

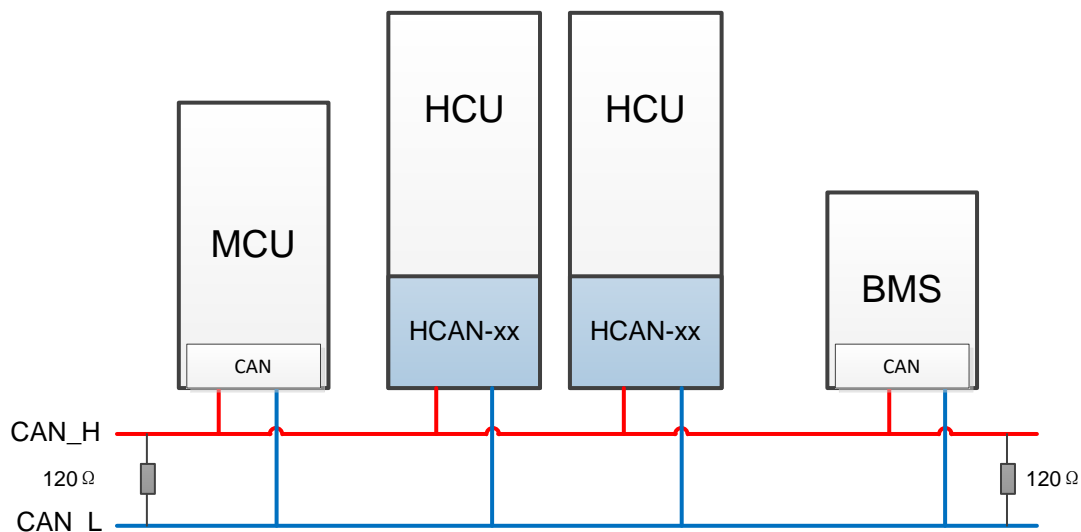
MD880 自定义 CAN 通讯模块 HCAN-X0 应用说明 v2.0

2020 年 8 月 4 日

1 系统方案

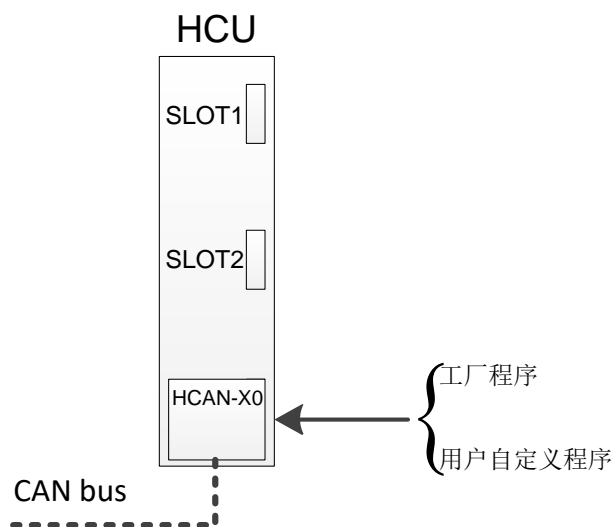
1.1 背景

目前 MD880 只支持 CANopen 通讯协议，CAN 特殊协议不支持，例如 SAE J1939。为满足不同的现场 CAN 通讯需求，基于 HCAN-10 当前硬件资源，开发自定义 CAN 通讯卡 HCAN-X0。



1.2 基本方案

- (1) HCAN-X0 硬件借用 HCAN-10 (CANopen 通讯模块)。
- (2) HCAN-X0 与 HCU 插接方式和标准扩展卡一致，可插接于任意扩展插槽；
- (3) HCU 识别 HCAN-X0 为通讯模块的一种，和 Profibus DP、CANopen、ProfiNET IO 等平行；



- (4) HCAN-xx 程序包含工厂程序和用于自定义程序两个部分组成，工厂程序包含 HCAN-X0 和 HCU 之间的数据交

互、系统调度函数和 CAN 底层配置等部分，并提供由用户自定义程序调用的相关接口函数。用户自定义程序可由终端用户自定义编写、系统集成商、或者工程部同事完成，用户自定义程序需把接收的 CAN 数据进行正确地解析，转化为与 HCU 标准通讯接口匹配的数据，再通过工厂程序提供的接口函数或变量，将控制数据发送至 HCU，实现对 MD880 驱动器的控制；同时，用户程序可以通过厂家程序提供的接口函数或变量获取到 HCU 的数据，实现对 MD880 驱动器实时状态的监视，再转化为所需的格式，通过 CAN 接口发送出去。

