

CANopen 总线技术及应用

王有庆

OEM 客户部纺机应用中心

021-62848800-206

youqing.wang@cn.schneider-electric.com

一、什么是 CAN ?

CAN, 全称为“Controller Area Network”, 即控制器局域网, 是国际上应用最广泛的现场总线之一。CAN 最初出现在 80 年代末的汽车工业中, 由德国 Bosch 公司最先提出。CAN 作为一种多主方式的串行通讯总线, 具有十分优越的特性:

- 采用短帧通讯, 实时性强, 抗干扰能力强
- 很远的数据传输距离(长达 10Km), 高速的数据传输速率(高达 1Mbit/s)
- 可靠的错误处理和检错机制
- 发送的信息遭到破坏后, 可自动重发
- 节点在错误严重的情况下具有自动退出总线的功能
- 可根据报文的 ID 决定接收或屏蔽该报文, 报文不包含源地址或目标地址, 仅用标志符来指示功能信息、优先级信息
- 软硬件标准完全开放, CAN 在软硬件成本上具有很大的优势

二、什么是 CANopen ?

CAN 主要描述设备之间的信息传递方式, 定义了开放系统互连模型(OSI)的最下面两层: 数据链路层和物理层。为了能够把 CAN 推广到更多的领域, 主要由欧洲公司发起的 CiA 组织(CAN in Automation)在 1995 年发表了完整版的 CANopen 通信子协议。目前, CANopen 已成为全欧洲最重要的现场总线标准之一。

CANopen 网络采用总线拓扑结构, 可以管理 128 个设备: 1 个主站和 127 个从站。总线的通讯速率主要取决于总线的长度和硬件驱动能力, 两者的关系可参考表 1。总线的 2 个终端需要安装 120Ω 的终端电阻。

表 1 CANopen 通讯距离与距离的关系

通讯速率 (kbits/s)		10	20	50	125	250	500	1000
最大距离 (m)	TSXCPP110	1000	1000	1000	500	250	100	25
	ATV31	5000	2500	1000	500	250	100	5

CANopen 不仅定义了 OSI 的应用层和通信子协议, 而且为可编程系统、不同器件、接口、应用子协议定义了大量的行规(CiA DS-301 和 CiA DSP302), CANopen 通信行规既支持对设备参数的直接存取, 也支持对

时间苛求的过程数据通信，遵循这些行规开发出的 CANopen 设备将能够实现不同公司产品间的互连。

CANopen 的设备接口和应用行规(CiA DS-4XX)定义了四种标准化的通信对象，通过通信对象标识符 (COB-ID) 来表示区分。

1. 过程数据对象 PDO(Process Data Objects)

主要用于实时性要求较高的过程数据通信。

PDO 被映射到单一的 CAN 帧中，使用所有的 8 个字节的数据域来传输应用对象，采用生产者/消费者 (Producer/Consumer) 的通信模式，每个 PDO 有一个独立的标识符并且可能只被一个节点发送，但它可以被多于一个节点接收，且接收者不需作出应答确认。

PDO 可以通过多种通信方式传送：周期性传送，事件触发，外部时钟，远程帧请求以及从特定节点接收到同步报文都可以启动 PDO 的发送。

2. 服务数据对象 SDO(Service Data Objects)

主要用于对实时性要求不高且数据量较大的配置参数的通信。

SDO 可以传输大于 8 个字节的配置信息，允许传送任意长度的对象，采用客户机/服务器 (Client/Server) 的通信模式，接收者将确认收到的每个段信息，发送和接收者间将建立点对点的通信。

3. 其它对象

包括主要用于对总线管理服务的网络管理对象 NMT 和用于总线的同步及故障处理的特殊功能对象 SFO 等。

CANopen 作为一种**技术先进、可靠性高、实时性强，功能完善、成本合理、完全开放**的现场总线，目前已在工业控制系统，汽车，公共交通运输系统，医疗设备，海运电子设备和建筑自动化系统中取得了广泛的应用，是将 CAN 应用推向深化的理想选择。

三、施耐德电气的 CANopen 方案应用

施耐德电气的 CANopen 解决方案，从 PLC (Twido、Micro、Premium) 和变频器、到远程 I/O(Advantys)等均提供 CANopen 总线模块，ATV31 和 ATV71 变频器则已内置了 CANopen 总线接口。

通过 PDO、SDO 等通信对象，CANopen 的应用可以非常简单地实现。如对实时性要求较高的数据，可以通过配置过程数据对象(PDO)，实现 PDO 数据与主站内存的映射，从而主站对从站的控制相当于对自身内存的读写操作；对实时性要求不高或数据量较大的数据，可以通过在主站内编程访问从站的服务数据对象(SDO)实现。

