



CANopen现场总线应用层 协议主站的开发与实现

学生：宋威

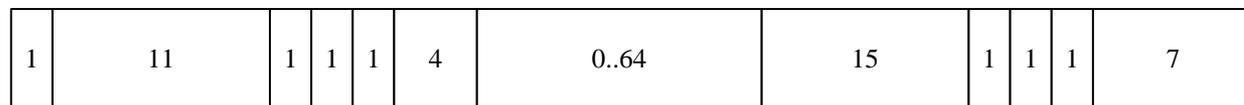
导师：方穗明

2008年5月29日

课题成果概要

- 完成了一个CANopen主站协议栈
 1. 基于标准C语言的非抢占固定优先级任务调度机
 2. 基于散列表扩展的动态可配置的CANopen对象字典
 3. 多任务实时并行可移植CANopen主站协议栈
 4. WindowsXP平台上的CANopen测试系统

CAN总线



报文起始位

CAN报文标识符

控制位：
远程、标准帧、
保留位

数据长度

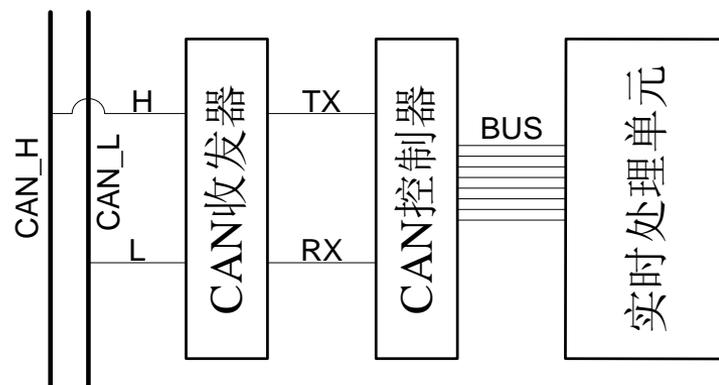
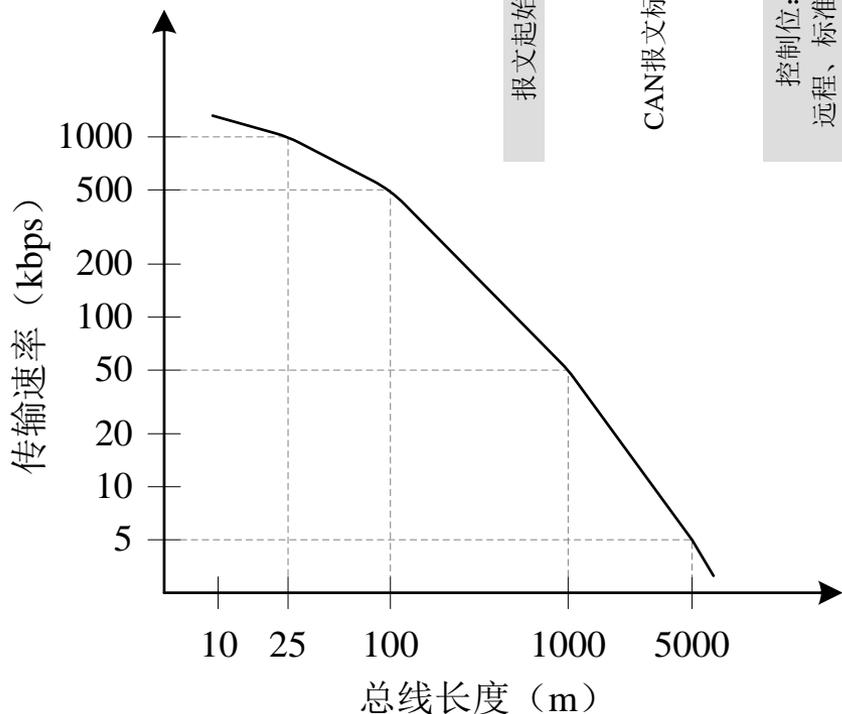
数据域

CRC校验

报文应答

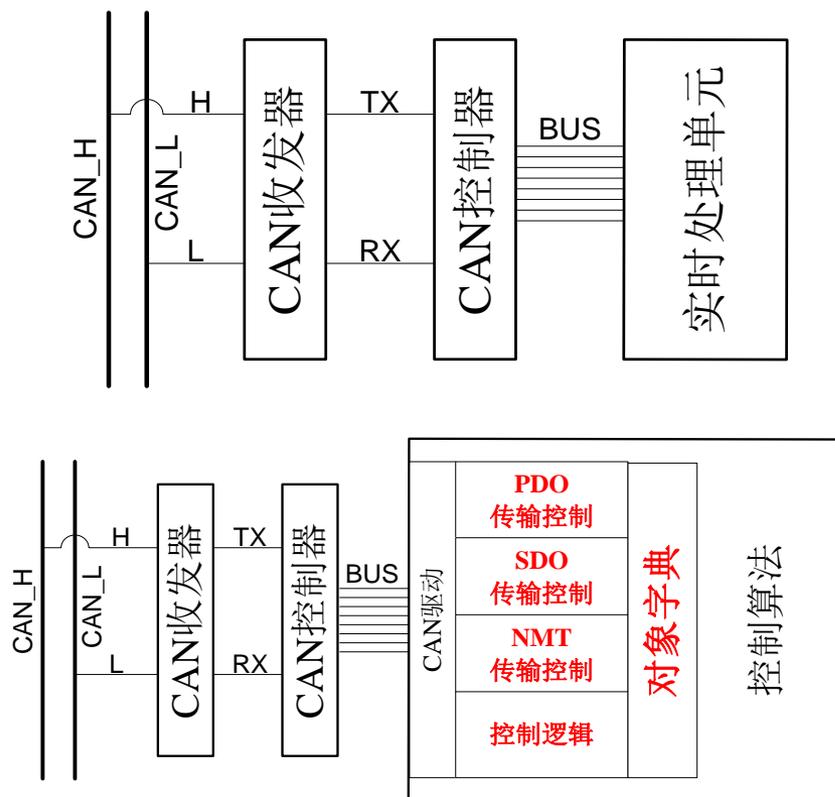
报文结束停止位

报文间隔



CANopen应用层协议

报文类型	功能码	COB-ID范围
NMT	0000	000h
SYNC	0001	080h
EMERGENCY	0001	081h~0FFh
TIME	0010	100h
PDO	0011-1010	181h~57Fh
SDO	1011-1100	581h~67Fh
NMT Error Control	1110	701h~77Fh



