

无人机仿真系统 CAN 总线调度及注册机制设计

贾振宇, 李春涛, 李妍

(南京航空航天大学 自动化学院, 南京 210016)

摘要: 针对无人机仿真系统传输数据量大, 实时性要求高等特点, 以基于 CAN 总线的飞行仿真系统为平台, 设计了总线时间调度以及状态检测机制, 对数据进行分组分时传输, 保证总线通信的实时性, 为仿真测试提供了一套正确有效的验证平台; 针对传统仿真系统接口资源固定, 不利于更改升级的问题, 设计了接口资源注册机制, 根据各接口节点的注册信息建立系统接口资源表, 进行统一管理, 提高了系统的灵活性, 便于仿真系统的开发。

关键词: CAN 总线; 飞行仿真系统; 时间调度; 状态检测; 接口注册

Design of CAN Bus Scheduling and Registration Mechanism for UAV Simulation System

Jia Zhenyu, Li Chuntao, Li Yan

(College of Automation, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)

Abstract: The UAV Simulation System has the characteristics of large amount of data and real-time requirements. The experimental platform is based on the hardware of a sample flight simulation system which communication is based on CAN bus. According to the characteristics of CAN bus, the mechanism of bus time scheduling and state detection is designed, which can guarantee the real-time performance of the communication, and provides a correct and effective verification platform for the simulation test. And as interfaces of traditional simulation system are fixed, it is not easy to be upgraded. The registration mechanism of interfaces is designed to improve flexibility of the system, which is easy to develop the simulation system.

Keywords: CAN bus; flight simulation system; time scheduling; state detection; interface registration

0 引言

制系统控制律与控制逻辑正确性和可靠性的测试平台, 飞行仿真平台对实时性有较高的要求, 以保证仿真结果的正确性。国内对无人机仿真系统的研究较为广泛, 西北工业大学搭建了基于 PC104 和 VxWorks 的飞行仿真系统^[1], 并对分布式飞行仿真技术进行了研究^[2]; 国防科技大学利用快速原型技术设计了实时分布式飞行仿真系统^[3]; 南京航空航天大学采用快速原型与 VxWorks 相结合的方式, 设计了实时仿真系统, 以保证系统实时性^[4]。

对于复杂的仿真系统, 仿真对象包括传感器、舵机和无人机相关子系统及设备, 仿真过程中会产生大量仿真数据交互, 如果不对数据交互进行统一调度, 大量数据交互存在数据传输冲突, 影响仿真系统的实时性。与此同时, 传统仿真系统接口资源相对固定, 主要针对某一固定类型无人机仿真对象。虽然传统仿真系统留有相应的备份接口资源, 但当接口资源变化较大时, 备份接口资源不能满足需求, 需要变更硬件和底层驱动, 工作量大且不够灵活, 不利于仿真系统的开发。

针对上述两个问题, 本文以基于 CAN 总线的某样例无人机仿真系统(以下简称仿真系统)为硬件平台, 根据 CAN 总

线的传输特性, 设计了主从静态分时调度机制以及状态检测机制, 对总线通信进行合理调度, 避免数据传输冲突, 保证传输的实时性和可靠性; 同时设计了接口资源注册机制, 对接口资源进行统一管理, 实现接口资源可扩展, 提高了仿真系统的灵活性, 为仿真系统的开发提供了方便。

1 仿真系统平台

仿真系统不仅要求仿真系统运行解算具有较高的实时性, 同样要求内部通信具有较高的实时性和可靠性, 对内部通信总线提出了较高要求。

目前工业上应用的总线有很多, 包括 RS485、I²C、1553B、CAN 等总线, 其中 CAN 总线凭借其实时性和可靠性等特点, 已经在航空工业、工业控制、安全防护、农业生产等领域得到了广泛应用^[5]。和其他总线相比, CAN 总线具有实时性高、可靠性高、受环境干扰小以及配置灵活等优点。

基于 CAN 总线通信的仿真系统平台如图 1 所示。为了保证模型解算的实时性, 减轻 CPU 单元的负担, 采用多块板卡分工协作的方式, 将仿真系统分为 CPU 单元和接口单元, CPU 单元负责模型解算和任务调度等功能, 接口单元负责与外界进行数据交换。CPU 单元与各接口单元之间通过内部 CAN 总线进行数据传输, 系统内部的数据通信为 CPU 单元与接口单元的数据交互, 各接口单元之间不存在数据交互, 故将 CPU 单元作为主节点, 各接口单元作为从节点, 各个节点挂接到 CAN 总线上。CAN 总线可分为上行总线和下行总线, 分别传输 CPU 单元向接口单元流向的数据和接口单元向 CPU 单元流向的数据, 以减轻 CAN 总线负载。

