动工作流程

Fig. 6　Workflow of the SPI underlying transceiver driver

3.2　全双工通信设计

SPI 通信机制中 SPI 时钟由主机提供,只有主机发起主动查询,从机才能将数据发给主机,从机不具备主动发送消息的能力。在本设计中由于 MCU 作为模块的 CAN 总线接口单元,需要与 CPU 进行实时通信。因此硬件设计上使用一根外部中断线,利用 CPU 的 GPIO 中断处理机制实现双处理器的 SPI 全双工通信。

1) GPIO 中断配置

CPU 采用 GPIO 中断信号用来感知 MCU 从机的数据上报请求。当 MCU 有数据发送请求时,会将数据放在 DMA buffer,并发送若干个连续方波信号至 FT-2000/4 的 GPIO 管脚,通知 CPU 读走数据。CPU 在 GPIO 中断初始

化函数中，设置中断类型为边沿触发，极性为下降沿触发，并挂接中断处理函数。创建 SPI 业务信号量 semID_SpiInt，任务获取到该信号量则代表着有 SPI 数据业务请求。当处理器检测到中断管脚有下降沿，进入中断处理函数，清管脚中断，释放业务信号量 semID_SpiInt 后退出。

2）通信业务设计

为实现全双工通信，CPU 主机端需实时关注接收信号线 SI 上的数据，因为在 CPU 作为主机给从机发送数据的过程中 MCU 可能在此时发送给主机有效数据。SPI 通信业务流程如图 7 所示。

