

# 基于 CAN 总线的 CANopen 协议讲座（六）

## CANopen 从站模块（TinyARM）

AN01010101 V1.00 Date: 2009/11/14

产品应用笔记

类别	内容
关键词	CANopen CAN 总线 CAN-bus 现场总线 总线协议
摘要	CANopen 协议以其成熟的结构、抗干扰能力强等优势越来越多的应用到各种工业活动中，也成为各种行业的标准，已经成为未来现场总线发展的一种趋势。

修订历史

版本	日期	原因
V1.00	2009/11/14	创建文档

## 销售与服务网络 (一)

### 广州周立功单片机发展有限公司

地址: 广州市天河北路 689 号光大银行大厦 12 楼 F4  
邮编: 510630  
电话: (020)38730916 38730917 38730972 38730976 38730977  
传真: (020)38730925  
网址: [www.zlgmcu.com](http://www.zlgmcu.com)



### 广州专卖店

地址: 广州市天河区新赛格电子城 203-204 室  
电话: (020)87578634 87569917  
传真: (020)87578842

### 南京周立功

地址: 南京市珠江路 280 号珠江大厦 2006 室  
电话: (025)83613221 83613271 83603500  
传真: (025)83613271

### 北京周立功

地址: 北京市海淀区知春路 113 号银网中心 A 座  
1207-1208 室 (中发电子市场斜对面)  
电话: (010)62536178 62536179 82628073  
传真: (010)82614433

### 重庆周立功

地址: 重庆市石桥铺科园一路二号大西洋国际大厦  
(赛格电子市场) 1611 室  
电话: (023)68796438 68796439  
传真: (023)68796439

### 杭州周立功

地址: 杭州市天目山路 217 号江南电子大厦 502 室  
电话: (0571) 28139611 28139612 28139613  
28139615 28139616 28139618  
传真: (0571) 28139621

### 成都周立功

地址: 成都市一环路南二段 1 号数码同人港 401 室  
(磨子桥立交西北角)  
电话: (028)85439836 85437446  
传真: (028)85437896

### 深圳周立功

地址: 深圳市深南中路 2070 号电子科技大厦 C 座 4  
楼 D 室  
电话: (0755)83781788 (5 线)  
传真: (0755)83793285

### 武汉周立功

地址: 武汉市洪山区广埠屯珞瑜路 158 号 12128 室  
(华中电脑数码市场)  
电话: (027)87168497 87168297 87168397  
传真: (027)87163755

### 上海周立功

地址: 上海市北京东路 668 号科技京城东座 7E 室  
电话: (021)53083452 53083453 53083496  
传真: (021)53083491

### 西安办事处

地址: 西安市长安北路 54 号太平洋大厦 1201 室  
电话: (029)87881296 83063000 87881295  
传真: (029)87880865

## 销售与服务网络 (二)

### 广州致远电子有限公司

地址: 广州市天河区车陂路黄洲工业区 3 栋 2 楼

邮编: 510660

传真: (020)38601859

网址: [www.embedtools.com](http://www.embedtools.com) (嵌入式系统事业部)

[www.embedcontrol.com](http://www.embedcontrol.com) (工控网络事业部)

[www.ecardsys.com](http://www.ecardsys.com) (楼宇自动化事业部)



### 技术支持:

#### CAN-bus:

电话: (020)22644381 22644382 22644253

邮箱: [can.support@embedcontrol.com](mailto:can.support@embedcontrol.com)

#### iCAN 及数据采集:

电话: (020)28872344 22644373

邮箱: [ican@embedcontrol.com](mailto:ican@embedcontrol.com)

#### MiniARM:

电话: (020)28872684 28267813

邮箱: [miniarm.support@embedtools.com](mailto:miniarm.support@embedtools.com)

#### 以太网:

电话: (020)22644380 22644385

邮箱: [ethernet.support@embedcontrol.com](mailto:ethernet.support@embedcontrol.com)