收稿日期: 2015-08-11; 修回日期: 2015-09-11。

作者简介: 贾振宇(1991-), 男, 山东菏泽人, 研究生, 主要从事飞行器控制方向的研究。

李春涛(1975-), 男, 山东临沂人, 副研究员, 主要从事无人机飞行控制方向的研究。

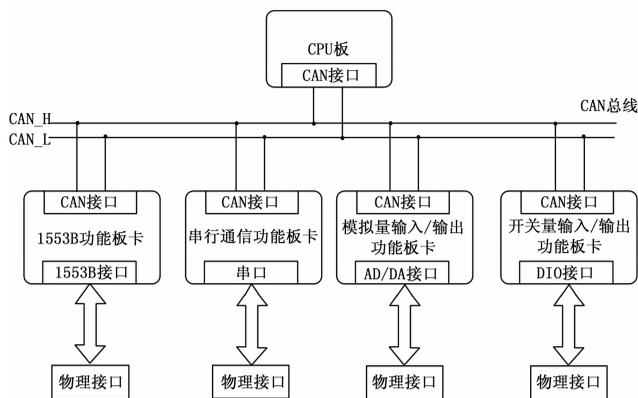


图 1 仿真系统结构示意图

2 CAN 总线通信机制设计

2.1 CAN 总线时间调度设计

仿真系统内部 CAN 总线分为上行总线和下行总线，上行总线数据传输方式为 CPU 单元向多个接口单元发送数据，由 CPU 单元确定数据发送时机，而下行总线是多个接口单元向 CPU 单元发送数据，总线数据传输时机不确定，具有随机性，存在同一时间几个接口单元同时发送数据的情况，虽然 CAN 自身带有非破坏仲裁机制，可以保证所有消息能够根据其各自的优先级进行连续发送^[5]，但延时时间不确定且总线负载不均衡，存在负载时高时低的情况。为了保证上行总线传输数据的实时性以及数据传输时机的可确定性，采用基于主节点时间的分时调度方式对整个内部总线通信网络进行管理^[6]，以避免总线数据传输的冲突。

总线分时调度时，主节点为每个节点提供一个时间基准，用来作为各节点发送消息的参考时间，不同类型的消息在各自特定的时间段进行发送，而无需和总线上的其他任何消息进行竞争，这样就可以避免丢失仲裁，同时延迟时间是确定的^[5-6]。

将 CPU 单元的硬件定时器作为全局时间的时钟定时器，定义 2 ms 为数据通信的原子周期，10 ms 为数据通信的基本周期，一个基本周期包括 5 个原子周期。每经过一个原子周期，主节点将通过总线广播一帧包含当前原子周期序号的时间同步帧，从节点根据原子周期序号判断当前所在的时间窗口。在一个基本周期中针对不同类型的接口数据分配不同的时间片，接口单元接收 CPU 单元广播的时间同步帧，根据时间同步帧中的原子周期序号判断当前是否是数据上传时间窗口，如果不是自身接口节点上传数据时间窗口，等待其他节点上传数据；如果是自身上传数据时间窗口，则上传数据。

如图 2 所示， t_1 时刻 CPU 单元发送时间同步帧，1553B 接口单元接收到时间同步帧进行对比，确认当前为 1553B 接口数据发送时间窗口，开始发送数据；而其他接口单元接收到时间同步帧对比后，确认不是对应接口的时间窗口，不发送数据。同样的，在 t_2 、 t_3 时刻，只有与时间窗口对应的接口单元发送数据，其他接口单元不发送数据。

分配至各通信接口单元专属时间窗口的多少由该单元所需上传数据量的大小所决定，表 1 是各接口的单位时间传输的数据量，可以看出串行通信接口单元的数据量最大而 1553B 接口单元、开关量接口单元和模拟量单元数据量较小。通信原子