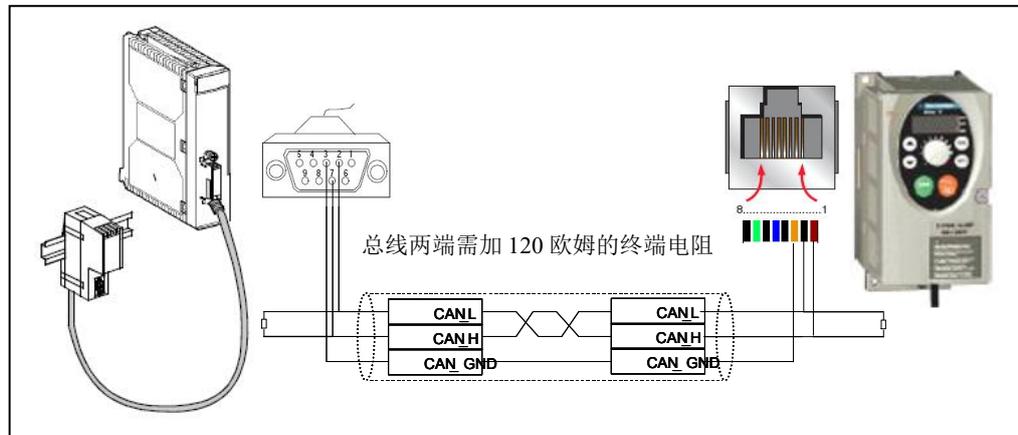
本文以 Micro PLC(版本号 \geq V4.4，加装 TSXCPP110 CANopen 通讯扩展卡)及 ATV31(内置 CANopen 接口)为例介绍 CANopen 方案的应用。

1. 硬件连接(见图 1)

采用屏蔽双绞线按图 1 所示，将 CANopen 接口的 3 根信号线对应互连成 CANopen 总线，并在总线的两端加上 120 欧姆的终端电阻。

需要注意的是，Micro PLC 的硬件版本号必须 \geq V6.0，低于这个版本时，可以采用相应补丁文件(FIRMWARE)将其升级至 V6.0。

图 1 TSX CPP110 CANopen 主站卡与 ATV31 的连接



2. 软件配置及编程

软件配置及编程需要：

- a. PL7 编程软件（版本号 \geq V4.4）
- b. Canopen 配置软件，如 SyCon(版本号 \geq 2.8)
- c. 相关 EDS 文件(Electronic Data Sheet，电子数据表)

(1) PDO 配置步骤

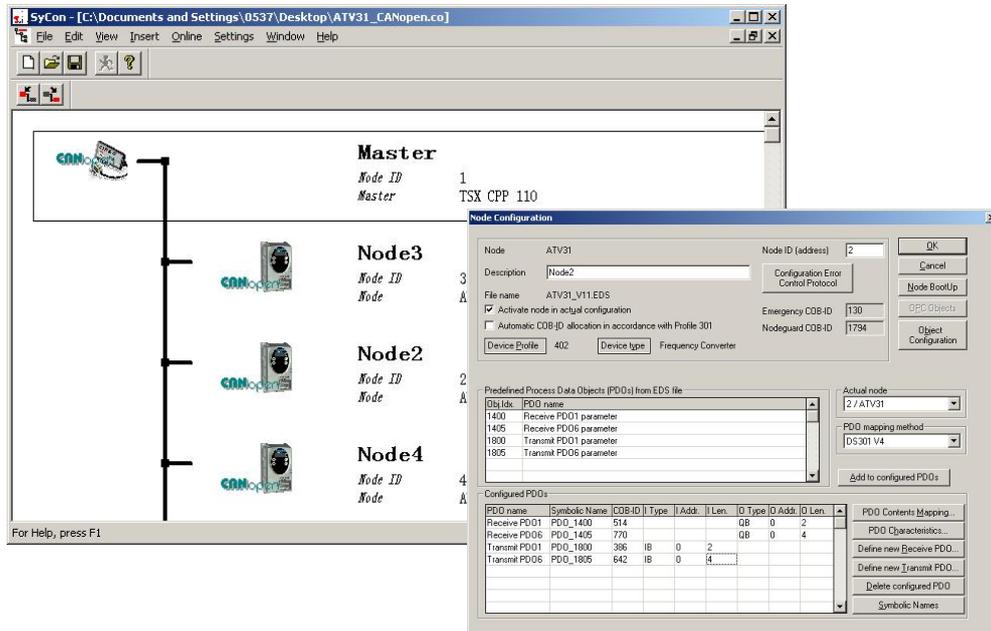
- a. SyCon 配置软件配置主站和从站 PDO，见图 2 所示。

ATV31 支持两种预设的 PDO(详见《ATV31 CANopen 用户手册》)：

PDO1: 一个接收的 PDO 字(如控制字 CMDD)和一个发送的 PDO 字(如状态字 ETAD)

PDO6: 两个接收的 PDO 字和两个发送的 PDO 字 (可根据需要设置不同的参数)

图 2 SyCon 配置 PDO



需要注意的是，PDO1 的通信对象标识符 COB-ID 是自动配置的，其中，Receive PDO1: 512+节点号；Transmit PDO1: 384+节点号而 PDO6 的 COB-ID 必须手工输入，方法是：

