

# AT89C52 单片机与 ARINC429 航空总线接口设计

赵世强, 赵满, 周义健  
(空军工程大学, 陕西 西安 710077)

**摘要:** DEII016 是一种可支持 ARINC429 航空总线协议的串行接收、发送器件。文中介绍了 DEII016 的主要功能和工作原理, 给出了 DEII016 的数据收发过程, 同时重点介绍了 DEII016 与 AT89C52 单片机接口电路的设计方法。

**关键词:** DEII016; 接收; 发送; 接口设计; AT89C52

**分类号:** TP334.7

**文献标识码:** B

**文章编号:** 1006-6977(2004)08-0025-03

## The Avigation Bus Interface Design Between AT89C52 and ARINC429

ZHAO Shi-qiang, ZHAO Man, ZHOU Yi-jian

**Abstract:** DEII016 is a serial transceiver which can support ARINC429 avigation bus protocol. The function and the principle of DEII016 are given. The data transferring process and the interface circuit design are mainly introduced in this paper.

**Keywords:** DEII016; interface design; transfer; receive; AT89C52

### 1 DEII016 的主要功能

DEII016 是 Device Engineering Incorporated 生产的一种可支持 ARINC429 航空总线协议的串行接收、发送器件。该器件的主要功能如下:

串行数据字长为 32 或 25 位;

串行数据速率为 100kbps 或 12.5kbps;

具有独立的收、发电路;

两个独立的接收器可直接与 ARINC429 总线接口;

具有串/并接收数据转换, 和并/串发送数据转换功能;

对接收的数据可进行奇偶状态校验; 并可对发送数据奇偶状态进行设置;

可自动产生字与字之间的间隔;

采用单一 +5V 供电。

### 2 DEII016 的工作原理

DEII016 的引脚功能及特点以前已有介绍 [1], 现仅对该器件的工作原理作以描述。

#### 2.1 DEII016 的复位及初始化

电路上电后, 首先在 MR 引脚施加 200 $\mu$ s 的低电平脉冲以使 DEII016 复位, 然后把控制

字写到 16 位数据总线上, 这样, 在 LDCW 引脚施加低电平便可把控制字写入到控制字寄存器, 尔后便可接收和发送数据。

#### 2.2 DEII016 的数据发送过程

DEII016 的数据发送时序如图 1 所示。发送数据时, 首先应查询 TXR 引脚是否为高电平, 若是高电平, 表明发送存储器已空; 此时置 ENTX 引脚为低电平可停止发送数据, 并把要发送的数据写到 16 位数据总线上。之后, 若要发送数据字 1, 则置 LD1 为低, LD2 为高; 若发送数据字 2, 则置 LD1 为高, LD2 为低, 此时再查询 TXR 引脚是否为低电平, 若为低电平, 表明发送数据已装载到存储器中, 此时再置 ENTX 引脚为高电平即可发送数据。DEII016 的发送存储器最多可装载 8 个 32 位字长的数据, DEII016 采用的是先进先出的发送顺序。

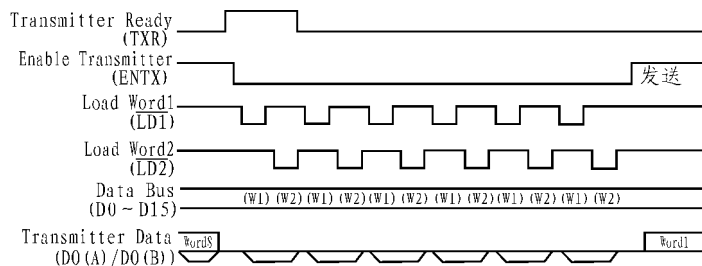


图 1 DEII016 的数据发送时序

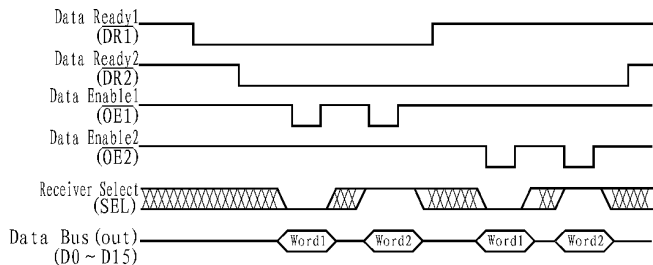


图 2 DEII016 的数据接收时序

### 2.3 DEII016 的数据接收过程

DEII016 的数据接收时序如图 2 所示。当接收器 1 接收到数据后,系统会置 DR1 为低电平;而当接收器 2 接收到数据后,系统将置 DR2 为低电平。欲使接收器 1 将接收数据的字 1 送至 16 位数据总线上,应先置 SEL 为低电平,再置 OE1 为低电平且置 OE2 为高。而后置 SEL 为高电平,便可将字 2 送至 16 位数据总线。要使接收器 2 的数据送至 16 位数据总线上,则置 OE2 为低电平,OE1 为高电平即可。