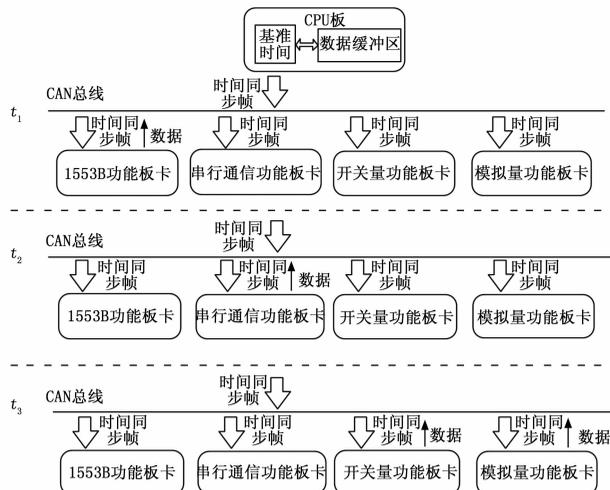


图 2 接口单元分时发送数据

周期为 2 ms，每个节点上传数据时间至少为 2 ms，按照 CAN 总线最大负载的 50% 计算，2 ms 内 CAN 总线可传输约 70 字节数据，可满足接口传输数据量的要求，因此接口类型时间窗口均为一个原子周期。

表 1 接口数据量

接口类型	单位时间传输数据量 (字节/秒)	原子周期平均数据量 (字节/2ms)
1553B	5 160	11
串口	23 920	48
开关量	400	1
模拟量	2 400	5

图 3 为 CAN 总线数据通信周期内各通信接口单元数据调度时间片分配示意图，将第 2 时间片分配给 1553B 接口单元，第 3 时间片分配给串行通信接口单元，模拟量接口单元与开关量接口单元共用第 4 时间片，第 5 个时间片为备用以便后期扩展，而第 1 时间片用于总线节点的状态检测，每两个基本周期进行一次状态检测。



图 3 总线时间调度时间片分布

2.2 状态检测机制设计

CPU 单元和接口单元采用了状态检测机制，即心跳机制。心跳是各通信模块定时发出的状态报告信号，也可以根据系统需求定义为一次设备握手操作^[7]。在仿真过程中，CPU 单元定时向各接口单元发送检测信号并接收接口单元反馈的状态信号，CPU 单元判断状态信号有无和状态信号内容，便可以识别出接口单元的工作状态，上报给仿真管理计算机。

这里采用了基于“请求—应答”方式的故障检测机制^[8]，包括的状态检测请求帧以及状态检测应答帧。状态检测的周期

为两个基本周期, 在第一个基本周期的第 1 时间片进行状态检测, 如图 3 所示。主节点周期发送状态检测请求帧, 各从节点在每个状态检测周期成功接收到该请求帧后需在同一个时间窗内返回一个状态检测应答帧。主节点在 3 个状态检测周期未能接收到某个节点状态检测应答帧, 说明接口单元至少 6 个周期内未传输数据或传输的数据有误, 认为该节点故障。

主节点周期向上下行总线发送状态检测帧, 检测上下行数据链路的状态。故障类型如表 2 所示。当节点数据链路故障时, 可通过软件对控制器进行复位重启, 而节点单元异常和数据链路断路故障时, 需要更换相应的接口单元并进一步检查硬件是否正常。

表 2 故障分类

故障类型	故障定义	故障处置
节点数据链路故障	节点无状态检测应答帧	节点总线控制器复位重启
节点单元异常	节点链路故障无法清除	更换节点板卡
数据链路断路	主节点未接收到任何节点状态检测反馈帧	检测总线数据链路

3 接口单元注册机制设计

为了提高仿真系统开发的灵活性, 实现对接口单元的灵活添加和删除, 根据仿真系统的结构特点, 采用接口单元上报资源信息的注册机制, 实现 CPU 单元动态获取各接口单元资源信息。

系统启动后, 各接口单元会向 CPU 单元发送资源注册帧, 资源注册帧包括接口单元类型 (串口、开关量、1553B 等)、设备号 (板卡唯一标识)、接口资源以及传输数据分配的 CAN 总线 ID 标识符。CPU 单元根据接收到的注册信息建立系统接口单元表和系统接口资源表, CPU 单元通过维护这两张表, 实现对接口单元的管理和调用。