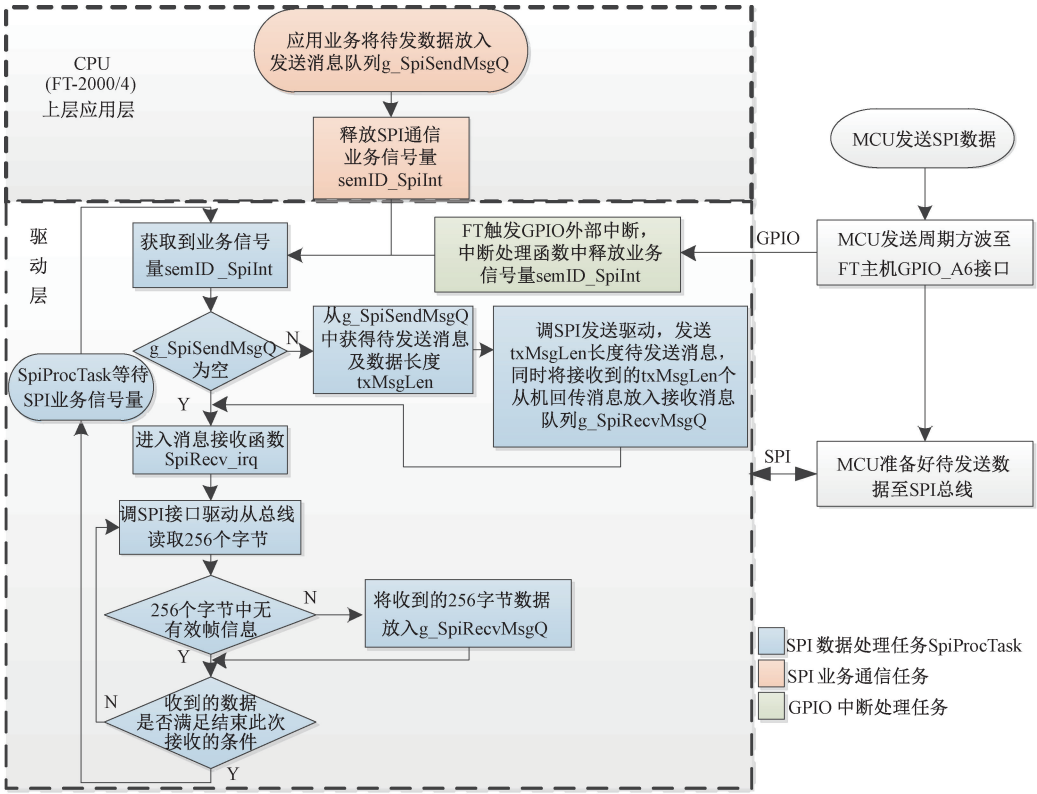

图 7　SPI 通信业务流程

Fig. 7　Process of SPI communication business

图中，SPI 业务通信任务位于上层应用层，SPI 数据处理任务 SpiProcTask 位于驱动层。当上层应用层 SPI 通信任务有消息发送请求时，会将待发消息放入发送消息队列 g_SpiSendMsgQ，并释放信号量 semID_SpiInt。SPI 数据处理任务 SpiProcTask 会持续监测业务信号量 semID_SpiInt。一旦获取到该信号量，便会进入业务处理流程，调用 SPI 驱动发送业务数据。如上节所述，CPU 触发 GPIO 外部中断后，也会释放信号量 semID_SpiInt，告知 CPU 有 MCU 数据到来，进入 SpiProcTask 任务流程中。在 SpiProcTask 中，首先判断当前发送消息队列是否为空，非空则进入发送函数中发送待发数据，然后进入接收数据函数 SpiRecv_irq。在 CPU 发送过程中，CPU 需要同时关注 SI 线上收到的 MCU 回传数据，在该过程中如果 MCU 有发送数据请求，MCU 端的待发送数据会在时钟驱动下直接被传输到 SPI 总线上。因此应对 CPU 在发数过程中收到 MCU 发送有效数据的场景，设计如下：发送函数中每发送 tx_len 个数据，将 SI 线上收到的 tx_len 个 MCU 回传数据全部放入接收待解析消息队列 g_SpiRecvMsgQ 中，这样来自 MCU 的所有数据都会被保存至消息队列之中，供上层应用解析。接收函数中，会每次按 256 字节数据读取 MCU 端数据，并对其中内容进行检查判断，并将有效数据放入消息队列 g_SpiRecvMsgQ。该过程的工作示意图如图 8。

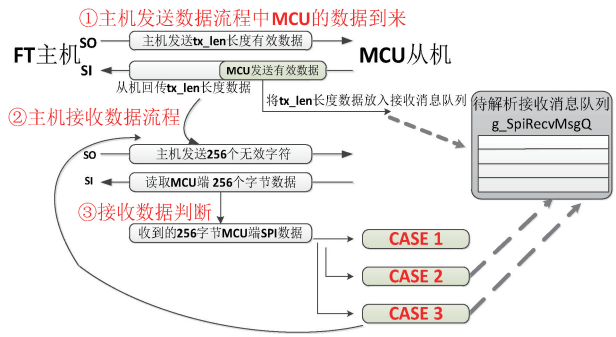

图 8　主从机同时收发场景处理

Fig. 8　Processing scenarios when the primary and secondary computers simultaneously send and receive data

其中,CASE1 为:若 256 字节中不包含帧起始/结束符,则认为无有效数据,结束此次流程;CASE2 为:若 256 字节中包含帧起始/结束符,若有结束符,且后面全部为 MCU 发的无效空白符,则认为该次流程数据传输完毕,将此 256 字节数据放入接收消息队列,结束此次流程;CASE3 为:256 字节中有起始符无结束符,认为此次收数为有效数据,且数据传输尚未结束。将此 256 字节数据放入接收消息队列,继续读取 MCU 端数据以进行下次判断。