## 3 AT89C52 与 DEII016 的接口设计

AT89C52(U1) 8 位微控制器与 DEII016 的 16 位数据总线进行数据交换时,必须先利用锁存器来锁存数据。其接口电路框图见图 3。发送数据时,U1 先把要发送的数据分成低 8 位和高 8 位送出,并进入 U3、U4 锁存,尔后再把 16 位数据一同送至 DEII016 (U2) 的数据总线上。U3、U4、U5、U6 均选用 74LS373 芯片,它们的内部均由 D 锁存器和三态输出门组成,其功能表如表 1 所列。当 OE 为低电平,且 G 为高电平时,输出 Q 与输入 D 之间为直通方式。而当 OE 为高电平,且 G 由高电平跳变为低电平后,D 端输入数据被 D 锁存器锁存,之后即使 D 端(P0 口)输入数据再发生变化,也不会影响已锁存的数据。如果 OE 为低电平,系统将输出原先锁存的数据。为了锁存低 8 位数据,需要由 U1 的 WR 和地址选通信号 Y7(由译码器 U7 译出)经或非门 U8 去控制 U3 的 G 端。实际上,只有 U1 在执行写操作时,WR 和 Y7 才同时为低电平,而此时低 8 位数据才能被 U3 暂时锁存。高 8 位数据的锁存与此类似

表 1 74LS373 功能表

输出控制 OE	使能 G	输入 D	输出 Q
L	H	H	H
L	H	L	L
L	L	X	Q <sub>0</sub>
H	X	X	Z

M, U1 的 P3.4 由高变低可打开 U3、U4 的输出门,从而使 U3、U4 把数据送至 U2 的数据总线,当 U1 的 P1.3 或 P1.4 由高变低以把 16 位数据装入发送存储器后,P3.4 会由低变高,此时 U3、U4 输出为高阻抗状态,但不会影响 U2 的数据输出状态。其发送程序如下:

```

mov  dptr, # e000h
mov  a, # d11    ;取字 1 的低 8 位
movx @dptr,a    ;把字 1 的低 8 位锁入 U3
mov  dptr, # c000h
mov  a, # d1h    ;取字 1 的高 8 位
movx @dptr,a    ;把字 1 的高 8 位锁入 U4
clr  p3.4       ;打开 U3、U4 的输出
nop
clr  p1.3       ;字 1 数据装入 DEII016
nop
setb p3.4      ;关闭 U3、U4 的输出
setb p1.3
mov  dptr, # e000h
mov  a, # d21    ;取字 2 的低 8 位
movx @dptr,a    ;把字 2 的低 8 位锁入 U3

```

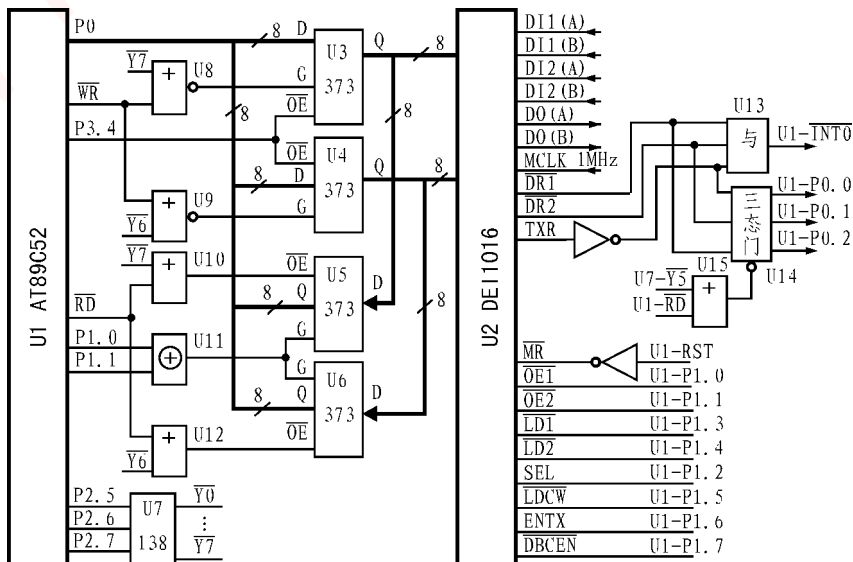


图 3 AT89C52 与 DEII016 的接口电路

```

mov  dptr, # c000h
mov  a, # d2h      ;取字 2 的高 8 位
movx @dptr, a     ;把字 2 的高 8 位锁入 U3
clr  p3.4         ;打开 U3、U4 的输出
nop
clr  p1.4         ;字 2 数据装入 DEH1016
nop
setb p3.4
setb p1.4