CANopen主站

■ 功能

- CANopen节点

- NMT Master

 - 发送SYNC、配置从节点、启动网络、监控节点。

■ 实现方式

- 任务循环

- 基于操作系统的多线程

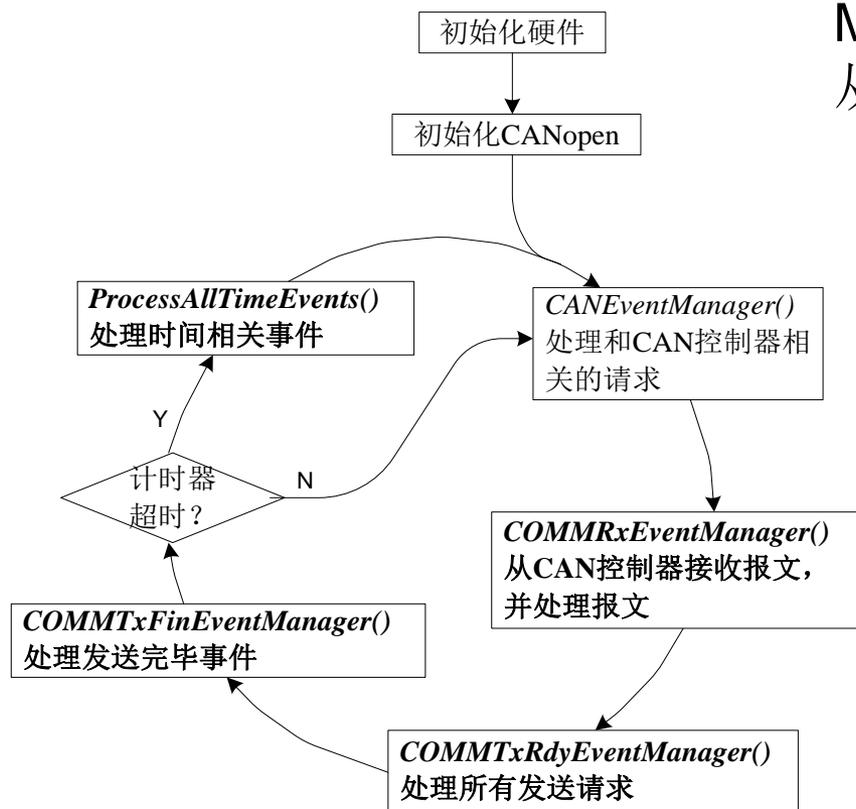
- 编写调度机实现多线程

■ 实现要求

- 实时、并行、动态可配置、可移植

实时、并行、动态可配置

MicroChip CANopen 从站的任务循环结构*

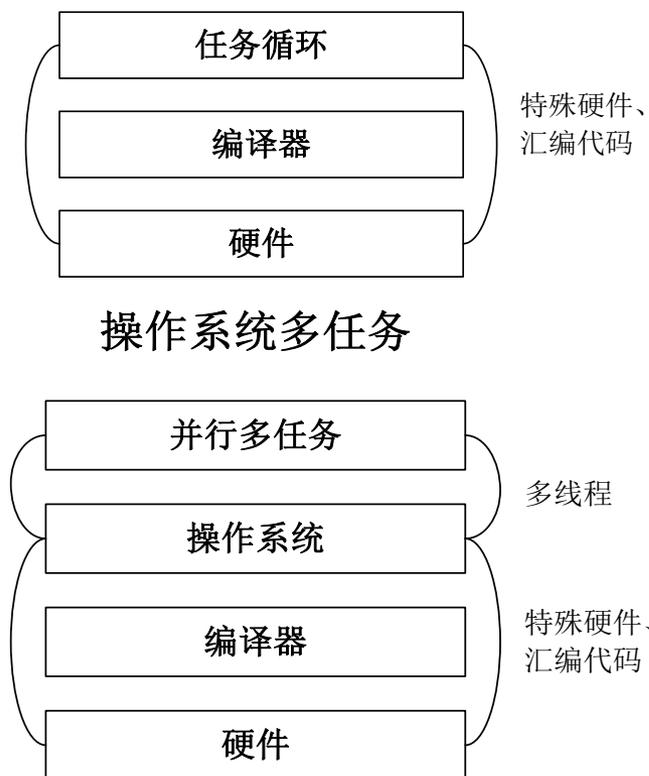


*R. M. Fosler. A CANopen Stack for PIC18 ECAN Microcontrollers. Microchip Technology Inc., 2005.

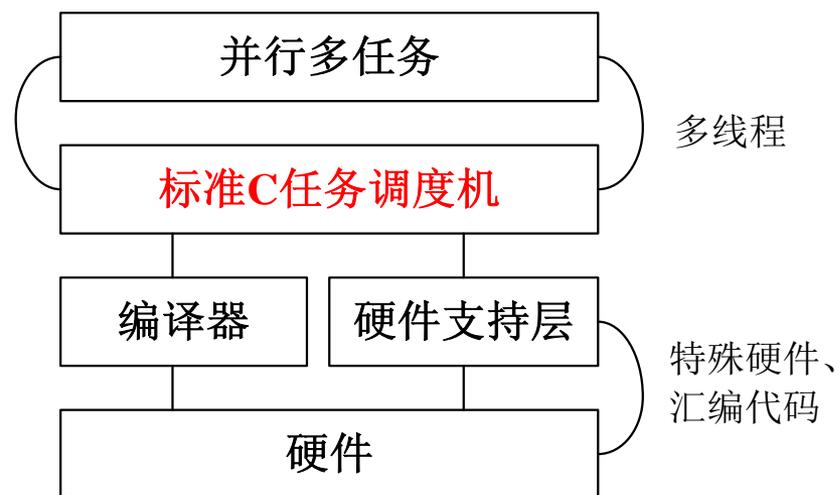
肖进军. 混合动力电动汽车CANopen总线协议的研究与实现. 北京工业大学工学硕士学位论文. 2007

可移植

任务循环



标准C任务调度机



标准C非抢占任务调度机（成果1）

■ 特性

- 动态任务分配
- 固定优先级非抢占式调度
- 小开销线程（无数据内存）
- 支持时间和事件触发方式

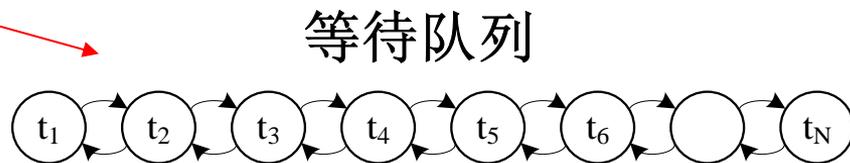
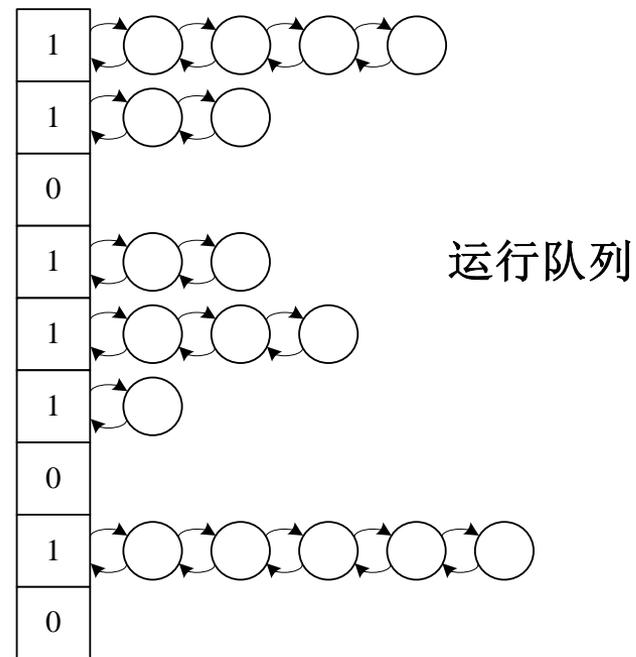
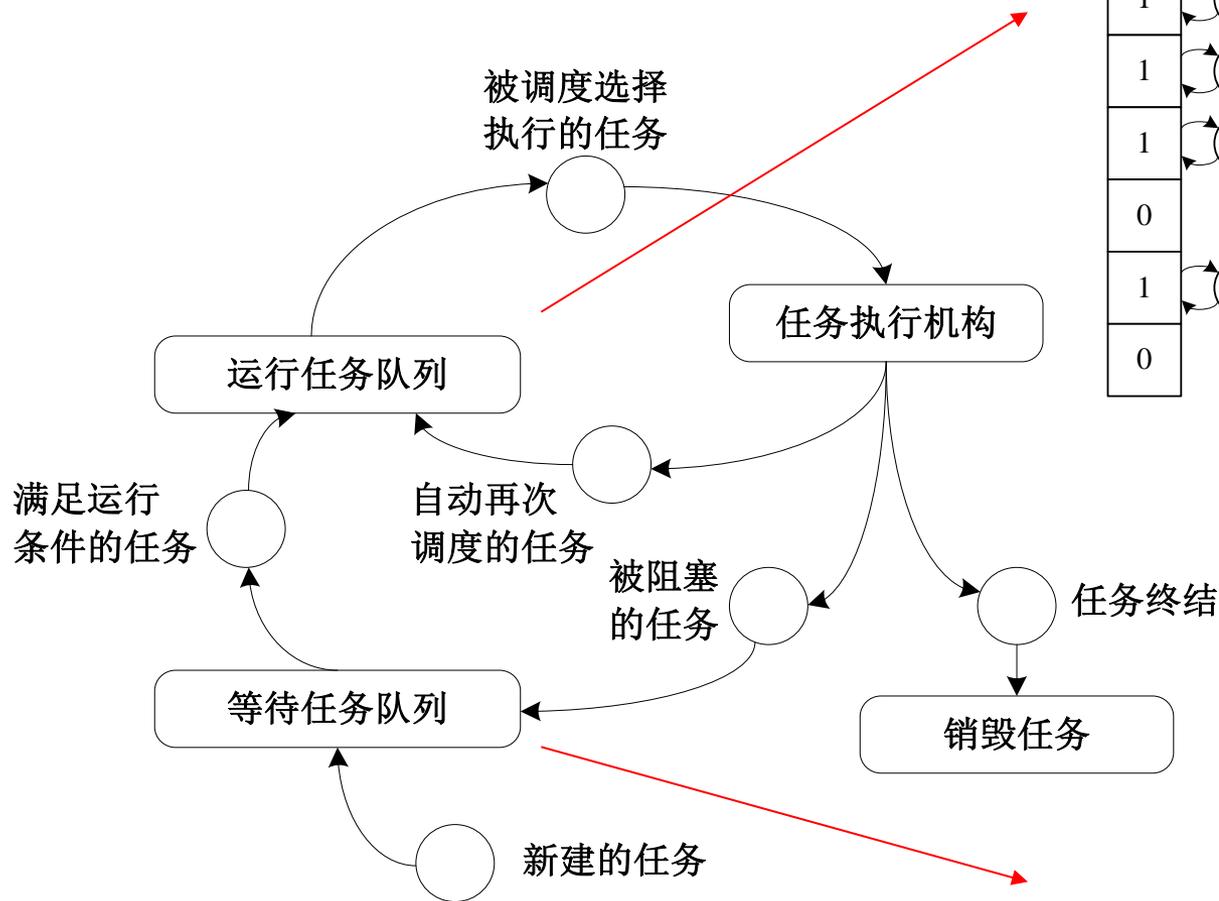
■ 原因