系统接口单元表包含了当前注册的所有接口单元的基本信息, CPU 单元可以查询此表获取当前可用接口资源。表中每个接口单元用一个结构体进行表示, 该结构体定义如下:

```
typedef struct
{
    链接指针;
    接口单元类型;
    设备号;
    输入接口个数;
    输出接口个数;
}DEV_Struct;
```

该结构体给出了链接指针、接口单元类型、设备号和接口资源的个数, 结构相对简单, 只存储了一些关键信息。整个系统接口单元表中的接口单元结构通过指针链接在一起, 有新的接口单元注册时, 直接将相应的单元结构体添加到队列中。如图 4 所示是系统接口单元表的示意图。

当仿真系统接口需求变更, 对接口单元进行删减时, 系统接口资源表也随之变化。系统中可能存在多个同类型的接口单元, 虽然接口类型和资源一样, 但每个接口单元的设备号唯一, CPU 单元可以根据设备号确定唯一的接口通道。

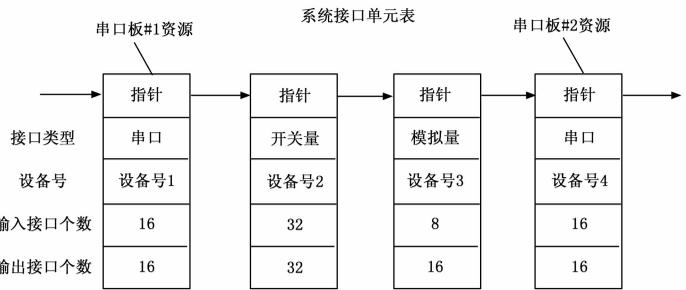


图 4 系统接口单元表

CPU 单元建立系统接口单元表的同时, 会根据设备号建立系统接口资源表, 用于存储分配给各接口的 CAN 总线 ID 标识符, 每个接口通道都有自己唯一的 ID 标识符。接口资源表分为两种类型: 接收接口资源表和发送接口资源表, 分别用来存放用于接收、发送的接口通道分配的 ID 标识符。接口资源表结构如表 3 所示。

表 3 系统接口资源表

设备号	通道 1	通道 2	通道 3	通道 4
设备号 1	ID_1	NULL	ID_2	ID_3
设备号 2	ID_4	ID_5	ID_6	ID_7
设备号 3	NULL	NULL	ID_8	ID_9
.....

当 CPU 单元要通过某个接口通道发送数据时, 首先会查询系统接口单元表, 找到对应接口所在单元的设备号, 根据设备号与通道号在系统发送接口资源表中找到对应的 CAN 总线 ID 标识符, 将数据组帧, 通过 CAN 总线发送给对应接口单元。具体流程如图 5 所示。

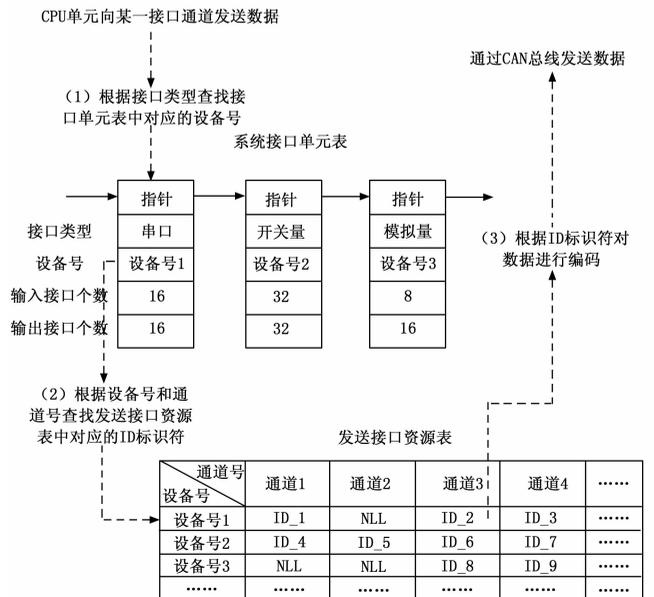


图 5 CPU 单元发送数据过程

相应的, 当 CPU 单元接收到数据时, 需要判断数据来自哪个接口通道, CPU 单元会对比系统接收接口资源表中的 ID 标识符, 找到对应的设备号和通道号, 再根据设备号查询系统

