

一、概述

本标准规定了电机控制器(Motor Control Unit, 简称 MCU)与整车控制器(Vehicle Control Unit, 简称 VCU)之间的通信协议

二、通讯规范

1、数据链路层应遵循的原则

总线通讯速率为：250Kbps

数据链路层的规定主要参考 CAN2.0B 和 SAE J1939 的相关规定

使用 CAN 扩展帧的 29 位标识符并进行了重新定义，以下为 29 标识符的分配表：

IDENTIFIER 11BITS														SRR	IDE		
PRIORITY			R	DP	PDU FORMAT (PF)								SRR	IDE			
3	2	1	1	1	8	7	6	5	4	3							
31	30	29	28	27	26	25	24	23	22	21	20	19	18				
IDENTIFIER EXTENSION 18BITS																	
PF		PDU SPECIFIC (PS)								SOURCE ADDRESS (SA)							
2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PRIORITY 优先级为 3 位，可以有 8 个优先级；

R 一般固定为 0；

DP 现固定为 0；

8 位的 PF 为报文的代码；

8 位的 PS 为目标地址或组扩展；

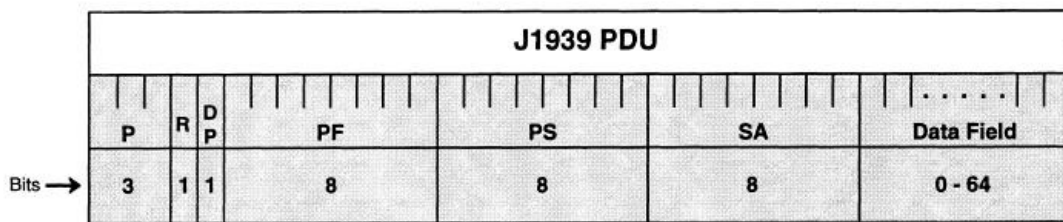
8 位的 SA 为发送此报文的源地址

›接入网络的每一个节点都有名称和地址，名称用于识别节点的功能和进行地址仲裁，地址用于节点的数据通信

›每个节点都至少有一种功能，可能会有多个节点具有相同的功能，也可能一个节点具有多个功能

›对于多字节数据，采用小端方式，如 4660=0x1234，首先发送 0x34，再发送 0x12

2、Protocol Data Unit (PDU):协议数据单元



PDU Specific (PS) 协议数据单元细节：PS 是一个 8 个位字段描述，它由 PDU FORMAT 格式决定。由 PF 决定 PS 是目的单元地址还是组扩展。如果 PF 字段的值小于 240，PS 字段表示的是目的单元地址。如果 PF 字段的值是 240~255，PS 字段表示组扩展值。

	PDU Format (PF) Field	PDU Specified (PS) Field
PDU1 Format	0~239	Destination Address (DA)
PDU2 Format	240~255	Group Extension (GE)

3、CAN 网络地址分配表

CAN 总线结点地址如果 J1939 已有定义，则尽量使用 J1939 已定义的地址；具有多个

功能的 ECU，可以使用多个地址，也可以重新定义新的地址；新定义地址，应使用 208~231 这段属于公路用车的预留地址，报文编号为分配给每个结点的能进行目的寻址的报文编号空间。

结点名称	结点地址
仪表 (METER)	23 (0x17)
整车控制器 (VCU)	208 (0xD0)
电机控制器 (MCU) 1	239 (0xEF)
电机控制器 (MCU) 2	240 (0xF0)
电机控制器 (MCU) 3	241 (0xF1)
电机控制器 (MCU) 4	242 (0xF2)
电池管理系统 (BMS) 1	243 (0xF3)
电池管理系统 (BMS) 2	244 (0xF4)
电池管理系统 (BMS) 3	245 (0xF5)
电池管理系统 (BMS) 4	246 (0xF6)
全局 (全部-任何节点)	255 (0xFF)

