

四通道隔离 CAN 通讯模块设计

凤 雷¹, 陈 帅¹, 吴家顺², 张博亚³

(1. 哈尔滨工业大学, 哈尔滨 150001; 2. 中国舰船研究设计中心, 武汉 430000;
3. 火箭军装备部驻哈尔滨地区军事代表室, 哈尔滨 150001)

摘要: 针对 CAN 总线通讯系统对实时性和可靠性的要求, 文章设计了一种基于 CPCI 总线的四通道隔离 CAN 总线通讯模块; 该模块与现有的同类 CAN 总线设备相比, 在成本和电路集成度上具有明显的优势; 该模块在硬件上采用了单片 FPGA 来实现 CPCI 接口以及 4 个独立的 CAN 控制器逻辑, 并且使用了 4 个隔离的 CAN 收发器 CTM1051 同时实现电平转换与电气隔离; 在固件设计中, 分别采用 PCI 总线 IP 核和 CAN 总线 IP 核来实现 PCI 接口和 CAN 总线控制器; 在软件设计中, 按照模块技术要求来设计设备驱动程序、仪器驱动程序和软面板; 经实验测试表明, 该模块的位宽容忍度范围为 $\pm 5\%$ 、采样点特征值为 75% 并能连续正常工作 3 小时以上, 具有良好的性能指标和可靠的四通道数据收发功能。

关键词: CAN 总线; CPCI 总线; 电气隔离

Design of Four-Channel Isolated CAN Communication Module

Feng Lei¹, Chen Shuai¹, Wu Jiashun², Zhang Boya³

(1. Harbin Institute of Technology, Harbin 150001, China
2. China Ship Development and Design Center, Wuhan 430000, China
3. Military Office of Rocket Armaments Department in Harbin District, Harbin 150001, China)

Abstract: In view of the real-time and reliability requirements of CAN bus communication system, this paper designs a four-channel isolated CAN bus communication module based on CPCI bus. Comparing with existing similar CAN bus device, the designed module has obvious advantages in cost and circuit integration. The module adopts a single FPGA chip to implement CPCI interface as well as four independent CAN controller logic, and uses four isolated CAN transceivers CTM1051 to achieve level conversion and electrical isolation at the same time. In the firmware design, PCI bus IP core and CAN IP core realize PCI interface and the CAN bus controller respectively. In addition, device drivers, instrument drivers and soft panel are designed as the module requirement. Through the experimental test, the module has a bit width tolerance range of $\pm 5\%$, a sampling point characteristic value of 75%, and can work continuously for more than 3 hours. It has good performance index and reliable four-channel data receiving and sending functions.

Keywords: CAN; CPCI; electrical isolation

0 引言

控制器局域网 (Controller Area Network, CAN) 总线是由德国汽车系统供应商 Robert Bosch 于 19 世纪 80 年代中期开发的一种串行通信总线网络^[1], 凭借着较高的性能, 该总线吸引了越来越多的关注。由于其成本低、实时处理能力强、在强电磁干扰环境下工作可靠^[2], 其应用范围已从汽车电子控制扩展到工业自动化、交通运输、医疗设备、楼宇自动化等工业控制领域^[3-4]。如今, CAN 总线已通过国际标准 (ISO 11898) 认证, 并成为了最有前途的现场总线之一^[5]。

随着 CAN 总线系统复杂程度越来越高, 总线节点不断增加, 通信负荷不断加重, 采用单通道 CAN 总线通信难以满足实时性的要求^[6-7]。除此之外, 为了满足对通信可靠性

和容错性的日益严格的要求, 可行的技术手段之一就是采用冗余通信网络, 而该网络通常是以多通道通信技术为基础的^[8-9]。当一个 CAN 总线节点具有多个通道时, 用户可以更方便的进行冗余设计; 另一方面, 当数据传输比较繁忙时, 多个通道可以分担通信负荷。因此, 多通道 CAN 总线通信具有一定的必要性和良好的应用前景。本文设计了一种 CPCI 四通道隔离 CAN 总线通讯模块, 该模块不仅成本低、集成度高, 而且具有良好的实时性、可靠性和容错性。

1 系统结构及分析

设计的 CPCI 四通道 CAN 总线通信模块作为外围功能模块插在 CPCI 机箱中, 可构成如图 1 所示的 CPCI 总线系统结构。四通道 CAN 总线通讯模块作为节点连接外部

收稿日期: 2020-07-23; 修回日期: 2020-09-08。

作者简介: 凤 雷 (1978-), 男, 黑龙江哈尔滨人, 副教授, 硕士研究生导师, 主要从事自动测试系统集成技术与电子设备工艺可靠性方向的研究。

引用格式: 凤 雷, 陈 帅, 吴家顺, 等. 四通道隔离 CAN 通讯模块设计[J]. 计算机测量与控制, 2021, 29(3): 238-242, 257.

