FIRMWARE MANUAL

DlaBoard_B

DlaBoard_B · 出厂固件 · 使用说明书（含 CAN 协议全文）

副标题	DlaBoard_B · 出厂固件 · 使用说明书（含 CAN 协议全文）
品牌	Daliang Auto
产品型号	DlaBoard_B（硬件已定版，原理图 2026-08-01）
固件版本	0.1.17（构建标识 5eaa56e，Release 构建，2026-11-11）
镜像校验	63156 字节，CRC32 69e61b5f（镜像自述头已盖章）
文档版本	Rev. 1.10
发布日期	2026-11-11
控制台	USB1（USART1），115200 8N1，CR/LF
整车总线	经典 CAN，1 Mbps，标准帧，DLC = 8

本书与规格书的分工

- 《DlaBoard_B 产品规格书》描述硬件：电气参数、引脚、连接器、I/O 资源、机械尺寸。硬件不改版，规格书不改版。
- 本书描述出厂固件：整车控制逻辑、CAN 协议、串口协议、诊断命令、参数、升级方式。固件每发布一版，本书随之修订一版。
- 两本书中均出现的量（供电范围、引脚、连接器针序），以规格书为准。

注意：本书将已验证与未验证的功能分开描述，阅读时请勿混同。本板处于整车集成阶段，部分功能已实现但尚未在整车上验证。各项验证状态见 §13，未验证的功能不构成保证。

警告：本板不含功率级，不驱动电机，也无栅极驱动。本板输出的是总线上的行驶请求，由电机驱动器执行。本板的急停由软件实现，见 §11 安全申报。

---

1. 固件概述

DlaBoard_B 在整车中负责确定驾驶意图：读取踏板、档位、整车使能、刹车开关、遥控和车载计算机的输入，将相互冲突的输入仲裁为一份行驶请求，每 20 ms 通过 CAN 发送给电机驱动器。

整车不允许驱动的原因共有十四条，外部现象相同（车辆不动）。固件将各条原因分开上报，见 §7。

1.1 四条对外通路

通路	对端	物理口	章节
整车 CAN	电机驱动器（一块双电机板，或两块单电机板）	CN34 / CN32	**§5（专章）**
车载计算机串口	Jetson 等自动驾驶计算机	**USB2**（UART5）	§6
控制台 / 上位机串口	操作人员，或 PC 上位机软件	**USB1**（USART1）	§2 – §3
触摸屏	车上的仪表与按键	**CN35**（USART2，9600）	§4

板上各连接器在出厂固件中的用途见 §1.3。该节同时列出固件不使用的接口。

注：两个 USB 口功能不同，不可互换。USB1 只传输 ASCII 文本，USB2 只传输与车载计算机通信的二进制帧。任何模式下控制台均可使用。

1.2 六种模式

模式	驾驶来源
`MANUAL`	驾驶员：踏板 + 档位 + 整车使能
`REMOTE`	遥控器
`TRAINING`	示教录制
`COLLECT`	采集（SLAM）：行为同示教，另上报采集帧与 IMU
`FOLLOW`	车载计算机跟随（`0x10` 指令帧）
`AUTO`	车载计算机自动驾驶（`0x11` 指令帧 + 速度闭环，见 §6）