- 抢占式调度机复杂且需要汇编支持
- CANopen的COB-ID直接提供优先级定义
- CANopen为一个软实时系统

任务结构

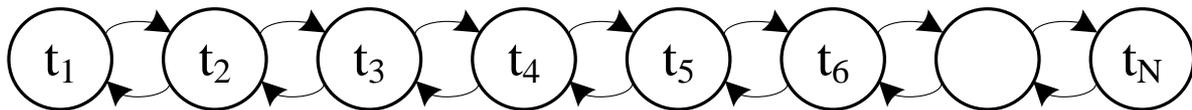
```
typedef struct _TaskObj {  
    char runPrio;           ← 固定优先级  
    char state;           ← 运行状态  
    char (*pFun) (struct _TaskObj *);  执行函数  
    int Argu[10];         ← 数据空间  
    int timeHigh;        } ← 时间信息  
    int timeLow;        }  
    struct _TaskObj * pNext; } ← 链表成员  
    struct _TaskObj * pPre; }  
} TaskObj, *pTaskObj;
```

任务调度机



等待队列的优化

等待队列



m 等待任务队列任务数

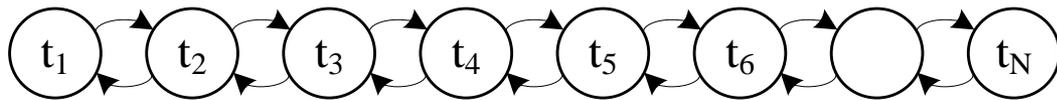
e 包含时间信息的任务
占任务总数的平均概率

r 调度周期内满足条件的
新任务

n 调度周期内产生的新
任务

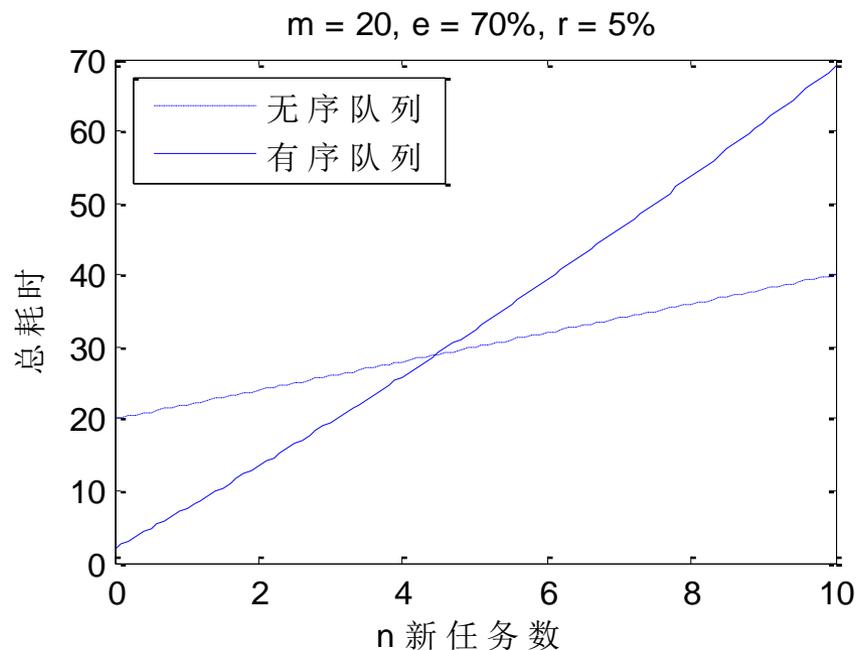
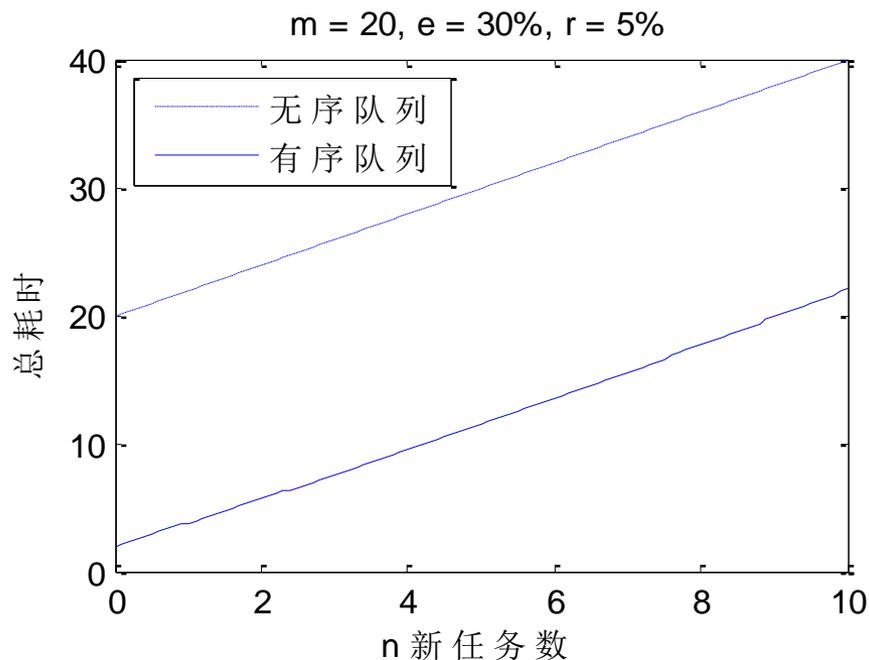
等待队列