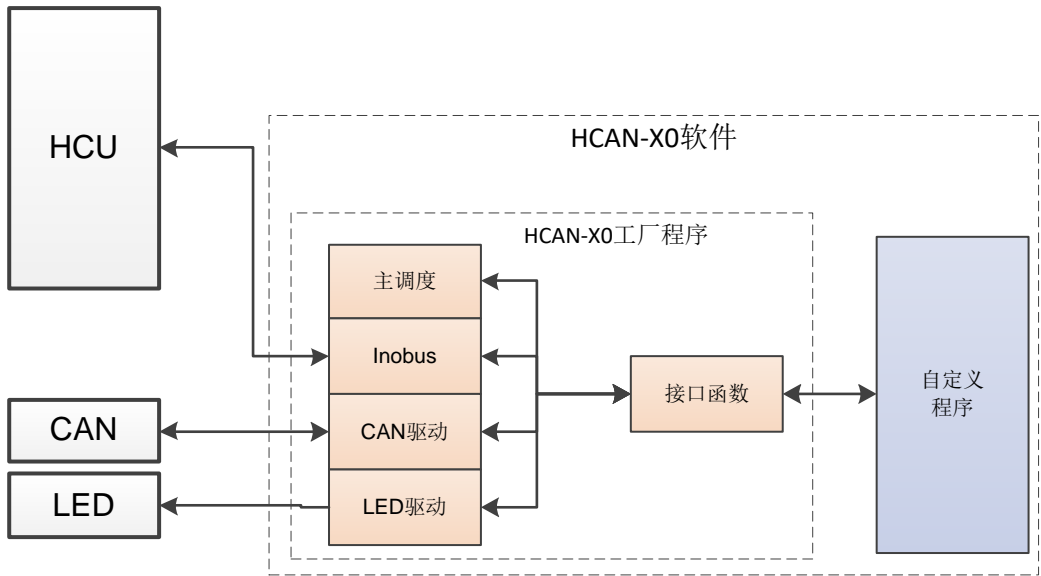
1.3 限定条件

- (1) 物理介质中采用屏蔽双绞线。此双绞线特定阻抗为 120Ω，电流对称驱动，两条线命名为 CAN_H 和 CAN_L。
- (2) 基本属性

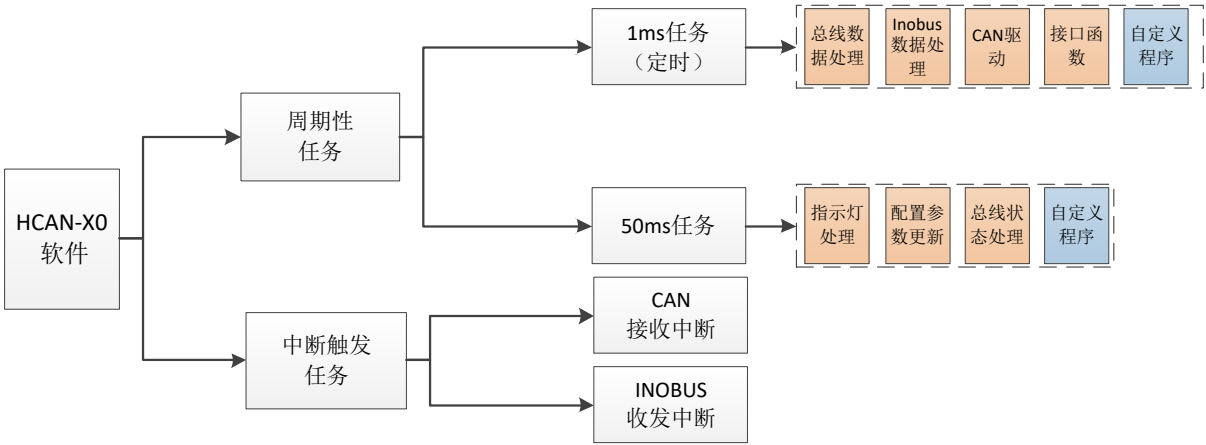
链路层协议	CAN总线
应用层协议	自定义
CAN-ID类型	11bits-CAN2.0A 29bits-CAN2.0B
波特率	500Kbit/s（默认） 1Mbit/s 、250 Kbit/s、125 Kbit/s 100 Kbit/s 、50 Kbit/s、20 Kbit/s
CAN帧长度	0~8字节
终端匹配电阻	120Ω