模式由触摸屏按键切换。任何模式下，§7 的十四条阻止条件均保持有效。

1.3 接口功能定义

《产品规格书》给出连接器与 MCU 引脚的对应关系。本节说明两项内容：出厂固件对各接口的用途，以及各接口收发的信号和格式。

针序、间距、板载分压、电平耐压见规格书（§7 通信接口、§8 通用 I/O、§11 连接器一览表），本节不重复引脚映射。

> 注：本节内容随固件版本变化，规格书中上述章节不随固件变化。

1.3.1 接口用途总表

连接器	MCU 资源	出厂固件的用途	收发内容
USB1	USART1	控制台 / 上位机	ASCII 文本行，115200 8N1，CR/LF。见 §1.3.2
USB2	UART5	车载计算机	22 字节定长二进制帧，115200 8N1。见 §6
CN34 / CN32	CAN1（PB9 / PB8）	整车总线	经典 CAN 1 Mbps，标准帧，DLC = 8。见 §5
CN35	USART2（PD5 / PD6）	触摸屏	DGUS / VGUS 变量帧，**9600 8N1**。见 §1.3.2
U7（5P）	PD0 / PD1 / PD3 / PD4	PRND 档位开关	四路干接点，低有效。脚 1 = 公共端接 GND，脚 2 = P、3 = R、4 = N、5 = D，见 §4.2
CN22	INPUT_PD10	整车使能开关	一路干接点，低有效。断开时不允许驱动，同时作为一路刹车意图
CN5 / CN6	PE2 / PE3	刹车开关，两路	干接点，低有效。两路为「或」关系，任一路闭合即为刹车意图（见 §1.3.4 第一条）
CN30	ADC1_IN8 / PB0	油门踏板	0 – 5 V 直流模拟量，见 §1.3.2
H2 · RC1	PC8 / TIM3_CH3	遥控转向	舵机 PWM。本板解析并回报，但转向字段恒发 0，见 §5.10
H2 · RC2	PC9 / TIM3_CH4	遥控油门	舵机 PWM。`REMOTE` / `TRAINING` / `COLLECT` 下有效
H2 · RC3	PC7 / USART6_RX	iBUS 串口遥控（默认）	六路中唯一接 USART 接收引脚的通道。`rc_source=0` 时该通道不可用
H2 · RC4	PC6 / TIM3_CH1	远程急停（`rc_source=0` 的 PWM 模式）	舵机 PWM 作两位开关使用，> 1500 µs = 刹车
H1	SWD	烧录 / 在线调试	见规格书 §10.1

以下板载器件不是对外连接器，但由固件使用，一并列出：

器件	接口	出厂固件的用途
6 轴 IMU	SPI1	`COLLECT` 模式的采集帧与 `IMU?`。初始化失败不阻塞主流程，IMU 字段填 0
EEPROM 8 KB	I2C1	参数持久化，双区轮换 + CRC32，见 §8
输入电压检测	ADC1_IN11 / PC1	`tlm` 中的板供电压
用户 LED	PE0	上电点亮后不再变化，不用于状态指示
蜂鸣器	PE1	出厂固件运行时不鸣响
用户按键	PC13	出厂固件不读取

1.3.2 各接口信号定义

油门踏板（CN30）：0 – 5 V 直流模拟量

项目	内容
输入信号	0 – 5 V 直流电压。电位器式或霍尔式踏板均可，输出量程须为 0 – 5 V
连接器	2.0 mm 3P：1 = GND / 2 = 信号 / 3 = +5 V（取自板上 5 V 轨，可直接为踏板供电）

项目	内容
板上前端	10 kΩ 串阻 + 20 kΩ 对地 + 100 nF，5 V 输入对应 ADC 端约 3.33 V
读取方式	12 位 ADC（0 – 4095），每个主循环周期采样一次，按两个端点线性映射为 0 – 1000‰
两个端点	参数 `thr_adc_dead`（松开点，默认 850）、`thr_adc_full`（踩到底，默认 3500），单位为 ADC 计数。`calib start pedal` 用于测定这两个值，见 §9
回位死区	参数 `thr_dead`（默认 30‰），小于该值按 0 处理
读数低于松开点，或两个端点设反	按未踩下处理（0‰），不输出比例值
接线确认	`tlm` 的 `veh` 行包含踏板‰，踩下踏板时该值应变化；`selftest run` 的 `pedal_adc` 报告原始 ADC 计数

> 注意：踏板信号线断开时固件不报错。信号脚断开后，板上 20 kΩ 对地电阻将 ADC 输入拉至 0 V，读数与松开踏板相同。车辆不会因此加速，但断线无法由固件检测，现象与驾驶员未踩踏板相同。