$$t_1 < t_2 < t_3 < \dots < t_N$$



等待队列的优化

- m 等待任务队列任务数
- e 包含时间信息的任务占任务总数的平均概率
- r 调度周期内满足条件的新任务
- n 调度周期内产生的新任务



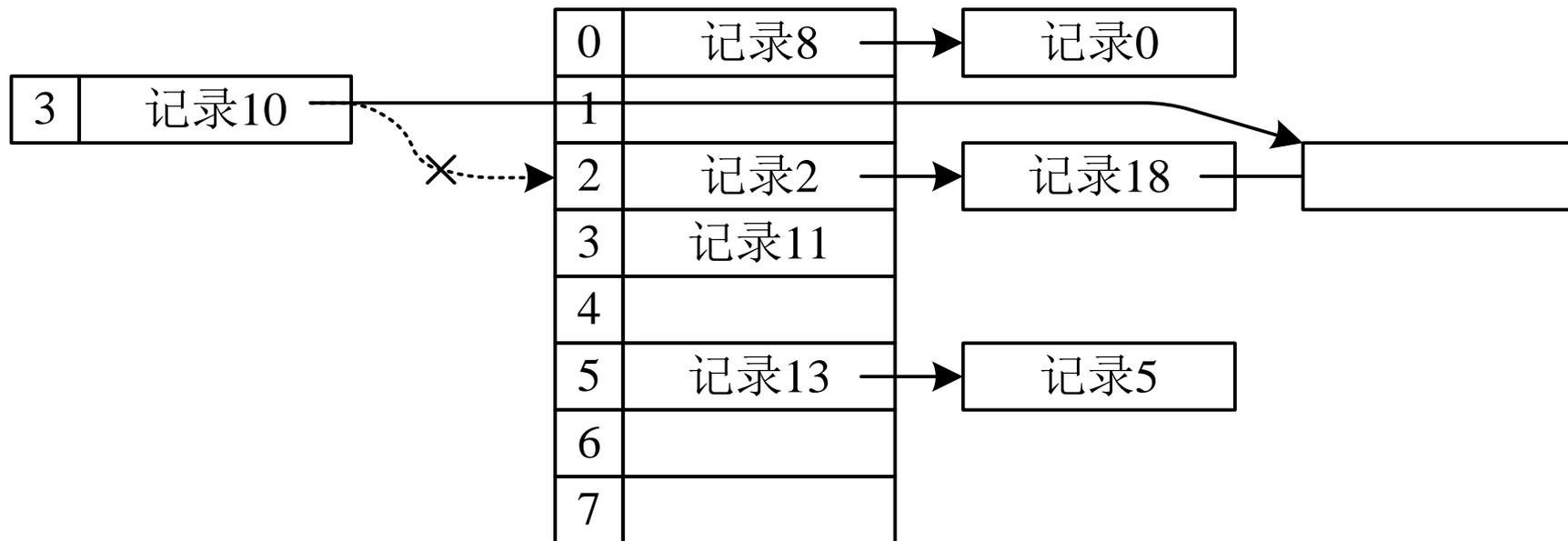
$$t = m + 2n \quad t' = \frac{e^2 n^2 + (2me^2 - e^2 - 4e + 4r + 4)n + 4mr + 4}{4}$$

对象字典——数组映射



散列表（链表法）（成果2）

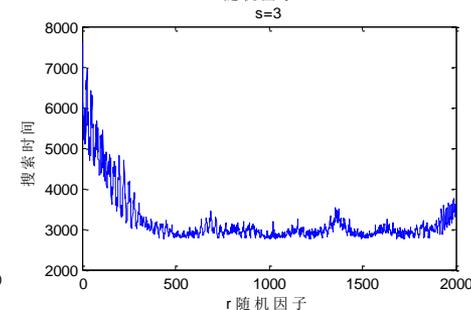
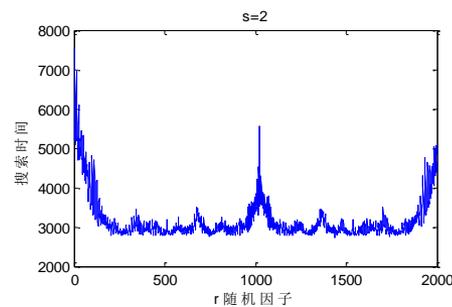
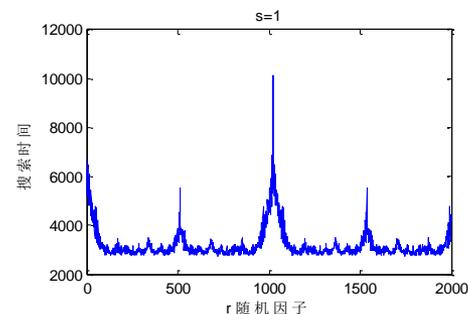
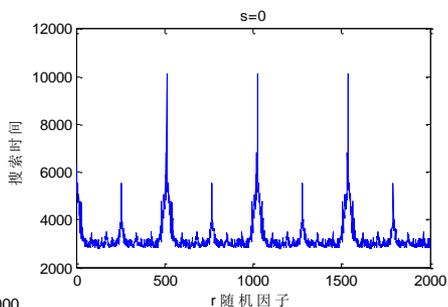
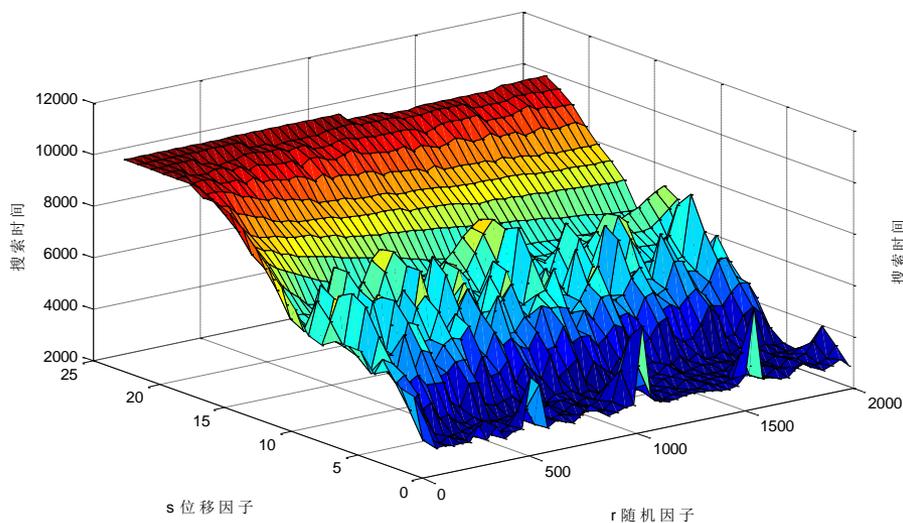
$$\text{hash}(\text{addr}) = \text{addr} \% 8$$