CAN 总线, CPCI 控制器运行 Windows 操作系统, 应用程序通过调用模块函数库中的接口函数实现对四通道 CAN 总线通讯模块通信功能的控制。在四通道 CAN 总线通讯模块的设计过程中, 需要考虑 CPCI 总线接口和四通道 CAN 总线接口的实现方式。

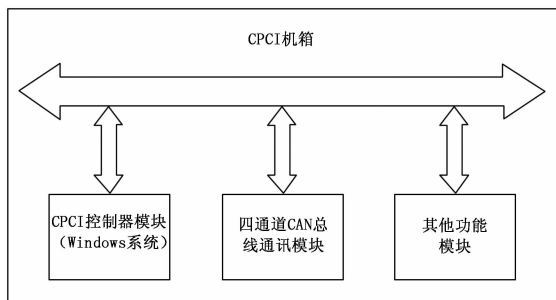


图 1 CPCI 总线系统结构

1.1 CPCI 接口

CPCI 总线接口用于和主机进行数据交互, 现如今有两种主流的实现方式:

1) 专用接口芯片: 例如 PLX 公司的 PCI9054 和 PCI9056。使用这些芯片的优点是可靠性高, 缺点是可能会造成一定的资源浪费, 并且缺乏设计灵活性。

2) 专用 IP 核: 使用 IP 核的优点是灵活性好, 开发周期短; 缺点是 IP 核的价格相对较高, 例如, Xilinx 的 LogiCore, Altera 的 MegaCore。

1.2 CAN 接口

CAN 接口作为连接处理器和 CAN 总线的桥梁, 主要由符合 CAN 总线协议规范的控制器和收发器组成。其中, CAN 控制器用于控制节点的数据通信, 完成收发报文与符合规范的 CAN 帧间的转换; 而 CAN 收发器的主要功能是完成控制器逻辑电平和总线差分电平间的转换, 实现物理层功能。

现如今 CAN 控制器有 3 种主流的实现方式:

1) 独立型 CAN 总线控制器芯片: 常见的有 NXP 公司的 SJA1000, 该控制器发展时间长, 产品性能好, 技术成熟, 但在多通道设计中, 其具有集成度低的缺点。

2) 集成有 CAN 总线控制器的单片机: 这种方式在一定程度上提高了集成度和可靠性, 但软件程序可移植性差, 而且单片机更新换代速度快, 产品的生命周期无法得到保障。

3) CAN IP 核: 采用 IP 核可以大大提高模块的集成度, 降低系统功耗, 提高产品的综合性能, 但核的价格普遍比较昂贵。

如今市场上有两种类型的 CAN 收发器可供选择:

1) 非隔离收发器: 常见的有 NXP 公司的 TJA1050。为了增强节点的抗干扰能力, 这种收发器在实现电气隔离时, 并不是直接和 CAN 控制器相连, 而是在两者之间加上高速光电耦合器件来实现电气隔离, 与此同时, 还必须在光耦的两端加上隔离的电源来保证光电耦合的有效性, 这

就需要设计额外的隔离电源模块, 增加了设计的复杂性。

2) 隔离收发器: 常用的隔离收发器有 ZLG 公司的 CTM1051 和 AD 公司的 ADM3052/3/4 系列, 此类芯片通常内置有隔离电源转换、信号隔离等电路, 具备较高集成度和可靠性, 减少了收发器外部辅助模块的设计工作。