CPU 与 MCU SPI 全双工通信时总线时序如图 9 所示。

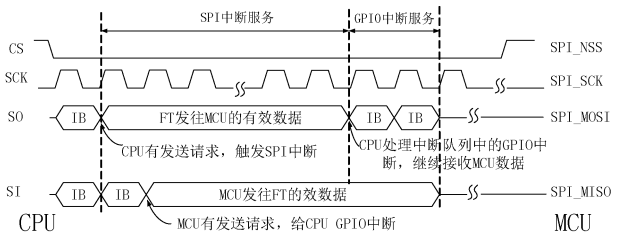

图 9　SPI 全双工通信时序

Fig. 9　Timing of SPI full-duplex communication

图 9 中,IB 为 CPU 发送的无效比特,用以驱动 SPI 总线接收。CPU 在发送有效数据的过程中,通过连续响应 SPI 中断服务和 GPIO 中断服务,将 SI 线上的来自 MCU 的有效数据全部接收。采用该设计,收发双发无需通过额外的协议握手指令,实现了双方同时发送接收数据,达到了高效全双工通信的效果[18]。

3.3　通信协议设计

1)协议帧设计

SPI 总线链路层数据通信采用一种改进的串行线路网际协议(serial line internet protocol,SLIP)。因其包装头小,被广泛用作封装 IP 包的方式。该协议可通过 END(0xC0)特殊字符作为数据报的开始和结束,将有效数据报与数据到来前的线路噪声作隔离。本设计中为保证 CPU 与 MCU 间通信可靠性,在数据报文尾部增加两字节 CRC 校验位,用来保存当前报文的 CRC 校验和。根据 CRC 校验和,收发端能够在链路层及时侦测到传输过程中的错误信息,作出异常处理,提高传输效率。帧格式如图 10 所示。

图 10　SLIP 通信协议帧格式

Fig. 10　Frame format of the SLIP communication protocol

其中,data 数据段为 SPI 总线的帧数据,其结构如图 11 所示。

图 11　数据段帧结构

Fig. 11　Frame structure of data segment

其中,FlagTag 为自定义帧头,SrcId 为消息源 ID,DestId 为消息目的 ID,MsgId 为消息 ID,Type 为帧类型,Len 为消息长度,buf 为有效数据内容。

2)CPU 解帧设计

接收消息队列 g_SpiRecvMsgQ 中保存了 SPI 总线上来自 MCU 的全部数据,其中包括无效字符以及带 SLIP 协议的有效数据。MCU 端通过 DMA 发送机制保证了有效数据在总线上的连续性。会存在一条完整的 SLIP 报文信息分布在消息队列中的多条消息中的情况,主机端需要从消息队列中提取出完整的帧信息。在应用层设计 SPI 总线接口数据处理任务,工作流程如图 12 所示。

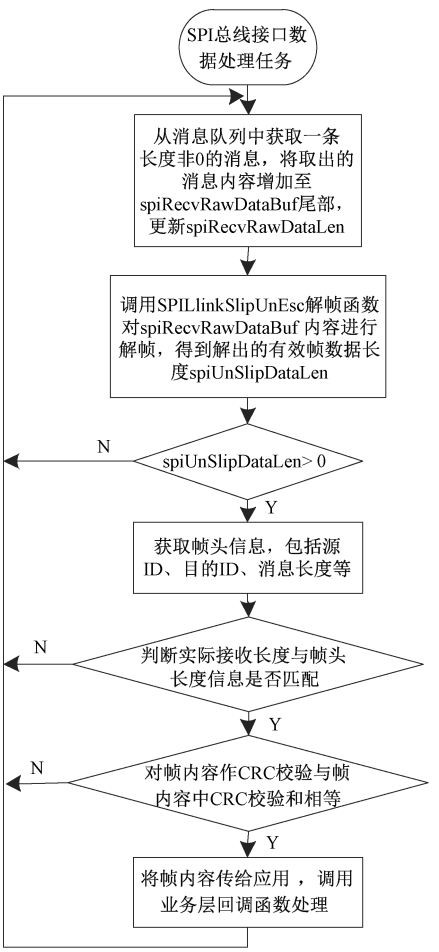

图 12　SPI 总线数据处理任务流程

Fig. 12　Task flow of SPI bus data processing

图 12 中,spiRecvRawDataBuf 为应用层待解析数据缓存,spiRecvRawDataLen 为其中的待解析数据长度。SPI 总线接口数据处理任务会持续从接收消息队列 g_SpiRecvMsgQ 中取出消息,增加至待解析数据 spiRecvRawDataBuf 的尾部,然后将待解析数据传递给解帧函数 SPILlinkSlipUnEsc。解帧函数若从中解出完整的