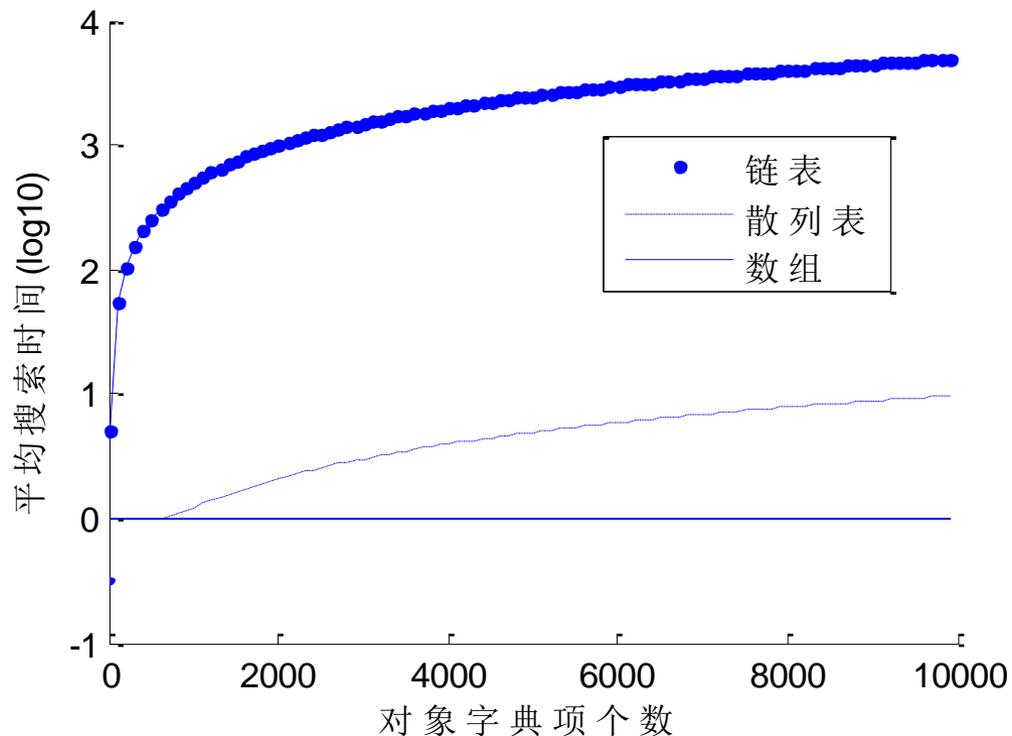
散列公式

$$h_addr = ((index * r \square s) + subIndex) \% N$$

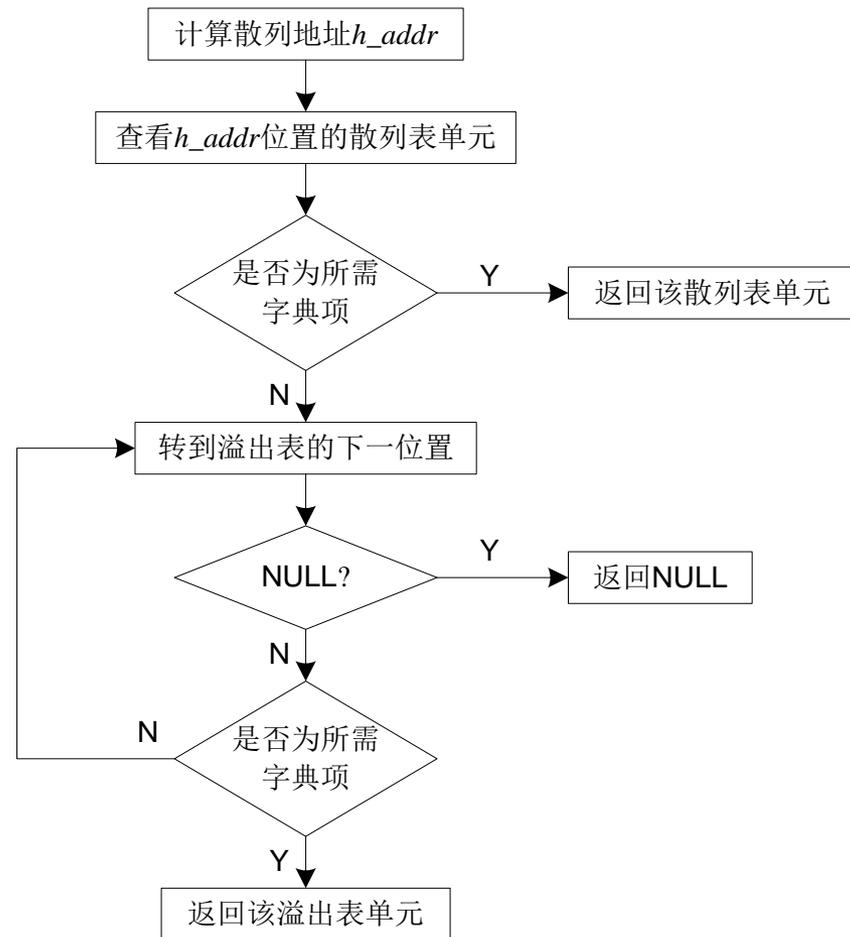
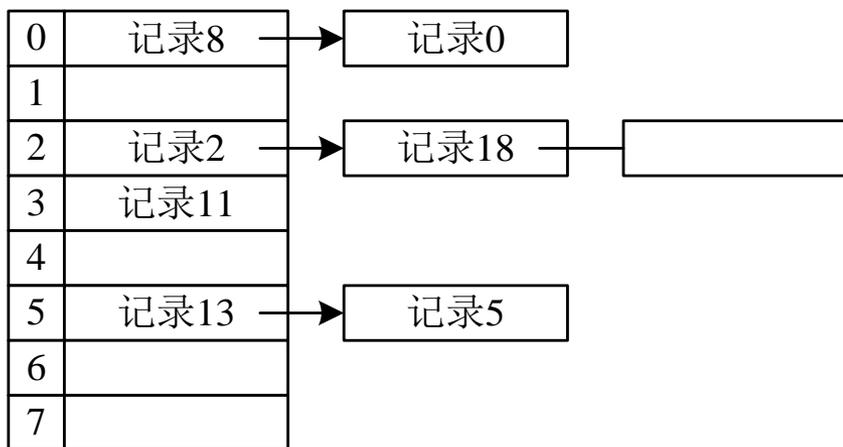
$$h_addr = (index * 1438 \square 2 + subIndex) \% 512$$



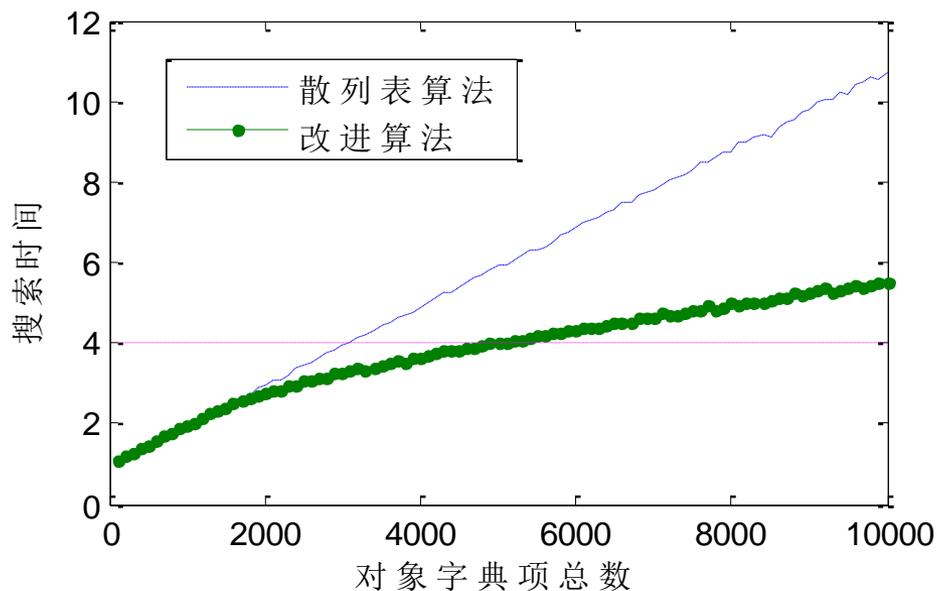
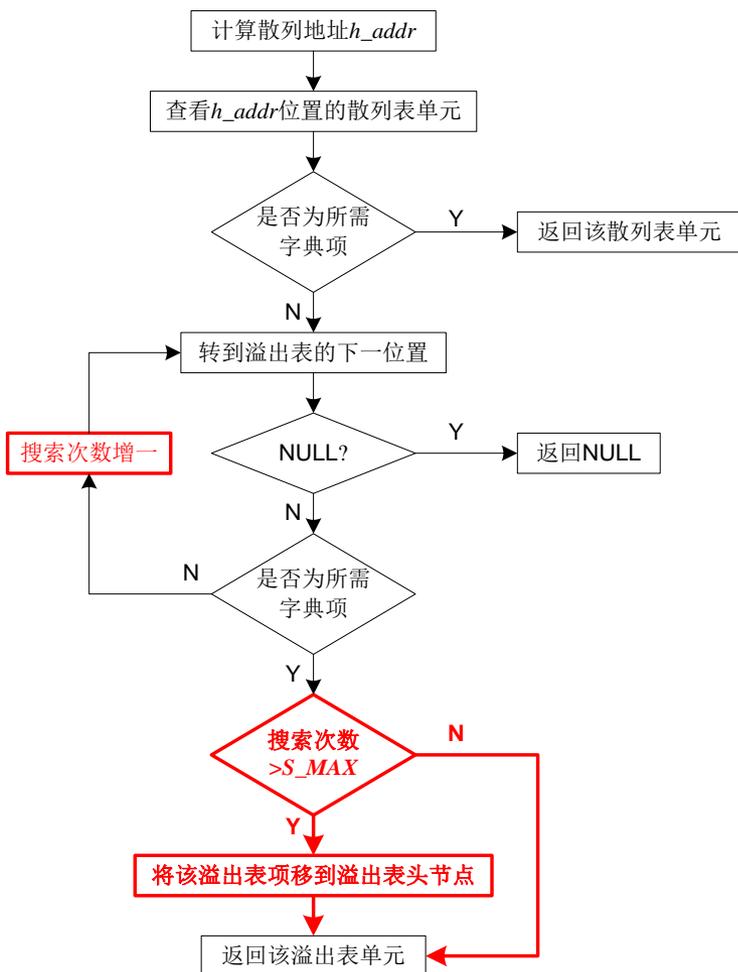
散列表算法性能