接口单元表，匹配接口类型。接口类型和通道号可以唯一确定数据来源，从而将接收到的数据存放到对应的数据缓冲区中，由相关处理程序进行解码处理。具体流程如图 6 所示。

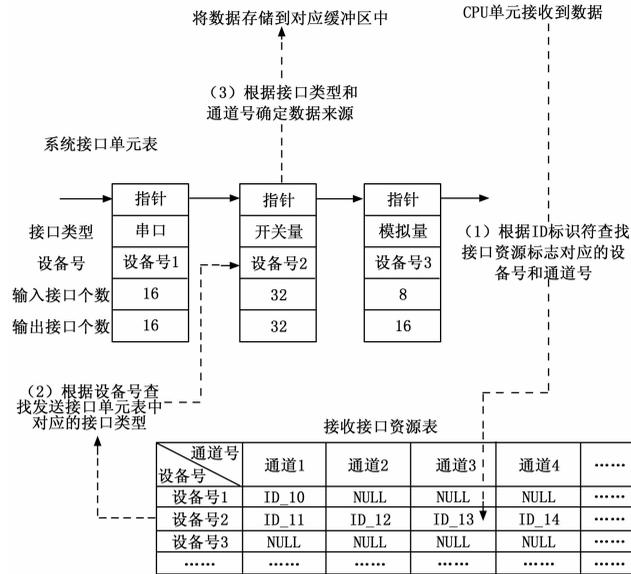


图 6 CPU 单元接收数据过程

4 系统测试

为了验证仿真系统 CAN 总线通信机制设计的性能和正确性，本文从 CAN 总线通信性能、总线时间调度两个方面进行测试^[9-11]。

4.1 CAN 总线可靠性测试

为了验证 CAN 总线的传输数据的可靠性，搭建了如图 7 所示的测试平台。CPU 单元作为主节点通过下行总线发送测试数据，接口单元作为从节点接收测试数据并通过上行总线转发测试数据，由 CAN 监测设备监测总线上的数据，与测试数据进行对比分析。主节点以 2 ms 的时间周期发送 10 帧测试数据，使总线在约 50% 的负载情况下进行 4 小时通信测试。

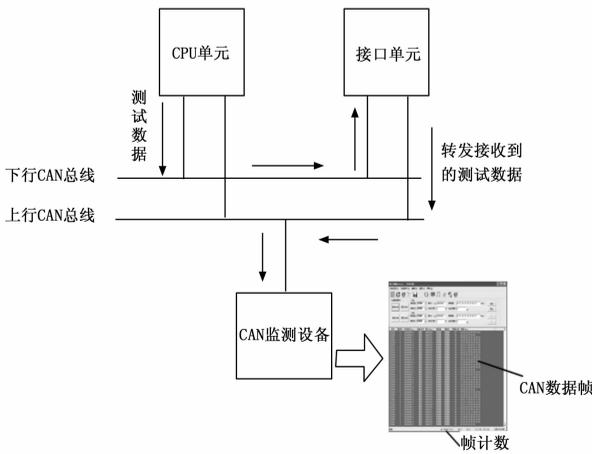


图 7 CAN 测试结构

总线测试情况统计如表 4 所示，可以看出通信过程中丢帧、错帧计数均为 0，表明 CAN 总线通信可以保证通信的可

靠性，满足飞行仿真系统通信的可靠性要求。

表 4 CAN 总线通信测试

通信时间(h)	帧计数	丢帧计数	错帧计数
1	18000000	0	0
2	36000000	0	0
3	54000000	0	0
4	72000000	0	0

4.2 总线时间调度测试

为了验证总线收发数据的调度情况，搭建起半物理仿真测试环境，将飞行控制系统与仿真平台连接，进行闭环仿真测试。在测试过程中对 CAN 总线收发数据任务进行了监测。

如表 5 所示，统计部分仿真发送任务和接收任务执行时间和运行周期。由测试结果可知，仿真系统中总线收发任务都能按照周期要求运行，且都能在其周期时间内执行完毕，可以说明总线任务调度正确。

表 5 部分收发任务统计结果

任务名	最大时间 /μs	最小时间 /μs	平均时间 /μs	周期 /ms	说明
TxMINS	46.63	17.34	31.70	20	惯导数据发送任务
TxGPS	54.78	30.48	37.60	50	GPS 数据发送任务
TxGyro	20.47	20.15	20.28	50	速率陀螺发送任务
TxAtmos	47.77	20.66	32.81	50	大气数据发送任务
RxServo	21.01	5.61	8.7225	20	舵机数据接收任务
RxMINS	17.49	15.01	16.36	10	惯导数据接收任务
RxMotor	27.54	23.97	24.87	80	发动机接收任务
RxDI	24.39	21.57	22.47	50	开关量接收任务