在实际设计的过程中, 需要根据实际情况例如实现难度、成本和板卡尺寸等限制条件来选择 CPCI 接口、CAN 控制器和 CAN 收发器的实现方式。

2 硬件设计

硬件设计包括硬件电路设计和固件设计, 下面分别介绍这两个方面的设计工作。

2.1 硬件电路设计

根据设计指标要求, 该模块应设计为 PCI 总线 3U 尺寸的单槽仪器模块, 实现四通道 CAN 总线通讯接口, 符合 CAN 2.0B 版本规范, 波特率最高 1Mbps, CAN 总线接口电路可实现电气隔离。综合考虑成本、集成度、实现难度等因素, 本次设计选择 PCI IP 核来实现 CPCI 接口, CAN IP 核来实现 CAN 总线控制器, ZLG 公司的 CTM1051 来作为 CAN 总线收发器。硬件电路结构如图 2 所示。硬件电路这部分主要包括以 FPGA 为核心的最小系统的外围电路设计和四通道 CAN 总线接口设计。

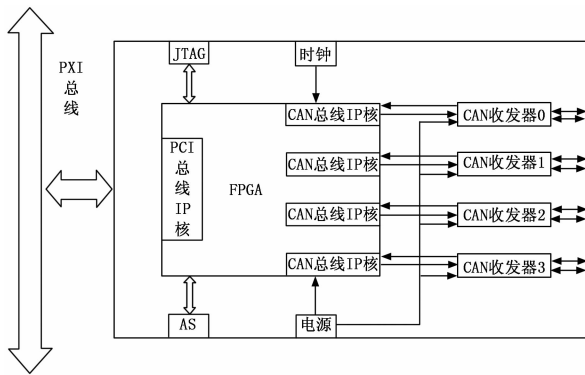


图 2 四通道 CAN 通讯模块的硬件电路结构

1) PCI IP 核: 使用 Altera 的开发工具 Quartus II 提供的 PCI 核, 并根据价格、I/O 资源和逻辑资源等条件来选择 Altera 的 FPGA 产品。通过在 Quartus II 中建立相关工程, 得到了该设计的整体资源占用情况, 考虑到实验室的库存, 最终选择了 Cyclone 系列的 FPGA EP1C12Q240C8, 这款 FPGA 具有 12 060 个逻辑单元、240 个管脚和 239 616 位的存储, 在满足设计要求的同时可以大大降低设计的成本。

2) CAN IP 核: 使用开源平台 OpenCores 提供的 CAN 总线 IP 核^[10], 这样一来可以节省成本。然而, 正如 OpenCores 网站的 bugtracker 页中所报告的那样, 该内核中仍然存在一些 bug。在文献 [11] 中, 作者通过搭建基于 UVM 的验证平台并采用适当的验证策略对该内核进行了验证, 并发现了其中存在的一些问题, 解决了发现的问题后, 成功通过了回归测试。根据给出的功能覆盖率和代码覆盖率可知修改后的 IP 核是可以使用的, 可用于实现 CAN 总

线控制器。

3) CAN 收发器：为了实现电气隔离并提高板卡集成度，决定采用 CAN 隔离收发器，常见的例如 ZLG 公司的 CTM1051 和 AD 公司的 ADM3053，为了进一步节约成本，最终选用国产芯片 CTM1051 来作为 CAN 收发器。