有效数据帧，则将解出的完整数据帧返回给 SPI 总线接口数据处理任务，由其作 CRC 校验。若校验通过则将数据帧传递给上层业务应用，校验不通过则丢弃该帧数据。

SLIP 协议的解帧函数 SPILlinkSlipUnEsc 的函数格式为：INT32 SPILlinkSlipUnEsc（UINT8 * pDest，UINT8 * pSrc，INT32 * pLength）。其中 pDest 为解析后的数据区首地址，pSrc 为待解析数据首地址，pLength 为待解析数据长度。工作流程如图 13 所示。

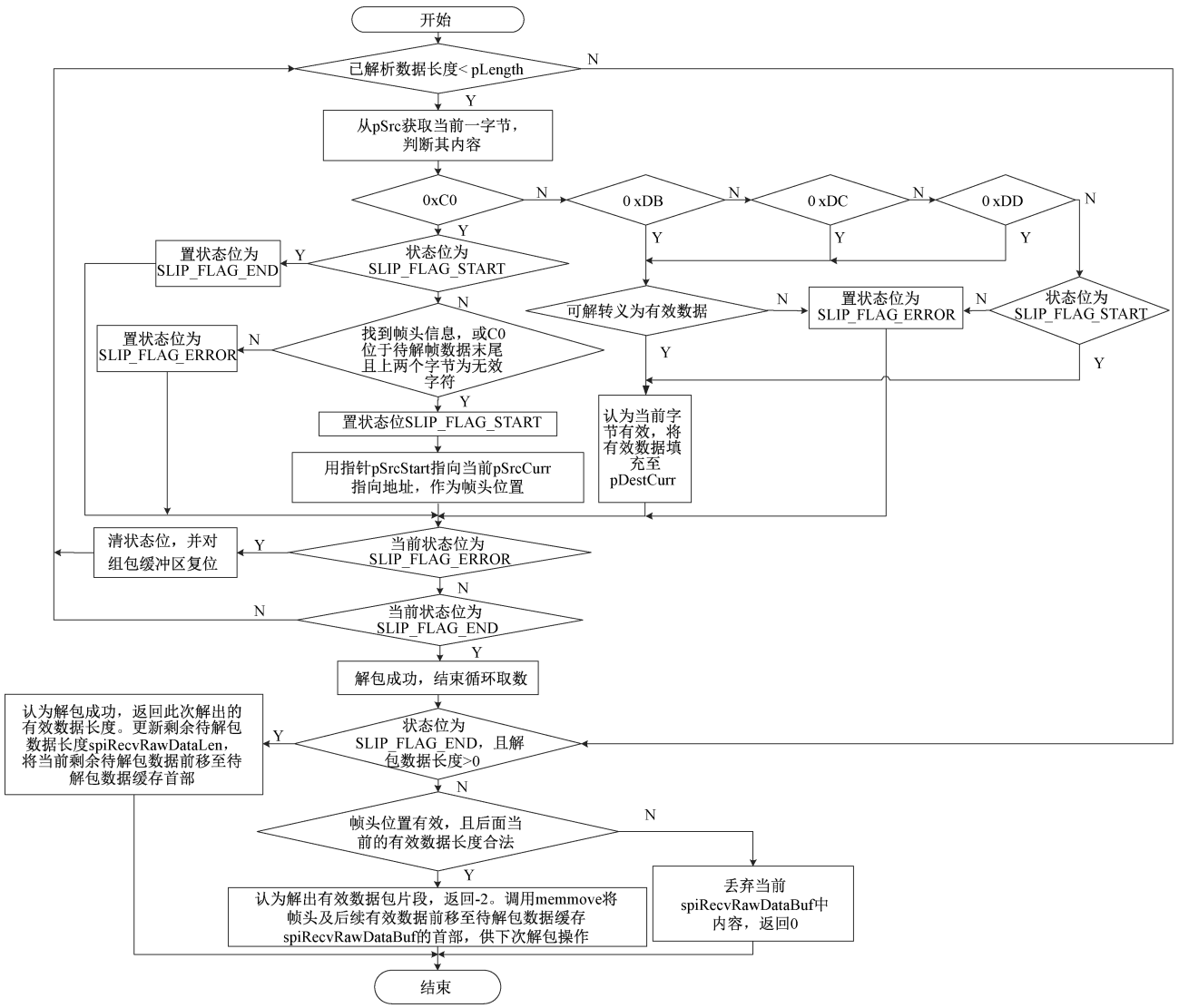

图 13　SLIP 解帧工作流程

Fig. 13　SLIP de-framing workflow

SPILlinkSlipUnEsc 中对待解析数据内容进行逐字节判断，并通过 3 种状态位标识当前解帧状态。状态位包括：帧起始（SLIP_FLAG_START）、帧结束（SLIP_FLAG_END）以及帧错误（SLIP_FLAG_ERROR）。待解帧数据中检测到帧头，则将状态位置为 START，并对后续待解帧内容按字节进行 SLIP 转义，直到检测到帧尾标志，将状态位置为 END，对待解帧数据缓存进行更新，并将解出的有效数据帧返回给上层调用。如果对待解析内容未解出完整数据帧，则将帧头位置作为新的待解析数据起始地址，供下次解帧使用。同时加入了异常处理机制，当解析到错误头信息时，会对状态位复位，清空当前待解析缓存，以保证下一包有效数据能够得到正常解帧处理。至此，CPU 便可将 SPI 总线上 MCU 发来的有效数据，即主控模块的 MCU 在 CAN 总线上接收到的应用层数据提取出来。