#### 无线通讯:

电话: (020) 22644386

邮箱: [wireless@embedcontrol.com](mailto:wireless@embedcontrol.com)

#### 串行通讯:

电话: (020)28267800 22644385

邮箱: [serial@embedcontrol.com](mailto:serial@embedcontrol.com)

#### 编程器:

电话: (020)22644371

邮箱: [programmer@embedtools.com](mailto:programmer@embedtools.com)

#### 分析仪器:

电话: (020)22644375 28872624 28872345

邮箱: [tools@embedtools.com](mailto:tools@embedtools.com)

#### ARM 嵌入式系统:

电话: (020)28872347 28872377 22644383 22644384

邮箱: [arm.support@zlgmcu.com](mailto:arm.support@zlgmcu.com)

#### 楼宇自动化:

电话: (020)22644376 22644389 28267806

邮箱: [mjs.support@ecardsys.com](mailto:mjs.support@ecardsys.com)

[mifare.support@zlgmcu.com](mailto:mifare.support@zlgmcu.com)

### 销售:

电话: (020)22644249 22644399 22644372 22644261 28872524

28872342 28872349 28872569 28872573 38601786

### 维修:

电话: (020)22644245

## 目 录

1. 引言.....	1
2. TinyARM简介 .....	1
3. TinyARM CANopen协议栈配置.....	1
4. CANopen协议栈硬件资源配置 .....	2
5. 设计对象字典.....	2
6. TinyARM CANopen API函数编程.....	3
7. 结语.....	4
8. 参数资料.....	4

## 1. 引言

在上期文章中我们了解了一种嵌入式 CANopen 协议转换模块及其使用方法，本期文章我们将介绍一种支持二次开发的 CANopen 从站模块的使用方法。

CANopen 协议作为 CAN 总线的应用层协议，拥有复杂的网络管理 (Network Management)、实时过程数据对象 (Process Data Object)、服务数据对象 (Service Data Object)、预定义连接和特殊功能对象等基本功能，另外 CANopen 协议还定义了指示灯规范、通信安全架构、在线配置、以及 EDS 文件规范、标准设备规范等一系列的协议规范。因此开发基于 CANopen 协议的设备有着非常高的难度以及较长的开发周期。

## 2. TinyARM 简介

TinyARM 模块内置了 CANopen 从站协议，为用户灵活快速设计 CANopen 从站，构建符合国际标准的工业现场总线及分布式网络提供了更多更好的选择。

TinyARM CANopen 从站模块提供了网络管理 (NMT)、实时数据发送 (TPDO)、实时数据接收 (RPDO)、服务数据对象 (SDO/Server)、同步报文 (SYNC)、紧急事件 (EMCY) 等支持，同时还能够根据需求支持不同的协议描述规范。使用 TinyARM CANopen 从站模块可快速开发基于 CANopen 协议的控制设备、数据采集设备等，配合 CANopen 系列主站设备组建 CANopen 网络，迅速搭建工业现场总线分布式监控系统。



图 1 TinyARM 外观图

## 3. TinyARM CANopen 协议栈配置

TinyARM 是一款可二次开发的嵌入式模块，提供 API 函数库。CANopen 协议栈也是以 API 函数库形式提供，使开发者在开发 CANopen 设备的时候使用起来非常方便。由于 CANopen 协议栈会使用到相应的硬件资源，因此开发者首先需要配置 CANopen 协议栈硬件资源，例如 CANopen RUN 和 ERROR 指示灯所对应的 I/O 端口等，正确的硬件配置可使协议栈稳定可靠的运行。