2.2 固件设计

固件部分主要是实现 PCI 总线接口和 CAN 总线接口并完成两者之间的连接，其在 SOPC Builder 中的连接如图 3 所示。

Use	Connections	Name	Description	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		clk_33m	Clock Source	clk_33m			
<input checked="" type="checkbox"/>		pci_compler_0	PCI Compier	clk_33m			
<input checked="" type="checkbox"/>		avlon_cblk_connection	Clock Input	clk_33m			
<input checked="" type="checkbox"/>		bar1_non_Prefetchable	Avalon Memory Mapped Master	[avlon_cblk_connection]			
<input checked="" type="checkbox"/>		bar1_Prefetchable	Avalon Memory Mapped Master	[avlon_cblk_connection]			
<input checked="" type="checkbox"/>		Control_Regiter_Access	Avalon Memory Mapped Slave	[avlon_cblk_connection]	0x00000000	0x00003fff	
<input checked="" type="checkbox"/>		Crainr_0	Interrupt Sender	[avlon_cblk_connection]			
<input checked="" type="checkbox"/>		Nmirq_1	Interrupt Receiver	[avlon_cblk_connection]	192 0	192 0	192 0
<input checked="" type="checkbox"/>		clk_16m	Clock Source	clk_16m			
<input checked="" type="checkbox"/>		clk	Clock Output	clk_16m			
<input checked="" type="checkbox"/>		can_av_top_0	can_av_top	clk_16m			
<input checked="" type="checkbox"/>		av	Avalon Memory Mapped Slave	[clk_33m]	0x00000000	0x000000ff	
<input checked="" type="checkbox"/>		clk_33m	Clock Input	clk_33m			
<input checked="" type="checkbox"/>		clk_16m	Clock Input	clk_16m			
<input checked="" type="checkbox"/>		can_av_top_1	can_av_top	clk_16m			
<input checked="" type="checkbox"/>		av	Avalon Memory Mapped Slave	[clk_33m]	0x00000100	0x000001ff	
<input checked="" type="checkbox"/>		clk_33m	Clock Input	clk_33m			
<input checked="" type="checkbox"/>		clk_16m	Clock Input	clk_16m			
<input checked="" type="checkbox"/>		can_av_top_2	can_av_top	clk_16m			
<input checked="" type="checkbox"/>		av	Avalon Memory Mapped Slave	[clk_33m]	0x00000200	0x000002ff	
<input checked="" type="checkbox"/>		clk_33m	Clock Input	clk_33m			
<input checked="" type="checkbox"/>		clk_16m	Clock Input	clk_16m			
<input checked="" type="checkbox"/>		can_av_top_3	can_av_top	clk_16m			
<input checked="" type="checkbox"/>		av	Avalon Memory Mapped Slave	[clk_33m]	0x00000300	0x000003ff	
<input checked="" type="checkbox"/>		clk_33m	Clock Input	clk_33m			
<input checked="" type="checkbox"/>		clk_16m	Clock Input	clk_16m			
<input checked="" type="checkbox"/>		irq_0	irq	clk_16m			
<input checked="" type="checkbox"/>		avlon_slave_0	Avlon Memory Mapped Slave	[cblk]	0x00000400	0x00000407	
<input checked="" type="checkbox"/>		clock	Clock Input	clk_33m			
<input checked="" type="checkbox"/>		interrup_sender	interrup Sender	clk_33m			

图 3 固件连接框图

1) 时钟：该模块使用 33 MHz 和 16 MHz 两个时钟，其中 33 MHz 来自机箱背板，提供给 PCI 总线和 Avalon 总线使用；16 MHz 时钟来自外部晶振，提供给 CAN 总线 IP 核使用。

2) PCI 总线 IP 核：采用的是 SOPC Builder 中自带的 PCI 总线 IP 核。在调用 PCI 总线 IP 核之前，需要正确设置该 IP 核的相关属性。

3) CAN 总线 IP 核：采用 OpenCores 开源平台提供的 CAN IP 核来实现 CAN 控制器。为了解决 IP 核中存在的一些问题，其源代码已经经过了修改。

4) 中断处理：功能是把来自于四路 CAN 总线的中断信号统一连接到本地 Avalon 总线的中断线上。

3 软件设计

为了实现四通道 CAN 总线通讯功能，需要在硬件设计的基础上进行相应的软件设计。首先，使用 NI-VISA 提供的工具 VISA Driver Development Wizard 来生成设备驱动程序，这个过程包括填写厂家代码和模块序列号、生成中断程序等操作。接下来介绍在 Windows 操作系统下利用 LabWindows/CVI 进行仪器驱动程序和应用程序设计的过程。

3.1 仪器驱动程序设计

根据 VPP 仪器驱动程序内部设计模型的要求，设计得到了该模块的函数树，如图 4 所示。在参考文献 [12] 中提到的容错冗余就可以通过图 4 中所示的仪器驱动程序来

实现。

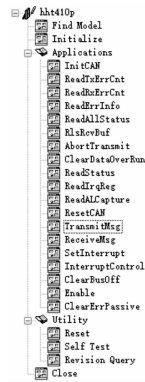


图 4 函数树

本文主要介绍使用仪器驱动程序设计的用于 CAN 总线通讯的 3 个主要程序。

1) CAN 通道初始化程序：通过配置 CAN 控制器的相关寄存器为后续数据传输做准备，流程图如图 5 所示。首先设置模式寄存器为复位模式，随后依次对 CAN 控制器中的多个寄存器进行配置，分别完成时钟分频器、滤波模式、验证和屏蔽码、中断使能以及波特率等属性的设置，最后进入工作模式并注册开启中断^[13]。通过以上步骤实现 CAN 通道初始化后，即可等待进行报文接收或发送操作。