4　实验结果与分析

根据 ICNI 系统健康管理设计需求，主控模块由底层至顶层分别就：1）CPU 与 MCU 间 SPI 通信能力；2）基于双处理器 SPI 通信的 CAN 总线通信能力；3）主控模块基于 CAN 总线的 ICNI 系统健康管理功能 3 个方面进行实验设计，进行了逐级测试验证并得出结论。最后，对行业内其他健康管理功能电路的硬件成本进行了比较分析。

4.1　双处理器 SPI 通信功能验证

CPU 向 MCU 发送本地查询命令,MCU 会回复 CPU ACK 以及查询信息,CPU 在确认收到的信息 CRC 校验无误后,会回复 MCU ACK 应答信息。在 CPU 软件中编写测试样例,以 10 ms 为发送周期,向 MCU 下发本地查询指令。查询报文每包数据为 32 字节。测试并统计来自 MCU 的回复包数、误包数以及通信时间。

CPU 外部输入系统时钟为 48 MHz,由于分频系数必须为偶数,因此取 4、6、8、12 MHz 为 SPI 总线时钟频率 SCK 有效时钟速率进行测试。如表 2 所示对 CPU 与 MCU 间 SPI 通信在不同 SCK 时钟下进行测试。正常情况下,CPU 每发送一包查询命令,会收到来自 MCU 的 ACK 应答和查询内容两包数据。经测试,SCK 为在 12 MHz 时,收到的 MCU 数据包 CRC 校验开始出错,CPU 会将校验出错的数据包丢弃出现误包。在 SCK 选用 8 MHz 时,双处理器间的 SPI 通信可靠且速率最高。

