

中国单片机公共实验室

CANopen 工程师应用笔记

CANopen 主节点的设计方案

- 基于 IXXAT 公司 CANopen Master API for Windows

CANopen 主节点除具备 CANopen 设备的基本条件外,还需具备 NMT Master 的功能,即对 CANopen 网络进行管理。对 CANopen 主节点的实现提出三种方案:

- 方案 1: 在 CANopen-Chip 基础上开发 CANopen 主站。
- 方案 2: 通过对 CANopen 协议栈源代码的二次开发在单片机上实现嵌入式 CANopen 主站。
- 方案 3: 利用 CANopen Master API 在 PC 机上实现 CANopen 主节点。

分析比较:

- 方案 1: 通过对 CANopen-Chip 的深入研究,在其 EDS 文件中发现 CANopen-Chip 仅适用于开发从站,其内部没有固化 CANopen 协议栈中的 Master 部分,而 CANopen-Chip 的可开发性较低。因此方案 1 不可行。
- 方案 2: 此方案可以在单片机上实现 CANopen 主站,但需要的开发周期较长。
- 方案 3: 利用 CANopen Master API 可在 windows 下开发 CANopen 主节点,较为方便快捷,同时还有界面可编辑性好、可读性强等优点。

综上,方案 3 最适合本应用系统,因此选用方案 3。

1 CANopen Master API 基础

1.1 CANopen Master API 简介

CANopen API 是在 windows 操作系统下快速开发 CANopen 主站的软件包。该软件包含两个组件:适用于主动型 PC/CAN 板的 CANopen Master Firmware (固件)和以 Windows dll 形式的程序接口(CANopen API)。CANopen Master 固件由完整的 CANopen Master 组成,它可以完全自主处理 CANopen 的通信流程(带超

时监视的发送/接收 SDO, 发送/接收 PDO, 同步产生, 突发处理, 网络管理, 周期节点维护, heartbeat 监视), 这样, PC 只需关注应用程序本身即可。

1.2 CANopen Master API 的实现原理

CANopen Master API 为 CANopen 网络管理和 CANopen 对象通信提供了相关的函数。CANopen Master API DLL 将固件下载至 board 的非易失性存储器中, 并建立用于编辑队列中数据的队列和句柄 (handle)。

CANopen Master 固件可独立运行于 CAN-board (USB-to-CAN) 上。上层与固件通信时使用 2 个命令队列 (CMD-queues), 与网络通信时使用 7 个数据队列 (data queues)。

在 board 上与 CANopen Master 固件相关的命令都通过两个命令队列传送。通信对象和紧急对象的交换则通过 7 个数据队列更快速的实现。见图 1.2.1:

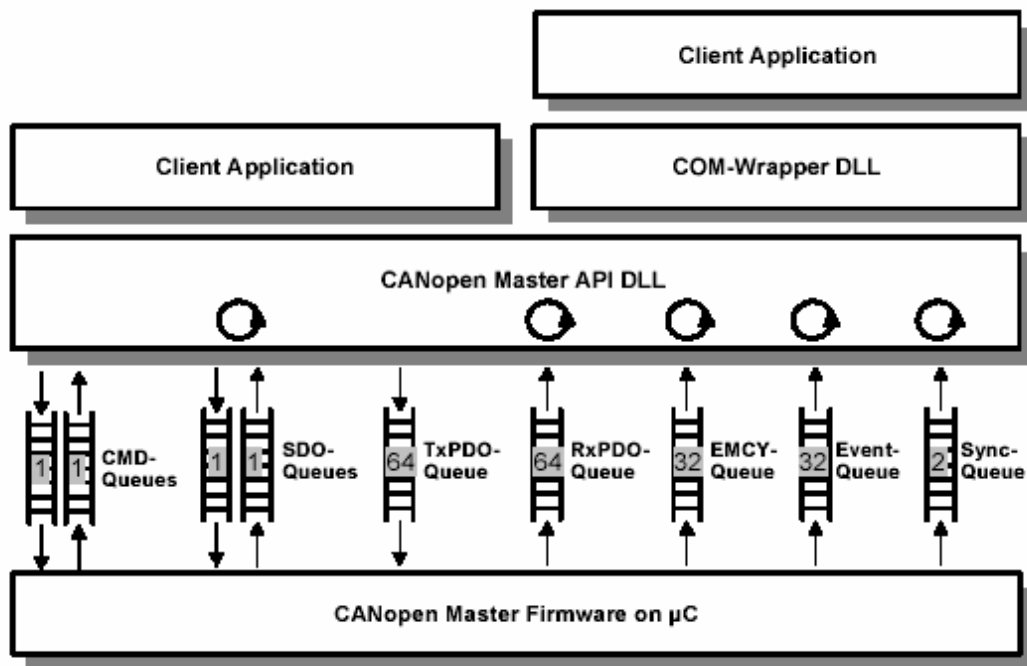


图 1.2.1 Master API DLL 与 Master 固件间的通信

固件经命令队列被配置并参数化为会话状态。CANopen Master API 的所有函数在 DLL 被编辑为操作结构, 然后传送至固件并被处理。

通过两个 SDO 队列可以对外部 CANopen 设备进行配置。可利用 API 函数进行异步 SDO 传输 (非模块化 Windows 应用程序) 及同步 SDO 传输 (模块化 Windows 应用程序)。

Windows 应用程序和 CANopen 网络之间进程数据的交换以 PDO 发送和 PDO 接收队列的方式进行。

另外，还有三个接收队列分别用于 Windows 应用程序对紧急对象、网络事件和同步对象的接收。一个 Callback 或 Windows-message ID 被分配给每隔队列，这样，当收到相应的 CANopen 报文时，客户机应用程序就会直接调用一个函数或者一个 Windows 报文被发送至应用程序窗口。

1.3 CANopen Master API 函数类别

CANopen Master API 的函数可以分为如下类别（标有*号的为主节点程序设计时用到的函数）：

1、基本 API 函数

此类函数用于初始化并参数化 API，选择 CAN-board 及在 CAN-board 上与 Master 固件通信。

● 选择 CAN-board

COP_InitBoard *

COP_ReleaseBoard *

COP_GetBoardInfo *

● 初始化并参数化 API

COP_InitInterface *

COP_DefineCallbacks

COP_DefineEventMsgs

COP_ResetDLL

● 与 Master 固件通信

COP_SendMsg

COP_GetMsg

COP_SetCommTimeOut

COP_GetStatus *

COP_TestCommand

2、网络管理函数

此类函数用于建立网络并控制 CANopen 节点。

- 建立 CANopen 网络

COP_AddNode *

COP_DeleteNode

COP_SearchNode *

COP_ChangeNodeParameter

- 控制 CANopen 节点

COP_StartNode *

COP_StopNode *

COP_ResetComm

COP_ResetNode

COP_EnterPreOperational *

3、CANopen 对象管理

此类函数用于创建并参数化 CANopen 通信对象。

- 进程数据对象 (PDO)

COP_CreatePDO *

- 服务数据对象 (SDO)

COP_CreateSDO *

COP_SetSDOTimeout

- 系统服务

COP_DefSyncObj

COP_EnableSync *

COP_DisableSync *

COP_InitTimeStampObj

COP_StartStopTSObj

4、CANopen 通信

用于独立 CANopen 设备间的直接信息交换。

- 进程数据对象 (PDO)

COP_ReadPDO *

COP_ReadPDO_S

COP_RequestPDO

COP_WritePDO *

COP_WritePDO_S

● 服务数据对象 (SDO)

COP_ReadSDO

COP_WriteSDO

COP_PutSDO

COP_GetSDO

COP_CancelSDO

● 系统服务

COP_GetEmergencyObj *

COP_GetEmergencyObj_S

COP_CheckSync

COP_GetEvent *

5、LMT 服务

COP_LMT_ConfigNode

COP_LMT_GetAddress

2. 主节点程序设计

在进行主节点程序设计时以CANopen从节点为参考，主节点的波特率、节点号、节点守卫参数、PDO的数据位以及各通信对象的COB-ID等均参照从节点的设计。

CANopen Master API for Windows程序设计可使用下列语言（并含有实例）：

1. C++ Builder
2. Delphi
3. VB
4. VC++

以下是我们用VB设计的例子

主节点程序设计时用到的函数已在上节中用*号标出。API函数调用流程图

如图2.1:

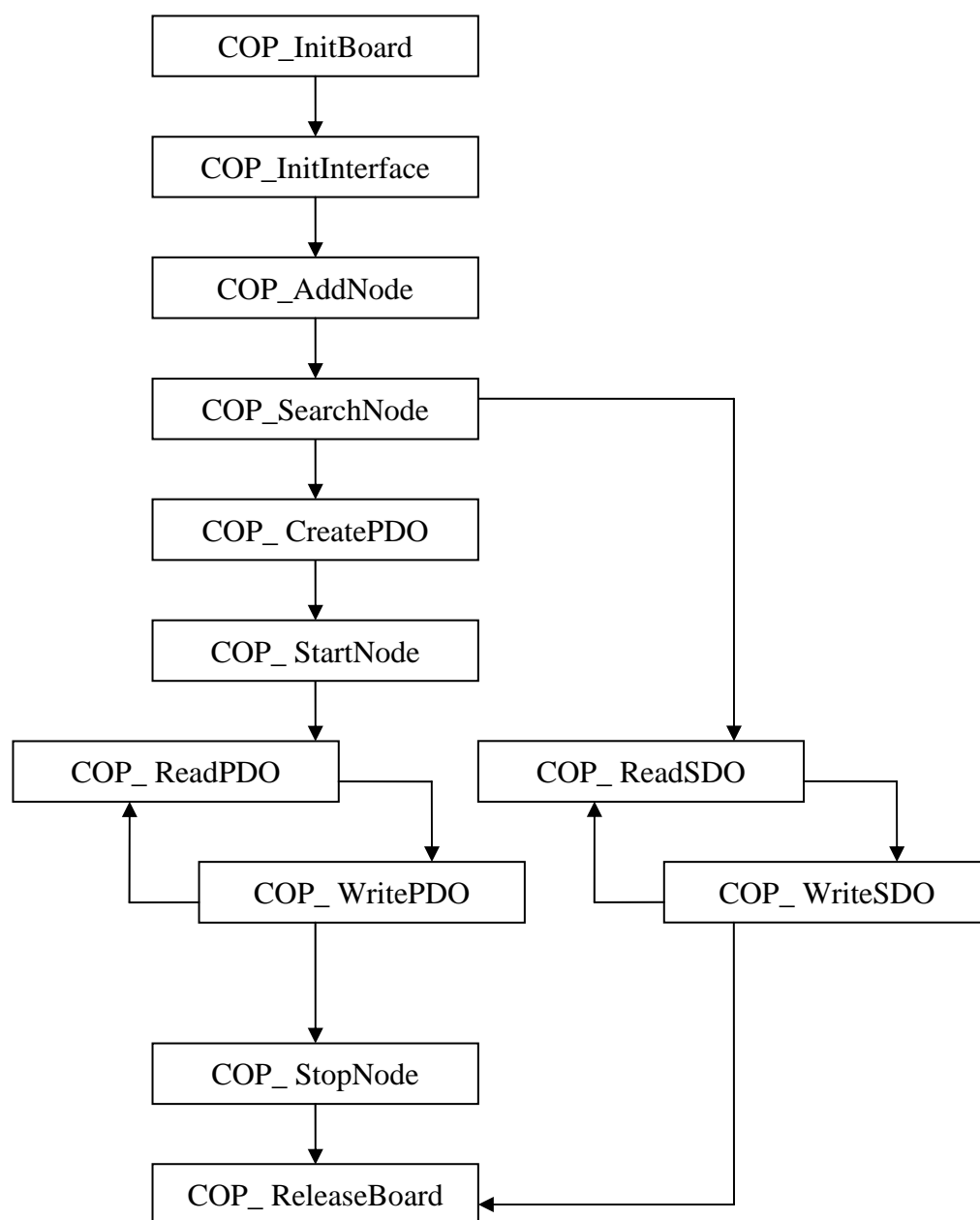


图2.1 API函数调用流程图

中国单片机公共实验室版权所有 2004 年 6 月

<http://www.bol-system.com>

Email: info@bol-system.com

电话: (010) 82357579/80