```

用 U1 接收数据,应将 DR1、DR2 及 TXR 经与门 U13 接至 U1 的 INT0,在 INT0 口产生中断后,U1 便执行读操作,并通过 RD 和 Y5 控制 U14、U15 以读入 DR1、DR2 及 TXR 的状态。U2 接收到数据或发送数据完毕后,即向 U1 的 INT0 口申请中断,并在中断程序中判断 DR1、DR2 及 TXR 的状态。TXR 为低表明发送完毕,而 DR1 或 DR2 为低则表明已收到数据。此时可置 P1.2(SEL) 为低电平,同时也置 P1.0(接 DEH1016 的 OE1) 或 P1.1 (接 DEH1016 的 OE2) 为低电平,以使字 1 的数据出现在数据总线上,而使 P1.0、P1.1 经 U11 异或后输出以使 U5、U6 的 G 端为高电平,从而把数据锁入 U5、U6。由于 RD 分别与 Y7、Y6 相或后的输出是与 U5、U6 的 OE 门相连接的,因此,U1 执行读操作时,应先后打开 U5、U6 的 OE 门来把数据读入。其接收程序如下:(以接收器 1 为例)。

```

clr  p1.2        ;置 DEH1016 的 SEL = 0,读字 1
clr  p1.0        ;置 DEH1016 的/OE1 = 0
setb p1.1       ;置 DEH1016 的/OE2 = 1
nop
mov  dptr, # e000h
movx a, @dptr   ;把字 1 的低 8 位读入
mov  10h, a

```

```

mov  dptr, # c000h
movx a, @dptr   ;把字 1 的高 8 位读入
mov  11h, a
setb p1.2      ;置 DEH1016 的 SEL = 1,读字 2
nop
mov  dptr, # e000h
movx a, @dptr   ;把字 2 的低 8 位读入
mov  12h, a
mov  dptr, # c000h
movx a, @dptr   ;把字 2 的高 8 位读入
mov  13h, a
setb p1.0      ;置 DEH1016 的/OE1 = 1
setb p1.1      ;置 DEH1016 的/OE2 = 1
setb p1.2      ;置 DEH1016 的 SEL = 1

```

DEH1016 的两路接收可能产生同时接收的现象。即在上述两路共用一个中断源时,如果某一路申请中断已进入接收程序,那么另一路同时接收的数据就可能被屏蔽。解决方法是:进入中断程序后,在接收数据过程中再查询另一路 DR 的状态。当然,如果每路单独使用一个中断源,效果将会更好。

## 4 结束语

实际上,上述 AT89C52 与 DEH1016 的接口电路经改进也可以应用到其它 16 位数据总线接口的电路中,而如果用 ISP 器件替代锁存器及其门电路,还可大大减小电路板的面积。

### 参考文献

- [1] 范秀峰,周宇晨. ARINC429 总线收发器芯片 DEH1016 的原理及应用[J]. 国外电子元器件,

# 嵌入式资源免费下载

## 总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)
19. [UART16C554 的设计](#)
20. [基于 VPX 的高性能计算机设计](#)
21. [基于 CAN 总线技术的嵌入式网关设计](#)
22. [Visual C 串行通讯控件使用方法与技巧的研究](#)
23. [IEEE1588 精密时钟同步关键技术研究](#)
24. [GPS 信号发生器射频模块的一种实现方案](#)
25. [基于 CPCI 接口的视频采集卡的设计](#)
26. [基于 VPX 的 3U 信号处理平台的设计](#)
27. [基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计](#)

## VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)

2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)
21. [基于 VxBus 的驱动程序架构分析](#)

## Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 CC++ 语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)

17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)
18. [Android 简单 mp3 播放器源码](#)
19. [嵌入式 Linux 系统实时性的研究](#)
20. [Android 嵌入式系统架构及内核浅析](#)
21. [基于嵌入式 Linux 操作系统内核实时性的改进方法研究](#)
22. [Linux TCP IP 协议详解](#)
23. [Linux 桌面环境下内存去重技术的研究与实现](#)

## Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)

## PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)

8. [基于 MPC8247 嵌入式电力交换系统的设计与实现](#)
9. [基于设备树的 MPC8247 嵌入式 Linux 系统开发](#)
10. [基于 MPC8313E 嵌入式系统 UBoot 的移植](#)

## ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)
13. [基于 ARM9 的嵌入式 Zigbee 网关设计与实现](#)
14. [基于 S3C6410 处理器的嵌入式 Linux 系统移植](#)

## Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COM Express Nano 工控板实现 IP 调度设备](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)
9. [基于 COM Express 的信号系统逻辑运算单元设计](#)
10. [基于 COM Express 的回波预处理模块设计](#)
11. [基于 X86 平台的简单多任务内核的分析与实现](#)
12. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)

Created in Master PDF Editor