2 软件方案

2.1 软件结构



2.2 软件调度



3 接口定义

3.1 HCAN 与 HCU 交互数据

(1) 数据结构定义

```
typedef struct
{
    Uint16 Para01;           // n36-11
    Uint16 Para02;           // n36-12
    Uint16 Para03;           // n36-13
    Uint16 Para04;           // n36-14
    Uint16 Para05;           // n36-15
    Uint16 Para06;           // n36-16
    Uint16 Para07;           // n36-17
    Uint16 Para08;           // n36-18
    int16  Para09;           // n36-19
    int16  Para10;           // n36-20
    int16  Para11;           // n36-21
    int16  Para12;           // n36-22
    int16  Para13;           // n36-23
    int16  Para14;           // n36-24
    int16  Para15;           // n36-25
    int16  Para16;           // n36-26
}PARAM_CODE_DEF;

typedef struct
{
    Uint16 SendBuff[16];     // 发送buff
    Uint16 RecvBuff[16];     // 接受buff
}PDO_DATA_DEF;

extern PARAM_CODE_DEF ParaCode;
extern PDO_DATA_DEF PdoData;
extern Uint16 GetCanBaudRate(void);
```

注意：交互数据使用的结构体定义和声明的变量不可修改，否则会编译出错。

(2) 交互数据详细说明

表 1 HCU 发送给 HCAN-X0 实时数据 PdoData.RecvBuff

序号	变量	方向	周期 (ms)	功能码 (总线适配器 A)	功能码 (总线适配器 B)
1	PdoData.RecvBuff[0]	HCU->HCAN	2	n1-04	n2-04
2	PdoData.RecvBuff[1]	HCU-> HCAN	2	n1-05	n2-05
3	PdoData.RecvBuff[2]	HCU-> HCAN	2	n1-06	n2-06
4	PdoData.RecvBuff[3]	HCU-> HCAN	2	n1-07	n2-07
5	PdoData.RecvBuff[4]	HCU-> HCAN	2	n1-08	n2-08
6	PdoData.RecvBuff[5]	HCU-> HCAN	2	n1-09	n2-09
7	PdoData.RecvBuff[6]	HCU-> HCAN	2	n1-10	n2-10
8	PdoData.RecvBuff[7]	HCU-> HCAN	2	n1-11	n2-11
9	PdoData.RecvBuff[8]	HCU-> HCAN	2	n1-12	n2-12
10	PdoData.RecvBuff[9]	HCU-> HCAN	2	n1-13	n2-13
11	PdoData.RecvBuff[10]	HCU-> HCAN	2	n1-14	n2-14
12	PdoData.RecvBuff[11]	HCU-> HCAN	2	n1-15	n2-15
13	PdoData.RecvBuff[12]	HCU-> HCAN	2	n1-16	n2-16
14	PdoData.RecvBuff[13]	HCU-> HCAN	2	n1-17	n2-17
15	PdoData.RecvBuff[14]	HCU-> HCAN	2	n1-18	n2-18
16	PdoData.RecvBuff[15]	HCU-> HCAN	2	n1-19	n2-19

表 2 HCU 发送给 HCAN-X0 参数 (ParaCode)