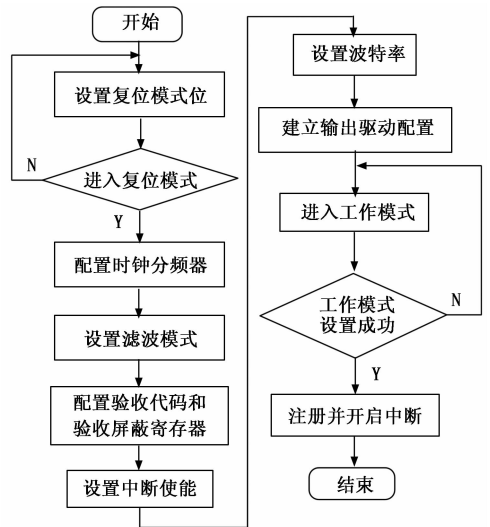


图 5 通道初始化程序流程图

2) CAN 报文发送程序：CAN 报文发送程序负责把报文发送到 CAN 总线上。报文发送之前得先读取 CAN 状态寄存器的值，只有在满足总线空闲、上一次报文发送完成和 CPU 可以向发送缓冲器写数据的条件下，才能将 CAN 报文结构体里设置的信息写到发送缓冲区并发送，发送每帧报文后，发送缓冲区都会自动清空，以备下一个要发送的报文。其流程图如图 6 所示。

3) CAN 报文接收程序：CAN 报文接收程序采用中断方式实现，负责接收总线上其他 CAN 节点发来的数据，流程图如图 7 所示。CAN 控制器完成数据接收后会产生相应

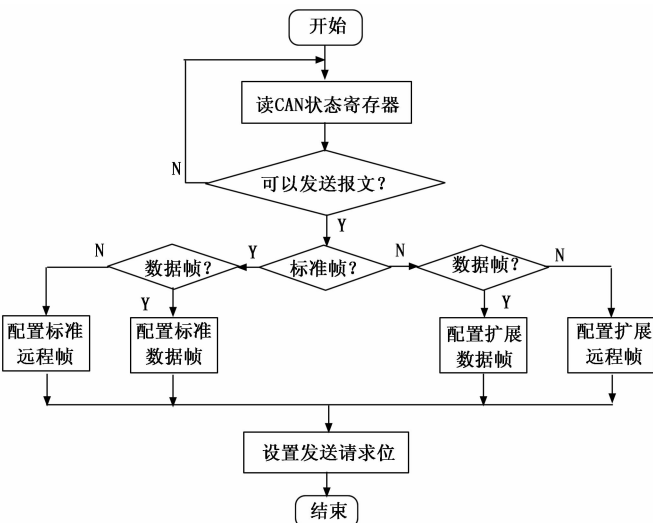


图 6 报文发送程序流程图

中断并跳转到中断服务程序, 在中断服务程序中首先关闭中断, 然后访问中断状态寄存器判断当前中断类型, 如果不是接收中断则需执行其他中断处理; 否则读取接收缓冲区中缓存的 CAN 总线报文, 读取完成后释放缓冲区用于下一次数据接收, 最后重新开启中断并退出中断服务程序^[13]。