统计仿真过程中系统的 CPU 利用率，测试结果如图 8 所示。由测试结果可知，系统 CPU 利用率在 55% 至 60% 之间，仿真过程中 CPU 利用率分布比较均衡，没有出现大的峰值，说明 CAN 总线收发任务调度均衡，没有出现任务扎堆运行造成 CPU 利用率瞬间增大的情况。

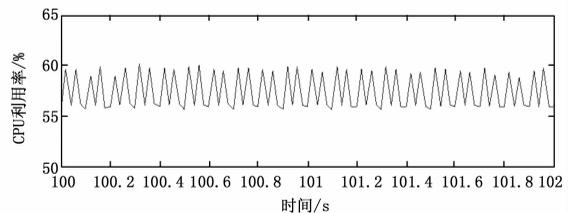
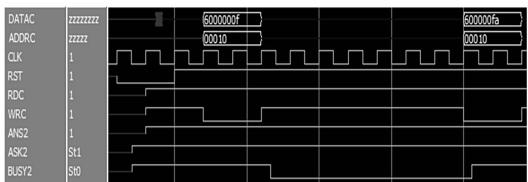


图 8 CPU 利用率测试结果

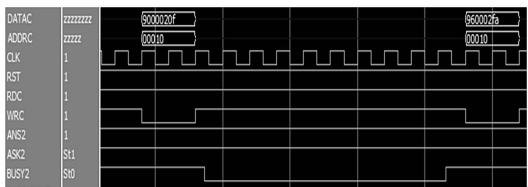
如图 9 所示是 CAN 总线收发任务的调度情况。图 9 中横轴是仿真运行时间，纵轴是收发任务名称，每个旗子符号代表任务执行。图 9 中 Rx FCC _ MINS 是 1553B 接收任务，Rx FCC _ BINS、Rx FCC _ Servo 以及 Rx FCC _ AirBrake 是串口接收任务，可以看出发送任务按照接口协议周期执行，在一个基本周期内任务执行时间不同，根据接口类型分时执行，说明收发任务均匀分布，没有出现发送数据竞争，总线调度机制可以有效的避免数据冲突的情况。

(下转第 186 页)

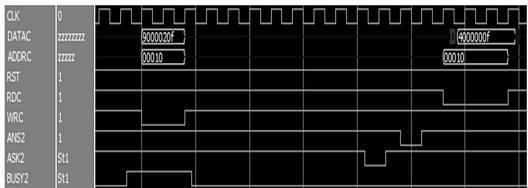
减计数器。时序仿真的结果如图 11 所示。



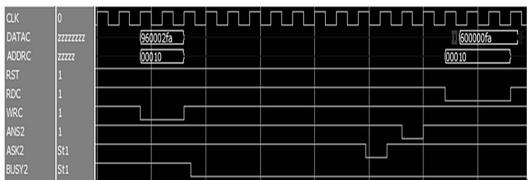
(a) 计数器初始化仿真图



(b) 计数器状态传送仿真图



(c) 计数器OFH输出仿真图



(d) 计数器FAH输出仿真图

图 11 计数器时序仿真结果

由仿真测试可知，中央控制器通过状态传送指令告知计数器的驱动的通断电，由计数器根据驱动的通断电状态对计数器进行计数，计数器能正常工作。

6 结语

本文以 ARM+FPGA 构建了 PLC 系统，对此 PLC 系统的 FPGA 构建的定时器/计数器进行了介绍。然后对设计所使用的软件平台、硬件平台和编程语言进行了说明。同时对定时和计数控制模块的总体结构进行了设计，重点对端口读写控制器及中央控制器与定时和计数控制模块的时序进行了设计，设计了端口读写控制器的内部结构以及其读写控制的功能，设计的定时器/计数器具有可重构的功能，可以根据需要来增减定时器/计数器的数量和种类。