档位、整车使能、刹车（U7 / CN22 / CN5 / CN6）：干接点

项目	内容
输入信号	无源触点：机械开关、继电器触点、行程开关。闭合接 GND 为有效
有效电平	低有效。板上串 100 Ω，MCU 内部上拉已开启，未接线的引脚稳定读高
外部电压	不得从外部施加电压。这几路按干接点设计，接开关即可。GPIO 为 3.3 V，电平耐压见规格书 §13
触点要求	档位四路默认按**四位置保持型**开关接线（当前档位对应的一路持续闭合）。挡杆若是自复位（点动）型，固件 0.1.3 起支持：参数 `gear_sw_type` 置 1 后改按边沿门锁解码；两种开关的断线失效方式相反，选型判据见 §4.2。刹车与使能使用普通开关
去抖	档位：解码结果稳定 **50 ms** 后生效，上电后 **500 ms** 内不解码。使能与刹车无软件去抖，直接跟随触点状态
断线行为	档位（默认保持型）：四路全高，无法解码，落 P。档位（`gear_sw_type=1` 自复位）：断线不产生边沿，档位停在原地，无任何指示（见 §4.2）。使能：读高，不允许驱动。刹车：读高，无刹车意图，该路断线不会被检测到

> 注意：刹车开关线路断开时读为未刹车，固件无法检测该故障。

遥控：两种模式，二选一

iBUS 与 PWM 是两种互斥的模式，不能同时使用。使用 iBUS 时，RC3 引脚（PC7）用作串口接收，该通道没有 PWM 脉宽；iBUS 单线传输全部通道，其余五个引脚不使用。

模式由参数 `rc_source` 选择：`0` = PWM，`1` = iBUS。出厂默认 iBUS。当前模式可从 `RC?` 输出中的 `src=ibus` 或 `src=pwm` 确认。

iBUS 模式（默认）

项目	内容
接口	H2 的 RC3（信号排从 RC6 一端数第 4 个）。六路中唯一接 USART 接收引脚的通道
协议	FlySky iBUS，115200 8N1，不反相，每帧 32 字节，周期约 7 ms。通道值 1000 – 2000，与舵机脉宽量纲相同
SBUS	不支持。SBUS 为反相信号，STM32F4 的 USART 无 RX 反相功能，使用 SBUS 须外接反相器

项目	内容
通道分配	ch1 转向、ch2 油门、ch5 油门限速、ch7 远程急停、ch6 转向限速
参考配置	富斯 i6X 设为 10 通道时：ch1/ch2 为右摇杆，ch5/ch6 = VrA/VrB 旋钮，ch7 = SwA 二位开关
坏帧	校验和错误或通道值越界时丢弃整帧，不使用部分数据。坏帧数见 `RC?` 中的 `err=`
查看方式	`RC?` 报告前八路原始值、帧数、坏帧数和链路三态

油门通道

油门使用 ch2（右摇杆上下）。该摇杆自动回中，中位对应停车。ch3（左摇杆上下）为带阻尼、不回中的油门杆，松手后不减速，上电时位置也不确定，因此不用作油门。

两个限速旋钮

旋钮	作用	不限速对应的值
ch5 (VrA)	驱动油门上限	**1000 = 不限**；0 表示车辆不行驶
ch6 (VrB)	转向速度上限	**0 = 使用转向板自身的默认上限**（即最快），因此旋钮拧到底时夹到 1

注意：两个旋钮「不限速」对应的值不同，且方向相反。转向侧 0 的含义由 CAN 协议规定。两者设反时，驱动侧车辆不行驶，转向侧方向机以全速运行。

PWM 模式（`rc_source = 0`）

项目	内容
输入信号	常规接收机的 PWM 脉冲，约 50 Hz（每帧 20 ms）
脉宽	1000 – 2000 µs，**中位 1500 µs**。固件测量正脉宽（上升沿到下降沿）
接线	H2 为 3 × 6 排针，每个通道一组 信号 / +5 V / GND，接收机舵机线直接插接
使用通道	RC1 转向、RC2 油门、RC4 远程急停。PWM 模式无限速旋钮
RC3	不可用，该引脚用作串口接收
急停判据	**> 1500 µs = 刹车**。须使用自保持位置的两位置开关通道；使用自复位按钮时，松手即解除刹车