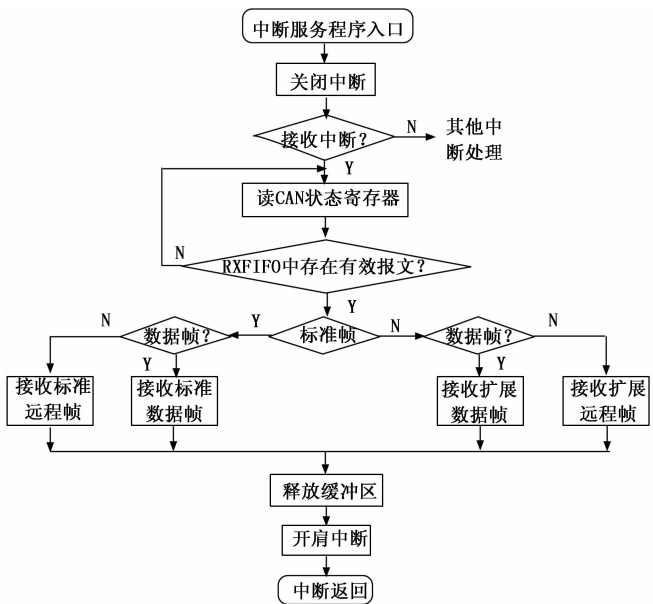


图 7 报文接收程序流程图

实际应用中, 由于数据量大, 在数据传输的过程中经常会出现数据丢失。为了尽量避免这种情况发生, 就需要进行数据缓存。常用的方法有两种:

- 1) 使用片上存储器: 例如 SRAM、DRAM 等。通过编写 Verilog 代码将片上存储器变成 FIFO。读取的数据直接存储在 FIFO 中, 然后主机通过 CPCI 接口从 FIFO 中读取数据。这样可以节省计算机的存储资源, 但是实现起来比较困难, 并且不利于集成度的提高。
- 2) 使用计算机的存储器: 在主机中, 一部分物理内存

可以通过 C 语言代码实现位 FIFO。读取的数据首先通过 CPCI 接口存储在 FIFO 中, 然后显示在软面板上。这种方式实现起来比较容易, 但是会消耗一部分计算机内存资源。

由于使用的 CPCI 机箱的嵌入式控制器具有足够的内存资源, 为了缩短开发周期并提高集成度, 最终采用第二种方式来进行数据缓存, 实际设计的缓存大小为 130 000 帧。

3.2 软面板设计

软面板主要通过调用模块的仪器驱动程序来实现 UI 功能, 例如初始化、消息发送和消息接收等。除此之外, 为了便于人机交互, 还设计了数据的实时显示和数据保存功能。设计的软面板如图 8 所示。



图 8 软面板

4 实验结果与分析

测试的目的是为了验证模块各项功能的正确性。为此, 搭建了如图 9 所示的测试平台。在测试平台中, 设计的四通道 CAN 总线通讯模块作为一个外围模块插在 CPCI 机箱中, 采用 ZLG 公司的 CAN 总线分析仪 CANScope^[14-16]和配套的模拟信号测试扩展板 CANStressZ 对该模块进行相关测试。CANScope 和计算机之间通过 USB 连接, 在计算机中安装配套软件 CANScope 来控制分析仪执行相关操作和测试。

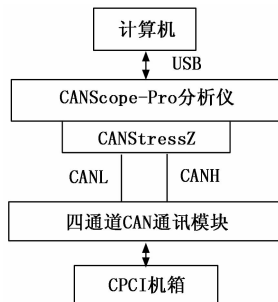


图 9 测试平台结构

在本次测试中, 测试了该模块的一些重要性能指标。除此之外, 还测试了模块在一些误操作或人为干扰等条件下的工作情况, 本文将主要介绍采样点测试、位宽容度测试、短路测试和开路测试。

4.1 采样点测试

信号位采样点是指 CAN 节点识别一个电平逻辑的位置。一般 CAN 节点都是采用一次采样, 也就是一个位采样

一次。由于 CAN 电平在变化的时候，都会出现抖动等“反射现象”，所以为了保证采样准确，采样点位置都在一个位的 50% 以后的区域，因为这是电平已经趋于静态，采样的逻辑值错误率也就比较低。

但采样点的位置也不是越往后越好，当超过 95% 时，如果有通讯节点间有位宽偏差，就很容易发生错误。所以 CANOPEN 国际组织（CANOPEN International Agency, CIA）规定采样点推荐值为 87.5%。

在多节点通信中，如果采样点适应范围无交集，会导致 CAN 重同步失败，通讯失败率增加，所以测试每个节点的采样点范围，对于设备兼容性和系统稳定性有着重要意义。

根据 CANScope 的用户手册可得采样点位置的评价标准，如表 1 所示。

表 1 采样点位置评价

采样点位置	评价
>90%	太靠后
80%~90%	良好(适合远距离传输)
70%~80%	良好(兼容性好)
<70%	太靠前

通过 CANScope 软件的采样点测试功能，得到了该模块的采样点测试结果，如图 10 所示，箭头区域是采样点的适用范围。可以看出，该模块的采样点特征值为 75%，采样点适应范围为 70%~80%，也就是说，只有当其他节点的采样点在此范围内时，它们才能与该模块通信。结合表 1 中的评价标准可知，该模块具有较好的兼容性。