参考文献：

- [1] 王成福. 可程序控制器及其应用 [M]. 北京: 机械工业出版社, 2007.
- [2] 孙培燕, 李克俭, 蔡启仲, 等. FPGA 数据总线宽度不相等的双口 RAM 的设计 [J]. 科学技术与工程, 2014, 35: 249-253.
- [3] 李 静, 蔡启仲, 蒋玉新, 等. 基于 FPGA 的多操作位逻辑运算控制器的设计 [J]. 测控技术, 2015 (2): 81-84.
- [4] 李 静, 蔡启仲, 张 炜, 等. 基于 FPGA 的 PLC 输入存储与读取控制器的设计 [J]. 仪表技术与传感器, 2014 (6): 33-36.
- [5] 周曙光, 李克俭, 蔡启仲, 等. 基于 FPGA 的 PLC 位信息输出控制器设计 [J]. 计算机测量与控制, 2014 (6): 1750-1753.
- [6] 谷 鑫, 徐贵力, 王友仁. FPGA 动态可重构理论及其研究进展 [J]. 计算机测量与控制, 2007, 11: 1415-S1418.
- [7] 夏宇闻. Verilog 数字系统设计教程 [M] (第 2 版). 北京: 北京航空航天大学出版社, 2008.
- [8] 王金明. 数字系统设计与 Verilog HDL [M] (第 3 版). 北京: 电子工业出版社, 2009.

(上接第 174 页)

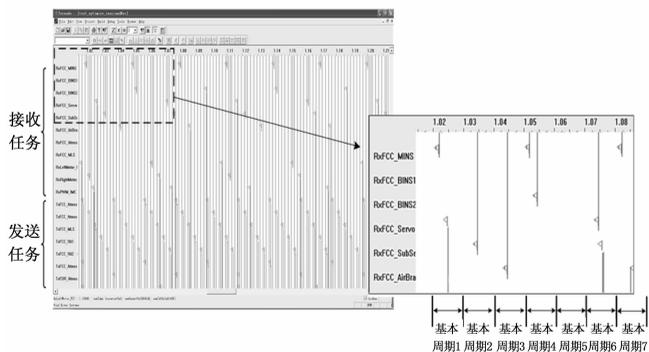


图 9 CAN 总线收发任务调度情况

5 结语

本文以基于 CAN 总线的某样例无人机飞行仿真系统为硬件平台，设计了基于主节点时间的主从静态分时调度机制、接口资源注册机制以及状态检测机制。最终实现 CPU 单元对资源接口的统一调度，灵活地对接口资源的增减进行管理。同时通过文中的分析和试验，表明总线时间调度能有效的调度数据收发，避免数据冲突；多板卡能够稳定、可靠地协同工作，完成对仿真对象的仿真模拟，整个仿真系统可以为无人机飞行仿真测试提供正确可靠的验证平台。

参考文献：

- [1] 李 谦. 基于 PC104 + VxWorks 的无人机仿真系统设计、实现 [D]. 西安: 西北工业大学, 2007.
- [2] 伍智峰. 分布式飞行仿真技术研究 [D]. 西安: 西北工业大学, 2003.
- [3] 冯 磊. 基于 RTW 的分布式实时仿真系统研究与实现 [D]. 长沙: 国防科技大学, 2005.
- [4] 向孝龙, 陈 欣, 等. 基于 RTW 和 VxWorks 的无人机仿真系统设计及实现 [J]. 计算机测量与控制, 2014, 22 (4): 1291-1293
- [5] 张宏伟, 王树新, 等. 基于 CAN 总线的自治水下机器人控制系统 [J]. 机器人, 2006, 28 (4): 248-252
- [6] 郑 雷. 无人机冗余飞行控制计算机设计及研究 [D]. 南京: 南京航空航天大学, 2011.
- [7] 余 伟, 郭 斌, 等. 嵌入式多板卡协同工作及热备份系统研究 [J]. 信息安全与通信保密, 2013, 5: 102-104
- [8] 米 毅. 基于 CAN 总线的小型飞行控制系统硬件设计与实现 [D]. 南京: 南京航空航天大学, 2009.
- [9] 罗 峰, 孙泽昌. 汽车 CAN 总线系统原理、设计与应用 [M]. 北京: 电子工业出版社, 2011.
- [10] 张颖超, 杨宇峰, 等. 基于 CAN 总线的温室监测系统的通信设计 [J]. 控制工程, 2009, 16 (1): 103-106
- [11] 曹桂平, 等. VxWorks 设备驱动开发详解 [M]. 北京: 电子工业出版社, 2011.