表 2　SPI 通信测试

Table 2　SPI communication test

SPI 总线时钟频率 SCK	主机发送包数	主机收到无误的数据包数	误包率/%	时间/s
4 MHz	50 000	100 000	0	280
6 MHz	50 000	100 000	0	186
8 MHz	50 000	100 000	0	124
12 MHz	50 000	83 328	14.7	82

4.2　主控模块 CAN 通信验证

ICNI 主控模块可通过 CAN 总线对机架内其他模块在线升级,因此通过在线升级能力测试可验证主控模块与系统内其他模块的 CAN 通信能力。上位机通过以太网获取到本地待升级的运行体文件,传输到主控模块 CPU,CPU 将数据缓存、分包,并按 SPI 业务协议传输给 MCU。MCU 根据在线升级目标地址,通过 CAN 总线网络将数据传输给对端模块。对端模块收到总线命令和升级数据,对自身 FLASH 进行编程,实现在线升级[19]。传输过程中,对端会对收到的每包数据做校验,并发起 ACK 应答回传给 CPU。通过主控模块的在线升级业务也可对 CPU 与 MCU 间 SPI 全双工通信能力充分验证。系统内 CAN 总线速率采用 1 Mbps,在线升级功能性能测试如表 3 所示。

如图 14 所示,测试结果均为在线升级成功,没有误包重传,且具备对多个模块同时 CAN 总线升级的能力,在线升级速率满足用户的需求。通过上述测试证明,主控模块与系统内其他模块 CAN 总线通信正常。

4.3　ICNI 健康管理功能验证

使用 CAN 总线抓包软件,对 ICNI 主控模块与系统内其他模块的 CAN 总线通信数据进行抓包分析。主控模块通过 CAN 总线下发周期 BIT 查询命令,查询系统内模块的健康状态信息。CPU 将对应的 ICD 命令通过 SPI 总线转发给 MCU,再发送到总线上。设置查询目标的本地地址为电源模块 1(PS1)的 MARK 地址,可看到电源模块收到来自主控模块的 BIT 查询命令及 ACK 应答消息,如图 15 所示。电源模块 1 收到主控模块的 BIT 查询指令,回复 ACK 以及 BIT 查询结果。主控模块将收到的 BIT 查询结果呈报至上位机软件。观察上位机监控软件,可见电源模块 1 状态为上线状态。拔出系统内的 BM 模块 1,再次上电测试,可见上位机软件中 BM 模块 1 的状态指示为下线状态,如图 16。经测试,ICNI 健康管理系统具备监控、测量、上报系统内各模块状态信息的功能,满足设计需求。

表 3　CAN 总线在线升级速率测试

Table 3　Online upgrade rate test based on CAN bus

升级对象	文件大小/KB	升级时间/s	速率/(KB·s^{-1})	重传包数
PS1(电源模块)	231	54	4.28	0
LMIX(LMIX 功能模块)	257	64	4.02	0
PS1 & LMIX(同时升级)	488	121	4.03	0