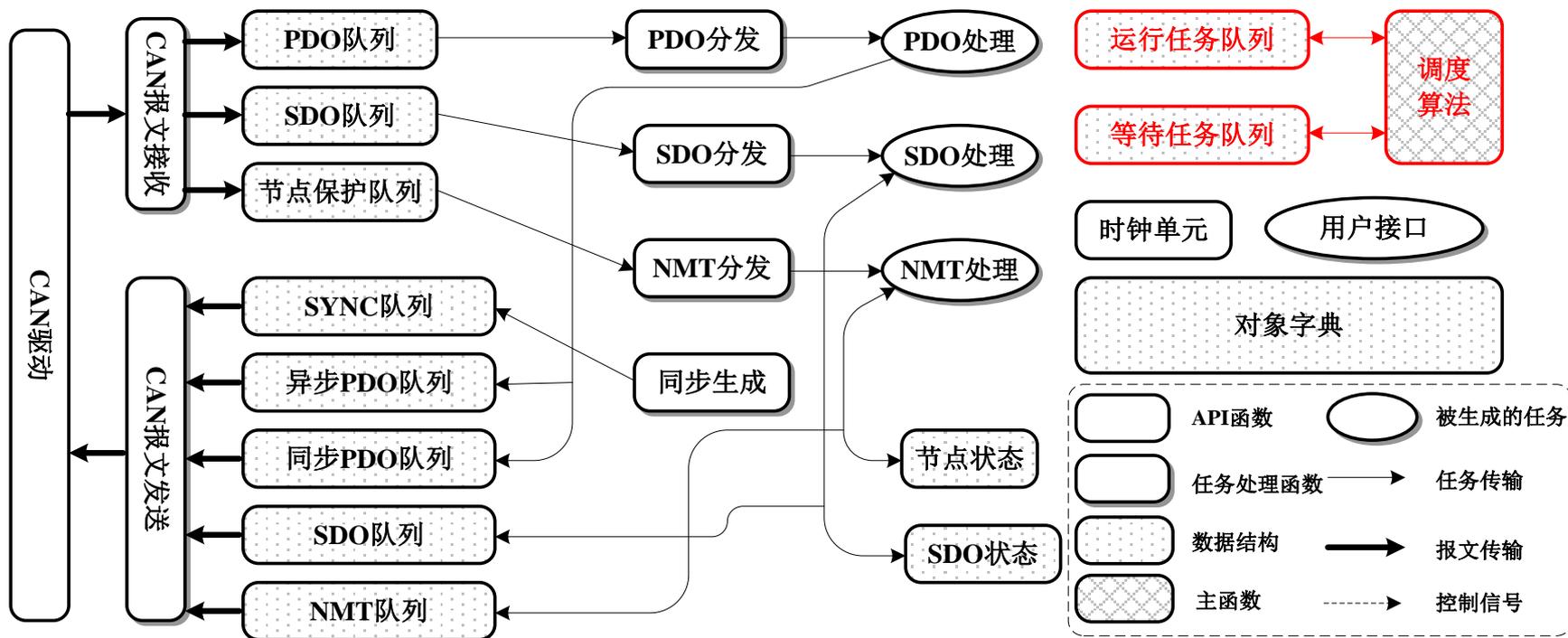
改进的散列表算法



改进的散列表算法



基于调度机的CANopen主站协议栈 (成果3)