序号	变量	方向	周期 (ms)	功能码
1	ParaCode.Para01	HCU-> HCAN	1000	n18-05
2	ParaCode.Para02	HCU-> HCAN	1000	n18-06
3	ParaCode.Para03	HCU-> HCAN	1000	n18-07
4	ParaCode.Para04	HCU-> HCAN	1000	n18-08
5	ParaCode.Para05	HCU-> HCAN	1000	n18-09
6	ParaCode.Para06	HCU-> HCAN	1000	n18-10
7	ParaCode.Para07	HCU-> HCAN	1000	n18-11
8	ParaCode.Para08	HCU-> HCAN	1000	n18-12
9	ParaCode.Para09	HCU-> HCAN	1000	n18-13
10	ParaCode.Para10	HCU-> HCAN	1000	n18-14
11	ParaCode.Para11	HCU-> HCAN	1000	n18-15
12	ParaCode.Para12	HCU-> HCAN	1000	n18-16
13	ParaCode.Para13	HCU-> HCAN	1000	n18-17
14	ParaCode.Para14	HCU-> HCAN	1000	n18-18
15	ParaCode.Para15	HCU-> HCAN	1000	n18-19
16	ParaCode.Para16	HCU-> HCAN	1000	

表 3 HCAN 发送至 HCU 实时数据 (PdoData.SendBuff)

序号	变量	方向	周期 (ms)	总线适配配置 A		总线适配器 B	
				功能码	连接器	功能码	连接器
1	PdoData.SendBuff[0]	HCAN->HCU	2	n1-68	-	n2-68	-
2	PdoData.SendBuff[1]	HCAN->HCU	2	n1-69	-	n2-69	-
3	PdoData.SendBuff[2]	HCAN->HCU	2	n1-70	-	n2-70	-

序号	变量	方向	周期 (ms)	总线适配配置 A		总线适配器 B	
				功能码	连接器	功能码	连接器
4	PdoData.SendBuff[3]	HCAN->HCU	2	n1-71	-	n2-71	-
5	PdoData.SendBuff[4]	HCAN->HCU	2	n1-72	-	n2-72	-
6	PdoData.SendBuff[5]	HCAN->HCU	2	n1-73	-	n2-73	-
7	PdoData.SendBuff[6]	HCAN->HCU	2	n1-74	-	n2-74	-
8	PdoData.SendBuff[7]	HCAN->HCU	2	n1-75	-	n2-75	-
9	PdoData.SendBuff[8]	HCAN->HCU	2	n1-76	-	n2-76	-
10	PdoData.SendBuff[9]	HCAN->HCU	2	n1-77	-	n2-77	-
11	PdoData.SendBuff[10]	HCAN->HCU	2	n1-78	-	n2-78	-
12	PdoData.SendBuff[11]	HCAN->HCU	2	n1-79	-	n2-79	-
13	PdoData.SendBuff[12]	HCAN->HCU	2	n1-80	-	n2-80	-
14	PdoData.SendBuff[13]	HCAN->HCU	2	n1-81	-	n2-81	-
15	PdoData.SendBuff[14]	HCAN->HCU	2	n1-82	-	n2-82	-
16	PdoData.SendBuff[15]	HCAN->HCU	2	n1-83	-	n2-83	-

3.2 HCAN 与外部 CAN 网络交互数据

由用户程序自定义。

3.3 LED 控制

HCAN-X0 包含 3 个 LED 组，分别为 LED1、LED2、LED3，其中 LED3 可由用户自定义控制；LED1、LED3、LED3 均为 3 色灯组，每个 LED 组包含 3 个不同颜色的指示灯，颜色分别为红色、绿色、橙色。

LED 序号	颜色	描述
LED1 组	红、绿、橙	InoBus 通讯建立指示灯，正常应为绿灯常亮
LED2 组	红、绿、橙	与 HCU 的参数、协议配置匹配指示，正常应为绿灯常亮
LED3 组	红、绿、橙	用户自定义控制

用户程序可调用以下 3 个函数控制 LED3 的不同颜色灯状态。

说明：

形参 onMs：标示亮保持时间，单位 ms

形参 offMs：表示灭保持时间，单位 ms

onMs	offMs	指示灯状态
0	0	灭
非 0	0	亮
0	非 0	灭
非 0	非 0	亮灭周期闪烁， 灭灯时间由 onMs 控制，灭灯由 offMs 控制

void ModuleLed3RedDispSet(Uint16 onMs, Uint16 offMs); // 控制用户指示灯 红

void ModuleLed3GreenDispSet(Uint16 onMs, Uint16 offMs); // 控制用户指示灯 绿

void ModuleLed3OrangeDispSet(Uint16 onMs, Uint16 offMs); // 控制用户指示灯 橙

3.4 CPU 负荷率

- (1) 调用以下 2 个函数可以获取当前 CPU 负载率。
- (2) 单位为%，如获取数值为 25，则表示当前 CPU 负荷率为 25%。
- (3) 用户编写自定义程序时，应关注 CPU 负荷率。一般情况下系统最大负荷率不要超过 90%；