图 14　LMIX 功能模块在线升级结果

Fig. 14　Online upgrade results of LMIX functional modules

4.4　元器件成本分析

如图 16 所示,本设计中,健康管理功能电路的最小系统,除基本的电源、时钟外仅使用 MCU 加一片 FLASH 静

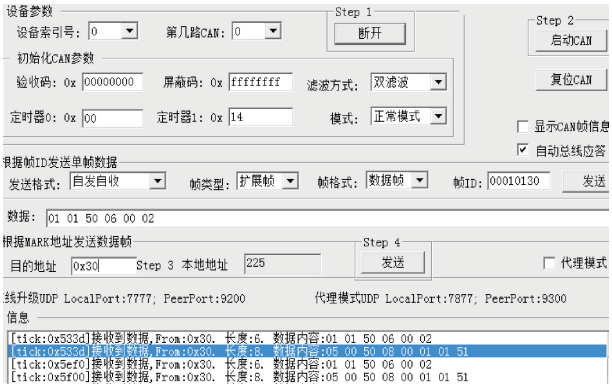

图 15　PS1 收到主控的 BIT 查询命令

Fig. 15　PS1 received a BIT query command from the controller

态存储器。在同等质量等级的国产化元器件选用要求下，本设计与行业内健康管理功能电路常选用的 DSP 最小系统(DSP+SRAM 内存)或 SOPC 最小系统(SOPC+DDR 内存+FLASH 静态存储器)的元器件总成本相比，总价降低至少 60%，单模块节约成本 6 000 元以上。

图 16　上位机中系统内各模块健康状态采集

Fig. 16　Health status collection of various modules in the upper computer system

5　结　　论

本设计在 ICNI 主控模块内基于 CPU+MCU 架构，通过 MCU 作为协处理单元完成主控 CPU 的 CAN 总线接口扩展，并基于 CPU 与 MCU 间的 SPI 全双工通信实现了一种基于 CAN 总线的健康管控系统，给出了国产化的软硬件详细设计。该设计对 CPU 接口资源要求仅为一组 SPI 接口和一根 GPIO 接口，硬件设计简单，节省 CPU 接口资源的同时减轻了模块布线布局和结构设计压力。同时不依赖 FPGA，为模块节省了逻辑资源。健康管理功能电路部分单套成本较业内同类产品相比成本更低。本设计为核心处理器的 CAN 总线接口扩展提供了一种解决方案，对机载系统主控模块设计有一定参考价值。

参考文献

[1]　王兴华，王军强，牛振中，等. 无人机综合模块化航空电子体系架构技术研究[J]. 航空计算技术，2021，51(5)：92-96.
WANG X H，WANG J Q，NIU ZH ZH，et al. Research of integration and modularization for UAV avionics system [J]. Aeronautical Computing Technique，2021，51(5)：92-96.

[2]　卢阳，杜增，谢林，等. 通用化 CNI 模块测试设备的设计与实现[J]. 电子设计工程，2023，31(7)：136-140.
LU Y，DU Z，XIE L. Designing and implementation of the universal CNI module test equipment [J]. Electronic Design Engineering，2023，31(7)：136-140.

[3]　文佳，钱东，梁天辰，等. 机载系统综合状态监测与诊断架构设计及应用[J]. 电讯技术，2023，63(7)：979-986.
WEN J，QIAN D，LIANG T CH，et al. Design and application of integrated condition monitoring and diagnostic architecture for airborne systems [J]. Telecommunication Engineering，2023，63(7)：979-986.

[4]　李思雨，程中华，刘子昌，等. 故障预测与健康管理系统研究与应用现状分析[J]. 火炮发射与控制学报，2023，44(6)：99-105.
LI S Y，CHENG ZH H，LIU Z CH，et al. Review and application status of prognostics and health management system[J]. Journal of Gun Launch & Control，2023，44(6)：99-105.

[5]　王博，仲维彬. 航空故障诊断与健康管理技术研究[J]. 现代导航，2019，6：454-457.
WANG B，ZHONG W B. Technology research on aviation fault diagnosis and health management[J]. Modern Navigation，2019，6：454-457.

[6] 冯旭.谈我国军用元器件自主可控的重要性[J].机电元件,2024,6(3):58-61.
FENG X. Discuss the importance of autonomous control of military components in China [J]. Electromechanical Components, 2024, 6(3): 58-61.

[7] 高艳,魏东梁,赵旭昌,等.一种交互式 CAN 通讯协议设计及软件实现方法[J].工业控制计算机,2024,37(5):7-8.
GAO Y, WEI D L, ZHAO X CH, et al. An interactive CAN communication protocol design and software implementation method[J]. Industrial Control Computer, 2024, 37(5): 7-8.