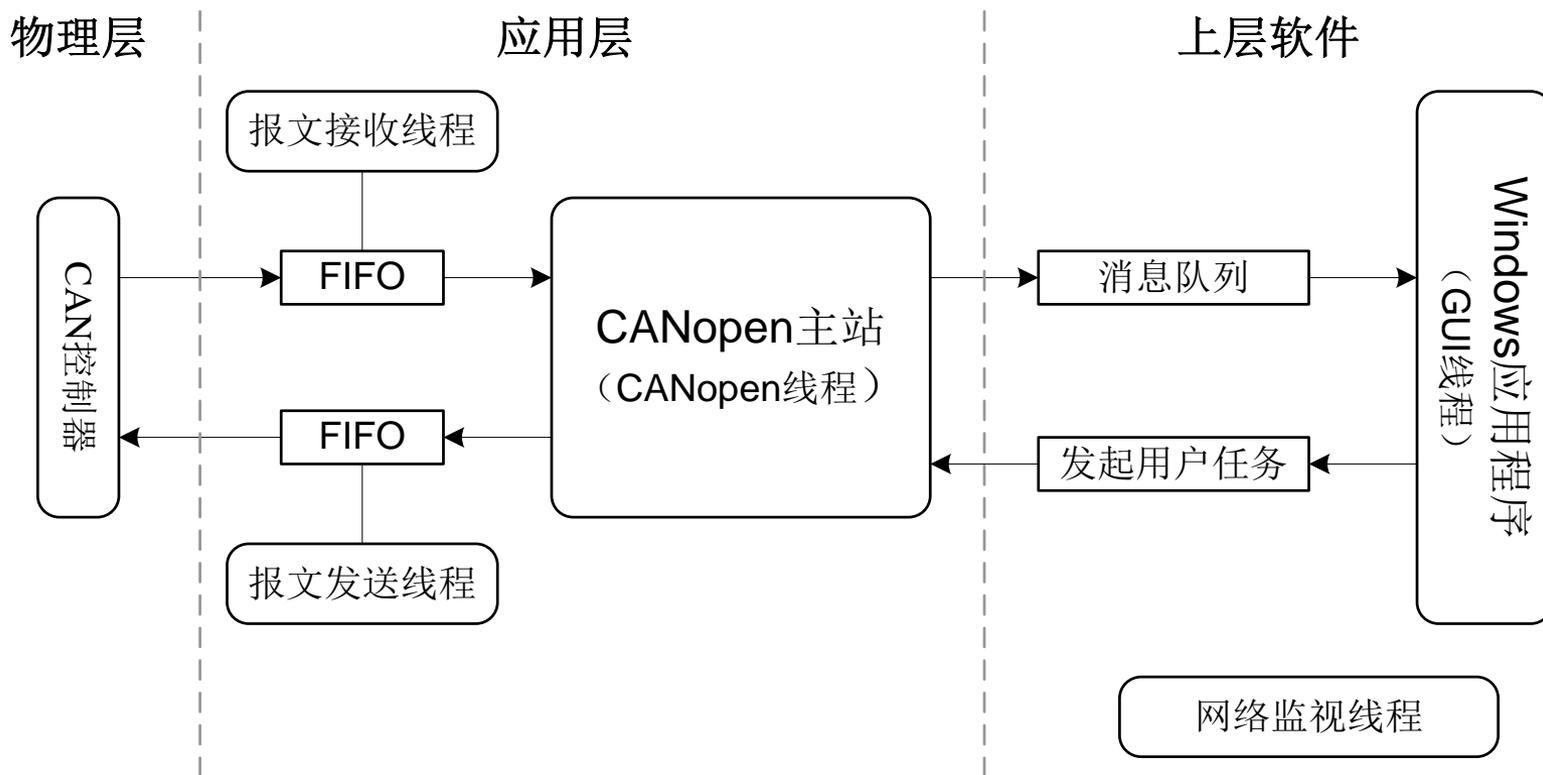
CANopen协议栈性能

	功能	必要性	CanFestival	本文的CANopen
CANopen应用层和 通讯协议	SYNC的生成	必要	完成	完成, 最小200微秒
	SDO客户端	必要	部分, 仅支持加速方式	部分, 仅支持加速方式
	SDO服务器端	必要	完成, 仅支持1个	完成, 可支持多个
	PDO发送	必要	完成	完成
	PDO接收	必要	完成	完成
	心跳报文机制	必要	完成	完成
	节点保护机制	必要	部分完成	完成
	从节点状态控制	必要	完成	完成
	紧急报文	必要	不支持	通知上层
	时间戳	可选	不支持	不支持

CANopen协议栈性能

	功能	必要性	CanFestival	本文的CANopen
CANopen 管理者框架协议	节点启动检查	可选	不支持	部分，除配置检查的所有部分
	配置及软件下载	可选	不支持	不支持
	第三方NMT命令	可选	不支持	完成
	SDO通道建立	可选	不支持	不支持
	主站竞争机制	可选	不支持	不支持
其它	PDO和SDO最大支持数量		宏配置，运行时不可改	支持CANopen规定的最大数量，受系统运行速度制约
	对象字典项的新建与删除		不支持	完成
	直接对从节点对象字典操作		不支持	完成
	可移植性		仅支持类linux操作系统	部分修改后可运行于所有支持C编译器的操作系统和无操作系统平台

WindowsXP CANopen测试平台 (成果4)



测试网络

协议栈名称	节点类型	硬件平台	协议栈
CANopen主站	主站	Pentium-4 3.0G, WindowsXP	自主开发
BK5120 总线耦合器	从节点	Beckhoff成品节点	未知
汽车仪表节点*	从节点	ARM9 Linux	Festival
电动机节点#	从节点	DSP, TMS320F2812, 无操作系统	MicroChip CANopen

* 陈涛. 汽车仪表的CANopen节点通信的研究与实现. 北京工业大学工学硕士学位论文. 2007

The CanFestival CANOpen stack manual. Can Festival OpenSource Project. 2007. <http://sourceforge.net/projects/canfestival>

肖进军. 混合动力电动汽车CANopen总线协议的研究与实现. 北京工业大学工学硕士学位论文. 2007

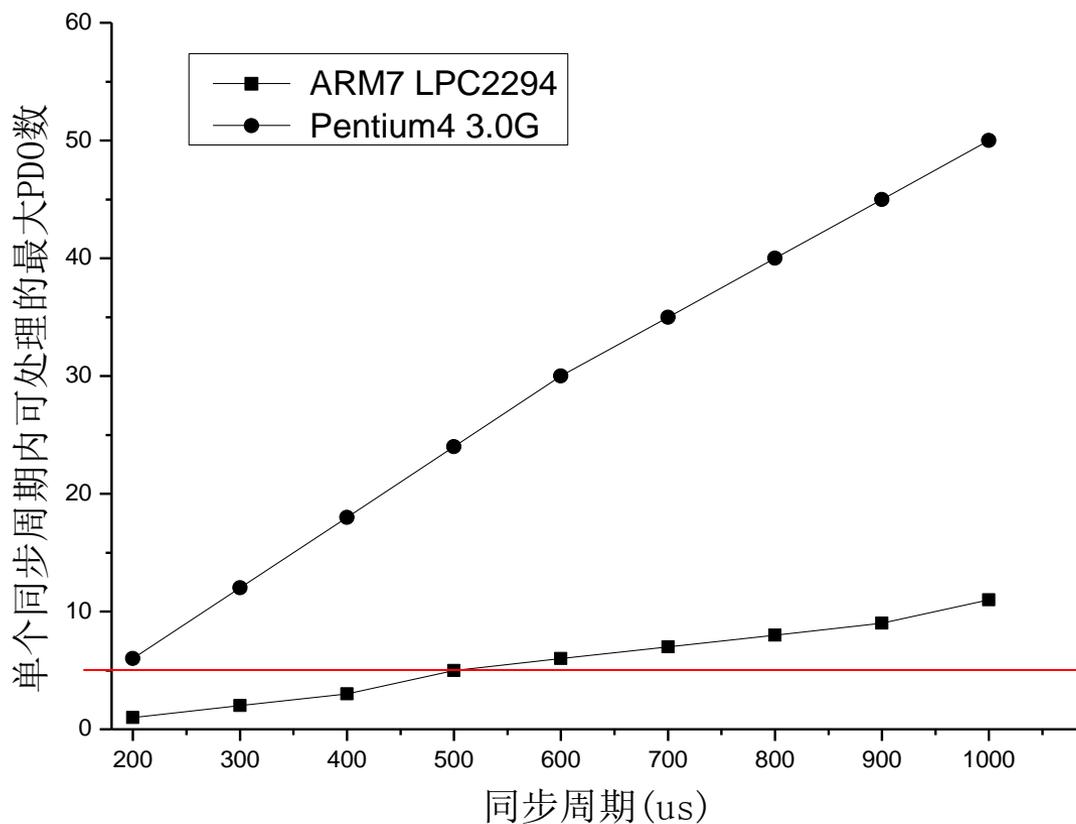
R. M. Fosler. A CANopen Stack for PIC18 ECAN Microcontrollers. Microchip Technology Inc., 2005.

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en020605

北京工业大学 电子信息与控制工程学院

School of Electronic Info. & Control Engineering, Beijing University of Technology

CANopen实时性测试



CANopen 节点启动测试

分:秒 报文类型: 报文内容

```
10:15 NMT CMD: COB 0 rtr 0 data 82 3
10:18 NMT Err: COB 703 rtr 0 data 0
10:18 Emergency COB 83 rtr 0 data 0 50 81 0 1 10 4 80
10:18 Emergency COB 83 rtr 0 data 0 50 81 0 1 10 4 81
10:18 Emergency COB 83 rtr 0 data 0 50 81 0 1 10 4 82
10:18 Emergency COB 83 rtr 0 data 0 50 81 0 1 10 4 83
10:20 RSDO: COB 603 rtr 0 data 40 0 10 0 0 0 0 0
10:20 TSDO: COB 583 rtr 0 data 43 0 10 0 91 1 f 0
10:20 RSDO: COB 603 rtr 0 data 40 18 10 1 0 0 0 0
10:20 TSDO: COB 583 rtr 0 data 43 18 10 1 2 0 0 0
10:20 RSDO: COB 603 rtr 0 data 40 18 10 2 0 0 0 0
10:20 TSDO: COB 583 rtr 0 data 43 18 10 2 0 14 1 0
10:20 RSDO: COB 603 rtr 0 data 40 18 10 3 0 0 0 0
10:20 TSDO: COB 583 rtr 0 data 43 18 10 3 0 c4 1 0
10:20 RSDO: COB 603 rtr 0 data 40 18 10 4 0 0 0 0
10:20 TSDO: COB 583 rtr 0 data 43 18 10 4 2 44 0 0
10:20 RSDO: COB 603 rtr 0 data 40 17 10 0 0 0 0 0
10:20 TSDO: COB 583 rtr 0 data 4b 17 10 0 0 0 0 0
10:20 RSDO: COB 603 rtr 0 data 2b 17 10 0 e8 3 0 0
10:20 TSDO: COB 583 rtr 0 data 60 17 10 0 0 0 0 0
10:20 NMT Err: COB 703 rtr 0 data 7f
10:21 NMT Err: COB 703 rtr 0 data 7f
10:22 NMT Err: COB 703 rtr 0 data 7f
10:23 NMT Err: COB 703 rtr 0 data 7f
```