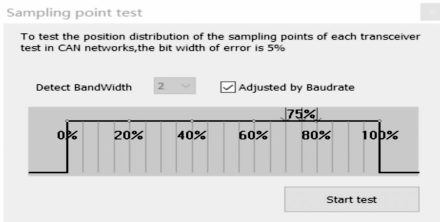


图 10 采样点测试结果

4.2 位宽容忍度测试

波特率就是信号位的最小脉宽，是保证 CAN 总线正常通讯的基础。波特率不匹配或者波特率存在偏差会导致 CAN 节点对总线信号的识别错误，造成总线通讯异常或者无法通讯。因此在对任何 CAN 节点进行测试标定时，第一步都需要进行信号的位宽度和波特率适应范围的测试，标定出被测试节点的位宽度和波特率适应范围，保证出厂后可以兼容所有的网络。

在测量波特率适应范围时，该模块不主动发送数据，通过软件的位宽容忍度测试功能来控制 CANScope 发送测试报文，得到该模块在常用的 1 000 kbps 下的位宽容忍度测试结果如图 11 所示。

位宽度和波特率适应范围的测试，体现了一个 CAN 节

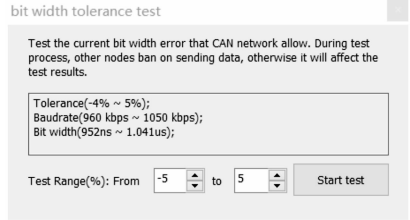


图 11 容忍度测试结果

点波特率的准确性和适应性，描述了节点在温度变化、网络其他节点波特率偏差情况下的兼容能力。根据各大汽车厂商的标准，位宽容忍度范围 $\geq \pm 3\%$ 即可认为该节点的位宽容忍度测试达标，因此，根据上述测试结果可知该模块的位宽容忍度指标已经达标。

4.3 短路测试

在 CANScope 软件中，可以通过配置 CANStressZ 扩展板来让 CAN 网络处于短路状态，然后让被测模块发送数据，示波器窗口显示的波形如图 12 所示。可以看到波形已经差到无法完成正常的通信。在短路状态被解除后，可以得到正常数据传输时的波形如图 13 所示。图 13 是软件 CANWave 窗口显示的波形，包含了差分电平、逻辑电平以及对报文的解析。

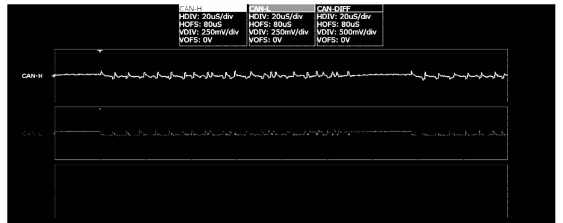


图 12 短路测试结果

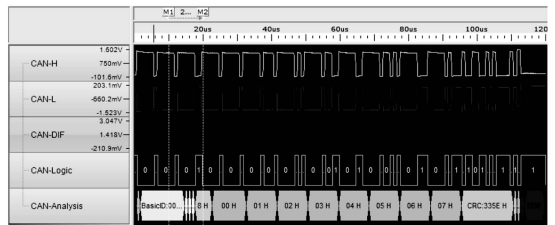


图 13 正常通信 CANWave 窗口波形

4.4 断路测试

在 CANScope 软件中，也可以通过配置 CANStressZ 扩展板来让 CAN 网络处于断路状态。然后，待测模块发送数据，得到的波形如图 14 所示。波形同样已经无法完成正常的通信。在解除断路状态后，又可以恢复图 13 中的正常数据传输时的波形。

除了上述测试外，为了测试模块在长时间工作条件下的稳定性，还进行了相关实验：采用扩展远程帧，数据内容固定，帧 ID 逐帧递增。被测模块每 50 ms 发送一帧数据，与此同时，CANScope 也每 50 ms 发送一帧数据，连续工作

(下转第 257 页)