[8] 魏胜利,陶平,徐睿,等.基于 CAN 总线的故障诊断模块设计[J].测控技术,2019,38(1):29-32.
WEI SH L, TAO P, XU R, et al. Design of fault diagnosis module based on CAN[J]. Measurement & Control Technology, 2019, 38(1): 29-32.

[9] 代华山.基于模块化综合的通信导航识别系统设计[J].信息技术与信息化,2022,11:154-160.
DAI H SH. Design of communication navigation identification system based on modular synthesis[J]. Information Technology and Informatization, 2022, 11: 154-160.

[10] 张洪亮.特种飞机综合化任务系统 PHM 设计与实现[J].电讯技术,2021,61(3):291-297.
ZHANG H L. Design and implementation of PHM for integrated mission system of special aircrafts [J]. Telecommunication Engineering, 2021, 61(3): 291-297.

[11] 金东勇,陈俊霞,肖文光.基于 CAN 总线的健康管理系统设计[J].电子设计工程,2019,27(4):189-193.
JIN D Y, CHEN J X, XIAO W G. Design of health management system based on CAN bus[J]. Electronic Design Engineering, 2019, 27(4): 189-193.

[12] 吴根平,王浩,刘巍,等.基于 STM32 的 CAN 总线智能测控模块的设计[J].舰船电子工程,2020,307(1):117-120.
WU G P, WANG H, LIU W, et al. Design of intelligent input and output module based on STM32and CANBUS [J]. Ship Electronic Engineering, 2020, 307(1): 117-120.

[13] 方海伊,豆海利,冯源.基于 ZYNQ-7020 的 CAN 总线节点接口设计与实现[J].物联网技术,2023,1:56-59.
FANG H Y, DOU H L, FENG Y. Design and implementation of node interface of CAN bus based on ZYNQ-7020[J]. Internet of Things Technologies, 2023, 1: 56-59.

[14] 邱凯强,王轩,包文帆.基于 FT2000/4 处理器的健康管理硬件设计[J].电子制作,2023,14:7-9.
QIU K Q, WANG X, BAO W F. Health management hardware design based on FT2000/4 processor[J]. Practical Electronics, 2023, 14: 7-9.

[15] 许亚星,徐楠,门茜,等.一种基于 SPI 的数据交换方法研究[J].山西电子技术,2022,6:74-75.
XU Y X, XU N, MEN Q, et al. SPI-based data exchange method research [J]. Shanxi Electronic Technology, 2022, 6: 74-75.

[16] 李菁,刘澎,刘培文.基于 SPI 的信号完整性案例分析[J].电子设计工程,2022,30(23):108-111.
LI J, LIU P, LIU P W. The case analysising of signal integrity based on SPI [J]. Electronic Design Engineering, 2022, 30(23): 108-111.

[17] 项涛,吕大鹏,董延军.FT2000A 处理器嵌入式操作系统适配技术研究[J].信息通信,2019,3:110-112.
XIANG T, LYU D P, DONG Y J. Research on adaptation technology of embedded operating system for FT2000A processor [J]. Information & Communications, 2019, 3: 110-112.

[18] 李增科,李云鹏,席东学.基于 SPI 协议的双 DSP 通讯设计实现[J].电子测量技术,2020,10:159-164.
LI Z K, LI Y P, XI D X. Design and implementation of dual DSP communication based on SPI protocol[J]. Electronic Measurement Technology, 2020, 10: 159-164.

[19] 刘维,白金辉.一种改进的机载主控系统高可靠在线升级技术设计与实现[J].信息技术与信息化,2023,11:73-78.
LIU W, BAI J H. An improved design and implementation of highly reliable online upgrade technology for airborne master control system [J]. Information Technology and Informatization, 2023, 11: 73-78.

作者简介

刘维(通信作者),硕士,工程师,主要研究方向为航空总线技术,嵌入式软件开发等。
E-mail:975048578@qq.com