三、 报文

1、 整车控制器发送报文一

OUT	IN	ID (0x0C01EFD0)						周期 (ms)
VCU	MCU	P	R	DP	PF	PS	SA	50
		3	0	0	1 (0x01)	239 (0xEF) (注 1)	208 (0xD0)	
数据								
BYTE	BIT	数据名		分辨率		偏移量	取值范围	
0		目标相电流 (目标转矩)		0.1A/bit		-3200A	-3200~3200A	
1								
2		目标转速		1rpm/bit		-32000rpm	-32000~32000rpm	
3								
4	0	控制命令				0	0: 停机	
	1						1: 运行	
	7-2						0: 转矩控制模式 1: 转速控制模式	
		保留					保留	
5		保留						
6		保留						
7		生命信号				0	0~0xFF	

2、 电机控制器发送报文一

OUT	IN	ID (0x1801D0EF)						周期 (ms)
MCU	VCU	P	R	DP	PF	PS	SA	50
		6	0	0	1 (0x01)	208 (0xD0)	239 (0xEF) (注 1)	
数据								
BYTE	BIT	数据名		分辨率		偏移量	取值范围	

0		母线电压	0.1V/bit	0	0~300V
1					
2		母线电流	0.1A/bit	-3200A	-3200~3200A
3					
4		相电流	0.1A/bit	-3200A	-3200~3200A
5					
6		转速	1rpm/bit	-32000rpm	-32000~32000rpm
7					

3、电机控制器发送报文二

OUT	IN	ID(0x1802D0EF)						周期(ms)	
MCU	VCU	P	R	DP	PF	PS	SA	50	
		6	0	0	2(0x02)	208(0xD0)	239(0xEF) (注1)		
数据									
BYTE	BIT	数据名			分辨率	偏移量	取值范围		
0		控制器温度			1℃/bit	-40℃	-40~210℃		
1		电机温度			1℃/bit	-40℃	-40~210℃		
2	0	状态位				0	0: 停机		
	1						1: 运行		
	7-2						保留		
3	0	故障位				0	0: 正常 1: 故障		
	1								过流
	2								过载
	3								过压
	4								欠压
	5								控制器过温
	6								电机过温
	7								电机堵转
4	0	故障位				0	0: 正常 1: 故障		
	1								电机主传感器
	2								电机辅传感器
	3								编码器对齐
	4								上电防飞车
	5								主加速器
	6								辅加速器
	7								预充电
5	0	故障位				0	0: 正常 1: 故障		
	1								功率管
	2								电流传感器
	3								自学习
	4								RS485
							CAN		

	5		软件		
	7-6		保留		
6			保留		
7		生命信号		0	0~0xFF

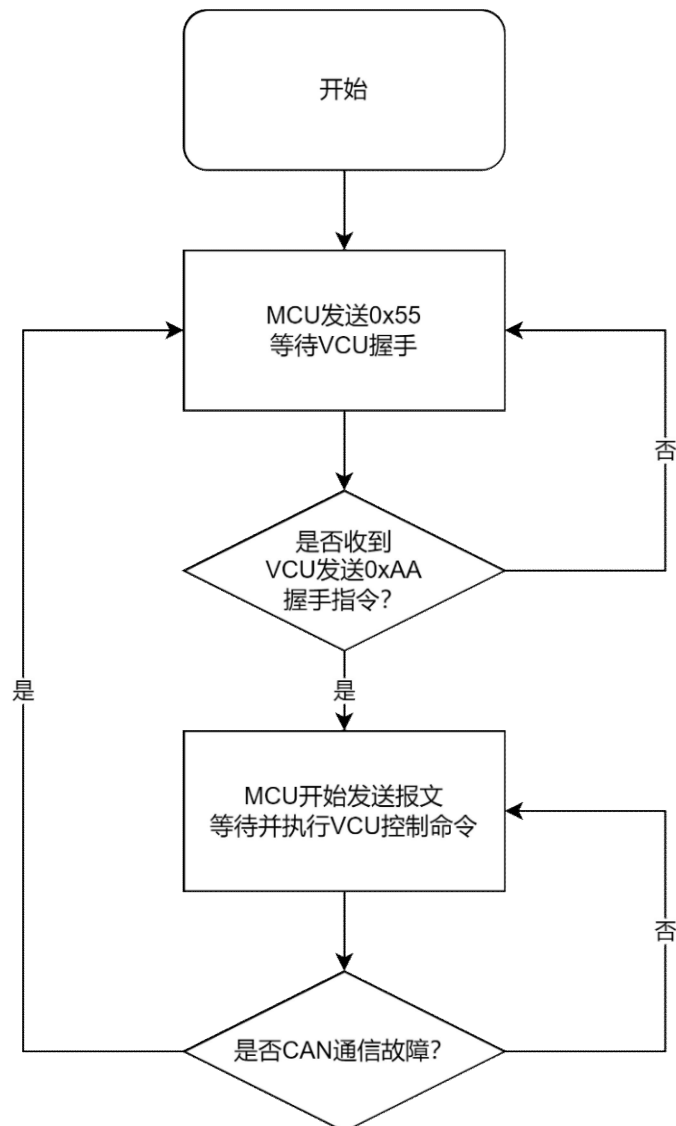
注 1:SA 可以通过 MCU 上位机设置,设置参数:控制器编号,默认 SA=控制器编号=239(0xEF)