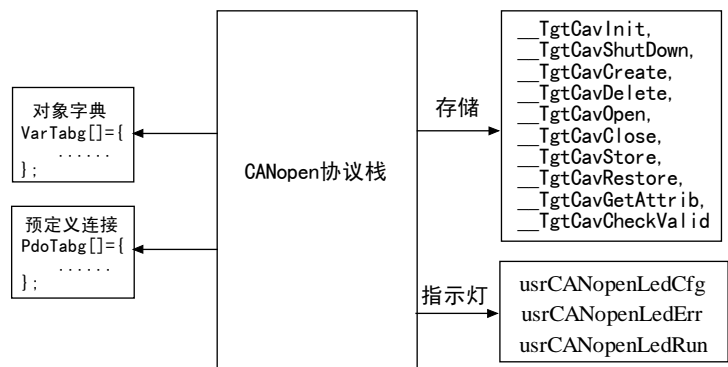


图 2 CANopen 协议栈与硬件资源结构

## 4. CANopen 协议栈硬件资源配置

### 1. 指示灯端口配置

按照 CANopen 协议 DS303-3 的定义，CANopen 从站需要使用两种颜色的 LED 指示灯来指示运行和错误状态，运行指示灯使用绿色，错误指示灯使用红色，根据实际硬件来设置 LED 控制端口。协议栈通过调用如表 1 所示的三个函数来实现，配置文件与协议栈结构如图 2 所示。

表 1 状态指示灯函数列表

函数名称	功能描述
usrCANopenLedCfg	指示灯初始化
usrCANopenLedRun	运行 LED，根据参数来判断当前 LED 所处的状态。
usrCANopenLedErr	错误 LED，根据参数来判断当前 LED 所处的状态。

### 2. 参数存储函数

参数存储相对来说更复杂一些，需要根据实际硬件条件来修改相应的实现函数，其中的 \_\_TgtCavInit, TgtCavOpen, \_\_TgtCavClose, \_\_TgtCavDelete, \_\_TgtCavCreate, \_\_TgtCavShutDown 等函数执行对存储器进行初始化或关闭存储器等操作，在协议栈初始化或关闭时被调用。函数 \_\_TgtCavStore, \_\_TgtCavRestore 主要负责参数的存储与读取，CANopen 协议栈的参数存储是以对象字典为最小单位，每存储或恢复一次对象字典的参数都会调用存储或恢复参数函数，存储、恢复函数与协议栈的关系如图 2 所示。

## 5. 设计对象字典

设计 CANopen 的对象字典主要分为以下两个步骤，其与协议栈结构关系如图 2 所示。

### 1. 生成对象字典

对象字典作为 CANopen 的核心，不同的设备描述规范或不同的应用就有作不同的对象字典，因此对于一款能支持二次开发的模块来说，根据具体需求生成相应的对象字典也是必不可少。TinyARM 中的 CANopen 对象字典可以通过 ODBuilder 软件来生成，生成的对象字典为三个标准 C 文件，文件名分别为：obdcfg.h, objdict.c, objdict.h，把工程中的同名文件覆盖即可将 CANopen 的对象字典项目更改为新设定的对象字典项目，CANopen 所支持的设备规范也可随之改变。

### 2. 对象字典的变量定义

对于对象字典索引在 0x1000~0x1FFF 范围内的对象字典，其变量存储地址已经在协议栈中预定义，不需要开发者再次定义。索引在 0x2000~0x9FFF 之间的索引，需要重新映射变量地址，否则对象字典无法正确存储数据。在 TinyARM CANopen 中，为了使开发者方便映射对象字典变量，在文件名为 `usrcopperdef.c` 的文件中定义了一个参数数组 `__VarTabg`，将需要定义的对象字典变量地址添加到该表格中即可，其格式与使用示例如程序清单 1 所示。

程序清单 1 对象字典变量地址映射示例

```
static BYTE ucTestdat1, ucTestdat2;
static DWORD ucTestdat3 = 0;
static tVarParam __VarTabg[] = {
    {kVarValidAll, 0x2000, 0x00, sizeof (BYTE), &ucTestdat1, NULL, NULL},
    {kVarValidAll, 0x2001, 0x00, sizeof (BYTE), &ucTestdat2, NULL, NULL},
    {kVarValidAll, 0x6100, 0x01, sizeof (WORD), &__ucTestdat3, NULL, NULL},
    .....
};
```