Uint16 GetSysLoadMax(void);

// 获取系统负荷最大值

Uint16 GetSysLoadRealTime(void);

// 获取系统实时负荷率

3.5 用户自定义函数

- (1) 用户自定义的执行通过 userFunc1MsProc()、userFunc50MsProc()、userFuncInitProc()调用；
- (2) userFunc1MsProc()执行周期为 1ms，userFunc50MsProc()执行周期为 50ms，userFuncInitProc()只上电初始化 1 次。

函数名称	描述
userFunc1MsProc()	1ms 调用 1 次周期循环函数
userFunc50MsProc()	50ms 调用 1 次周期循环函数
userFuncInitProc()	上电只调用 1 次初始化函数

4 开发和维护

4.1 开发模式

- (1) 厂家程序提供 obj 工程，用户自定义编写用户程序；
- (2) 由用户程序解析 CAN 协议和转化；
- (3) 由用户程序控制 CAN 数据的收发；

4.2 维护

- (1) 由用户程序开发人员控制软件版本
- (2) 烧录方式：开发状态或者试用版本使用 HEX 文件串口烧录或者仿真器升级；稳定版本申请非标号，由公司进行加密 BIN 文件在线升级。

4.3 参数示例

0001	n36-00	扩展插槽选择	1[扩展插槽1_1]	-
0002	n36-01	模块在线状态	1[在线]	-
0003	n36-03	波特率选择	5[500kbps]	-
0004	n36-04	通信协议设定	18181	-
0005	n36-11	通信卡参数1	1	-
0006	n36-12	通信卡参数2	2	-
0007	n36-13	通信卡参数3	3	-
0008	n36-14	通信卡参数4	4	-
0009	n36-15	通信卡参数5	5	-
0010	n36-16	通信卡参数6	6	-
0011	n36-17	通信卡参数7	7	-
0012	n36-18	通信卡参数8	8	-
0013	n36-19	通信卡参数9	9	-
0014	n36-20	通信卡参数10	10	-
0015	n36-21	通信卡参数11	11	-
0016	n36-22	通信卡参数12	12	-
0017	n36-23	通信卡参数13	13	-
0018	n36-24	通信卡参数14	14	-
0019	n36-25	通信卡参数15	15	-
0020	n36-26	通信卡参数16	16	-

0001	n1-00	总线适配器配套总线类型	10[自定义CAN模块]	-	0
0002	n1-01	通讯断线检测延迟时间	1.00	s	0.00
0003	n1-02	总线适配器通信模式选择	0[标准模式]	-	0
0004	n1-03	通信连续CRC校验错误检测设定	0	-	0
0005	n1-04	过程数据输出1	U7-01[固定值100%]	-	0
0006	n1-05	过程数据输出2	U7-02[固定值200%]	-	0
0007	n1-06	过程数据输出3	U7-01[固定值100%]	-	0
0008	n1-07	过程数据输出4	U7-02[固定值200%]	-	0
0009	n1-08	过程数据输出5	U7-01[固定值100%]	-	0
0010	n1-09	过程数据输出6	U7-02[固定值200%]	-	0
0011	n1-10	过程数据输出7	U7-01[固定值100%]	-	0
0012	n1-11	过程数据输出8	U7-01[固定值100%]	-	0
0013	n1-12	过程数据输出9	U7-01[固定值100%]	-	0
0014	n1-13	过程数据输出10	U7-02[固定值200%]	-	0
0015	n1-14	过程数据输出11	U7-02[固定值200%]	-	0
0016	n1-15	过程数据输出12	U7-01[固定值100%]	-	0
0017	n1-16	过程数据输出13	U7-03[固定值400%]	-	0
0018	n1-17	过程数据输出14	U7-04[固定值600%]	-	0
0019	n1-18	过程数据输出15	U7-05[固定值-100%]	-	0
0020	n1-19	过程数据输出16	U7-06[固定值-200%]	-	0