四、握手

MCU 上电后,循环发送 8 个字节 0x55, ID=0x1801D0EF(注 1), 周期 50ms, 等待 VCU 握手。VCU 收到 MCU 发送等待握手指令后, 发送 8 个字节 0xAA, ID=0x0C01EFD0(注 1), 与 MCU 握手。

MCU 收到 VCU 发送握手指令后, 握手成功。MCU 开始发送报文 (电机控制器发送报文一和二)。等待并执行 VCU 控制命令 (整车控制器发送报文一)。

MCU 连续 10 次收不到 VCU 控制命令 (整车控制器发送报文一), 或者连续 5 次收到 VCU 控制命令的生命信号不对, 认为 CAN 通信故障, MCU 会停机并重新执行握手流程。



MCU 握手流程图

五、 代码示例

示例基于 STM32F4 HAL 库，用于 VCU 设计参考

1、 VCU 发送握手指令

ID=0x0C01EFD0(注 1)

示例：

```
void VCU_SendHandshake(void)
{
    CAN_HandleTypeDef *CANxHandle = &hcan1;
    CAN_TxHeaderTypeDef CAN_TxHeader;
    uint32_t TxMailbox;
    uint8_t TxBuf[8];
    uint8_t Index;

    for (Index = 0; Index < 8; Index++)
    {
        TxBuf[Index] = 0xAA;
    }

    CAN_TxHeader.ExtId = 0x0C01EFD0;
    CAN_TxHeader.IDE = CAN_ID_EXT;
    CAN_TxHeader.RTR = CAN_RTR_DATA;
    CAN_TxHeader.DLC = 8;
    CAN_TxHeader.TransmitGlobalTime = DISABLE;
    HAL_CAN_AddTxMessage(CANxHandle, &CAN_TxHeader, TxBuf, &TxMailbox);
}
```

2、 VCU 发送控制命令（整车控制器发送报文一）

发送周期 50ms

ID=0x0C01EFD0(注 1)

示例：

```
uint16_t TargetPhaseCurrent01A;
uint16_t TargetSpeedRPM;
uint8_t ControlCmd;
uint8_t LiveCounter;
void VCU_SendCommand(void)
{
    CAN_HandleTypeDef *CANxHandle = &hcan1;
    CAN_TxHeaderTypeDef CAN_TxHeader;
    uint32_t TxMailbox;
    uint8_t TxBuf[8];

    TxBuf[0] = (uint8_t)(TargetPhaseCurrent01A);
    TxBuf[1] = (uint8_t)(TargetPhaseCurrent01A >> 8);
    TxBuf[2] = (uint8_t)(TargetSpeedRPM);
```

```
TxBuf[3] = (uint8_t)(TargetSpeedRPM >> 8);
TxBuf[4] = ControlCmd;
TxBuf[5] = 0;//Reserved
TxBuf[6] = 0;//Reserved
TxBuf[7] = LiveCounter++;

CAN_TxHeader.ExtId = 0x0C01EFD0;
CAN_TxHeader.IDE = CAN_ID_EXT;
CAN_TxHeader.RTR = CAN_RTR_DATA;
CAN_TxHeader.DLC = 8;
CAN_TxHeader.TransmitGlobalTime = DISABLE;
HAL_CAN_AddTxMessage(CANxHandle, &CAN_TxHeader, TxBuf, &TxMailbox);
}
```