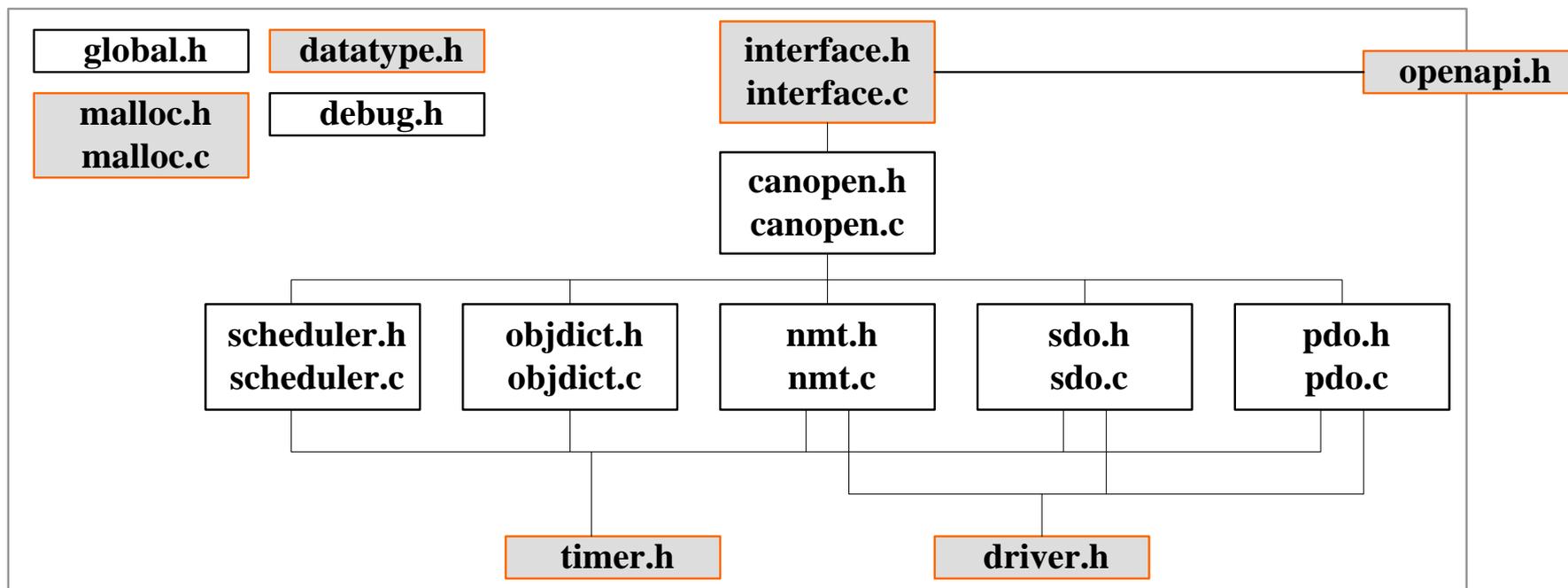
启动节点

节点启动报文

节点启动检查

心跳报文

关于移植的考虑



课题总结

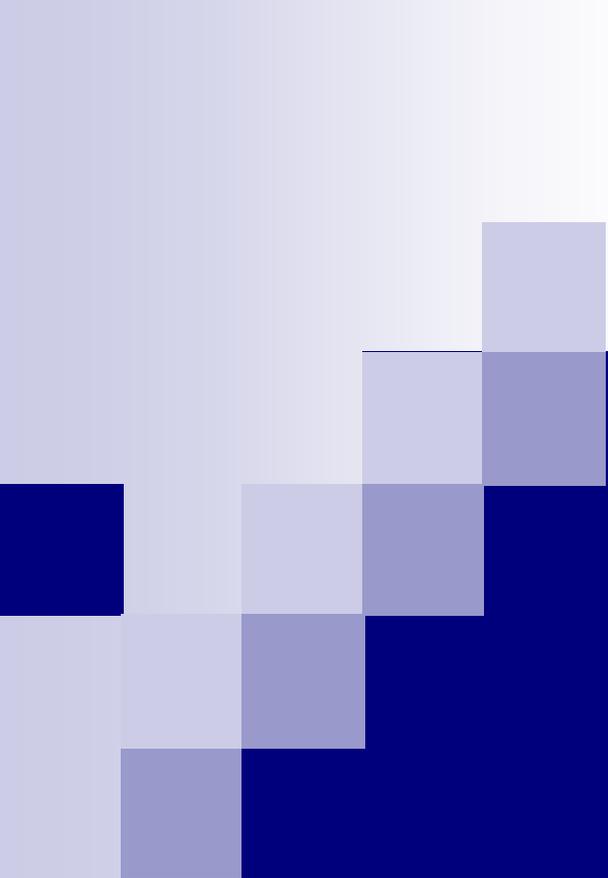
- 完成了一个可移植的实时可配置CANopen主站协议栈，并通过了实际网络测试
- 两个创新点
 - 基于标准C语言的任务调度机
 - 可移植性和高实时特性
 - 基于散列表的对象字典设计
 - 在不大量损失速度的前提下提供了可配置性

其他细节

- 国内第一个自主设计的具备完备功能的CANopen主站系统。
- 该CANopen主站协议栈已经成功移植到uC/OS-II嵌入式系统上 (Freescale 16位CPU)。
- 对任务队列等核心数据对象采用了锁机制，解决了在有操作系统情况下的互斥问题。
- 改进核心任务调度算法，使CANopen主站在空闲情况下主动睡眠，提高了整个系统的实时性能。
- 完成了用户接口函数的接口标准化，在WindowsXP平台下，所有的接口函数可编译成dll文件提供给第三方。
- 在WindowsXP平台下可以和高层数据库结合使用（MySQL）。

论文发表情况

1. Wei Song, Shizhen Yan, Zhe Xu and Suiming Fang. Transplantable CANopen Master Based on Non-preemptive Task Scheduler. *Proceedings of the 1st IEEE International Conference on Automation and Logistics*. 2007: 557-562
2. 宋威, 方穗明, 张明杰, 徐喆. 任务调度在CANopen主站设计中的应用. *计算机测量与控制*. 2008, 16(4)
3. 宋威, 方穗明, 姚丹, 张立超, 钱程. 多FPGA设计的时钟同步. *计算机工程*. 2008, 34(7)
4. 宋威, 方穗明. 基于BUFGMUX与DCM的FPGA时钟电路设计. *现代电子技术*. 2006, 29(2): 141-143
5. 张明杰, 余春暄, 宋威. CANopen协议栈的实现与MySQL数据库的结合. *计算机应用研究*. 2008, 25(z)
6. 闫士珍, 徐喆, 宋威, 张卓. 基于MC9S12DP512和 μ C/OS-II的CANopen主站开发. *计算机工程与科学*. 2008, 30



谢谢各位老师！