### 3. 预定义连接设置

在 CANopen 运行时通常采用 DS301 所建议的预定义连接 COB-ID（通信对象标识），其中预定义连接集包括 PDO、SDO、NMT、SYNC 以及 EMCY 等服务。协议栈在启动时已经将（除 RPDO 和 TPDO 以外）一些服务的 COB-ID 设定为预定义连接，开发者只需要重新设定 RPDO 和 TPDO 的预定义连接即可。同样开发者只需要在 PDO 预定义表格中添加项目即可，实现示例如程序清单 2 所示。

程序清单 2 PDO 预定义连接定义

```
static tPdoParam ROM __PdoTabg[] = {
    /* RPDO 预定义连接 */
    {kPdoValidAll, 0x1400, kPdoTypeRecv, 0x00000200, 254, 0, 0, 1, {0x62000108, 0, 0, 0, 0, 0, 0},
    NULL},
    .....
    {kPdoValidAll, 0x1800, kPdoTypeSend, (0x00000180), 254, 5000, 0, 4, {0x60000108, 0x60000208,
    0x64010110, 0x64020120, 0, 0, 0, 0}, NULL},
    /* TPDO 预定义连接 */
    .....
}
```

## 6. TinyARM CANopen API 函数编程

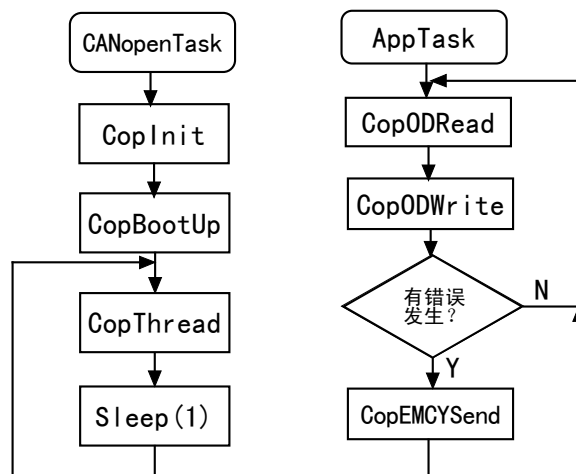


图 3 CANopen 协议栈工作流程

如图 3 所示为 CANopen 协议栈工作流程图，其中 CopInit 为初始化 CANopen 协议栈，CopBootUp 发送 CANopen 启动信息并且进入预操作状态，CopThread 为 CANopen 协议栈线程函数需要应用程序循环调用。开发者可以通过 CopODRead 和 CopODWrite 对对象字典进行读写操作。

**实现实例：**程序清单 3 展示了如何启动 CANopen 协议栈并使之处于运行状态，

程序清单 3 协议栈运行

```

void CANopenTask (void *pdata)
{
    .....
    sprintf(cName, "/CommDevice/CANopenSlv/CommDevice/can1 0x%0x 0x%0x", ucBaudRt, ucNodeID);
    CopInit(0, cName, (void *)&__GCopSlvCbInf);          /* 初始化当前 CANopen 协议栈 */
    CopBootUp();
    while (1) {
        CopThread(NULL);
        OSTimeDly(OS_TICKS_PER_SEC/200);
    }
}
  
```

## 7. 结语

开发者利用 TinyARM CANopen 可完全自主进行二次开发，方便的实现各种标准 CANopen 从站设备。例如，数字量输入输出设备、电机控制设备、数字传感器设备、织布机设备、充电设备等各种设备。使开发者在最短的时间内开发出 CANopen 从站设备，抢占市场先机。

广州致远电子有限公司提供各种接口的 CANopen 接口卡、CANopen 终端设备（包括数据采集，电机控制等）、CANopen 分析仪、网关及网桥、OPC 服务器以及基于 CANopen 协议的整套组网方案及应用开发

## 8. 参数资料

[www.can-cia.com](http://www.can-cia.com)

[www.embedcontrol.com](http://www.embedcontrol.com)