- i. 先取消“自动配置 COB-ID”选项
- ii. 在“配置 PDO”的 COB-ID 栏中手工输入值，其中 Receive PDO1: 768+节点号；Transmit PDO1: 640+节点号
如对节点 3，Receive PDO1: 771；Transmit PDO1: 643

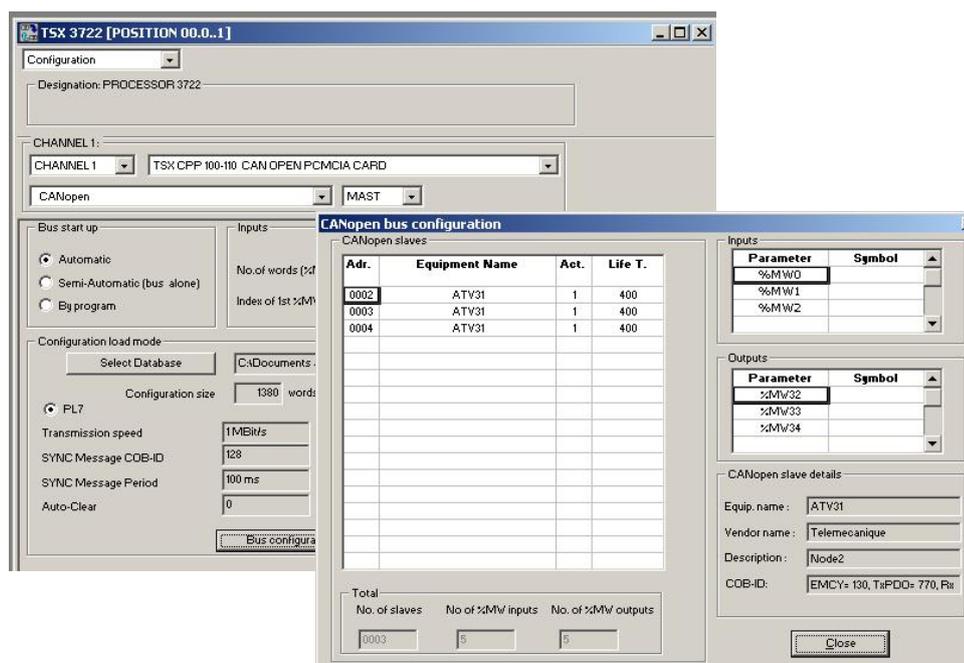
PDO 配置完成后，存盘为扩展名为*.co 的 PDO 配置数据文件。

b. PL7 CANopen 配置及编程，见图 3 所示

在 PL7 编程软件的硬件配置中，将通讯扩展口配置为 TSXCPP110 CANopen 主站卡，单击“Select Database”项下传 PDO 配置数据文件，然后单击“Bus Configuration”项可以看到所配置 PDO 数据与 PLC 内存的映射。通过 PLC 自身相应内存的读写操作，即可实现主站对从站的控制，无需编写特定的通讯程序。通过 PL7 编程软件在联机状态下，可以在 CANopen 主站卡相应的调试窗口中对 PDO 和 SDO 进行监测和调试。

需要注意的是，通过 PL7 编程软件下传 PDO 配置数据文件时，需要占据较大容量的内存空间，因此在从站数量较多的情况下，必须额外订购内存扩展卡，以保证足够的编程空间。

图 3 PL7 组态 CANopen 主站卡



(2) SDO 通讯编程

SDO 的通讯需在 PL7 软件中应用 READ_VAR 及 WRITE_VAR 函数编程来实现，详见 PLC 编程软件帮助。

```
WRITE_VAR(ADR#0.1.SYS,          `SDO',          subindex:index,
NodeID, %MWi:L, %MWk:4)
READ_VAR(ADR#y.1.SYS,          `SDO',          subindex:index,
NodeID, %MWi:L, %MWk:4)
```

例：通过 SDO 设置站号为 2 的 ATV31 变频器的加速时间 ACC 为 100ms(ACC 的分辨率为 0.1ms)，则编程如下：

```
%MW100:=16#203C; (*SDO index, 参见《ATV31 通讯变量手册》*)
%MW101:=16#0002; (*SDO sub index, 参见《ATV31 通讯变量手册》*)
%MW102:=16#03E8; (*1000=>16#03E8*)
%MW113:=2;      (*第 4 个通讯管理字, 发送 SDO 字节的长度*)
WRITE_VAR(ADR#0.1.SYS,'SDO',%MD100,2,%MW102:1,%MW110:4);
```

(3) 其它事项

- 相关的变量应避免在 PDO 或 SDO 中重复映射和读写，如 ATV31 的控制字 CMDDD 在 PDO1 中已映射过，如再次在 PDO6 中映射，则可能导致 ATV31 的控制时序的紊乱。
- 在 POD 中映射过的参数，可以用 SDO 读，但不能写。
- 如程序中有多个 SDO 需进行通讯，应将它们的读写时序错开。