两种模式共用的部分

项目	内容
失效判据	**200 ms** 无新帧判为失效（参数 `rc_stale`，可设 50 – 2000 ms）
三态	`NEVER`：从未接入，不产生任何主张；`UP`：在线；`LOST`：曾在线、现已断开，摇杆按中位处理，`REMOTE` / `COLLECT` 下强制刹车。见 §4.4
未选用的模式	不产生任何主张，包括刹车。因此使用 iBUS 时，未接的 PWM 线不会导致车辆保持刹车

注：iBUS 链路断开后，内存中保留最后一帧的通道值，仅凭通道值无法判断链路是否在线。固件以帧计数和时间戳判定链路状态。

注意：接收机在遥控器关机后的行为因型号而异。富斯 i6X 配套接收机在遥控器关机后停止输出，控制器 200 ms 后判为 LOST 并强制刹车。部分接收机在遥控器关机后继续输出失控保护值，此时控制器判定链路正常。更换接收机后须重新确认其行为；对于关机后继续输出的接收机，须在接收机中将急停通道的失控保护值设为刹车位。

触摸屏（CN35）：DGUS / VGUS 变量帧

项目	内容
串口参数	**9600 8N1**，与两个 USB 口的 115200 不同
帧格式	`A5 5A <LEN> <CMD> <ADDR_HI> <ADDR_LO> <数据…>`。`LEN` 从 `CMD` 字节起计，多字节量为大端，无校验和
板 → 屏	`CMD = 0x82` 写变量，每次一个 16 位字。文本控件写入 UTF-8 字节并以 `FF FF` 结尾（用于清除控件中的残留字符）
屏 → 板	`CMD = 0x83` 变量回读：屏上报地址与键值，固件取第一个字作为键值
屏发送的变量	模式 `0x05A1`、档位 `0x05E1`、速度档 `0x05C1`
板发送的变量	车速 `0x02E5`、连接状态 `0x05E6`、电池 `0x05EB`、母线电压 `0x05EC`（0.1 V/字）、母线电流 `0x05F0`（0.1 A/字，含符号）、整车使能 `0x0608`（文本，使能时显示 `R`）、采集确认 `0x0618`、存图状态 `0x0619`
模式键值	手动 `0x0010` / 遥控 `0x0008` / 示教 `0x0004` / 跟随 `0x0002` / 自动 `0x0001` / 采集 `0x0020`
档位键值	P `0x0008` / R `0x0004` / N `0x0002` / D `0x0001`
速度档键值	极低 `0x0008` = 35% / 低 `0x0004` = 60% / 中 `0x0002` = 80% / 高 `0x0001` = 100%

> 注意：仅在 U7 未接时，触摸屏可以设置档位。从 U7 解码出一次有效档位后，档位以实体开关为准，触摸屏不再能设置档位。详见 §4.2。
> 注意：该通路无校验和，也无超时重发。触摸屏未连接、波特率错误、线序接反时，现象均为屏幕显示不变化。

整车 CAN（CN34 / CN32）与两个 USB 串口

口	收发内容	详见
CN34 / CN32	经典 CAN，**1 Mbps**，标准帧，DLC = 8。本板发送 `0x210`（20 ms），接收 `0x211` / `0x212`（拓扑 B 另加 `0x213` / `0x214`）。120 Ω 终端由 SW3 投切	§5（逐字节定义）
USB1	仅 ASCII 文本：每行一条命令，以回车换行结束；回复为文本行	§2 – §3
USB2	仅二进制：22 字节定长帧，小端，`SOF = AA 55`，CRC16-MODBUS 覆盖第 2 – 19 字节	§6

1.3.3 出厂固件不使用的接口

以下接口硬件已连通，出厂固件不使用，可直接用于二次开发（规格书 §17.3）。

连接器	MCU 资源	状态
CN36	USART3（PC10 / PC11）	引脚已配置为串口复用，但固件不初始化该串口，也不收发
CN33	I2C2（PB10 / PB11）	不初始化
H3	SPI2（PB12 – PB15）	不初始化
CN20 / CN19	PA2 / PA3，TIM2_CH3 / CH4	轮速脉冲：固件对脉冲计数，但计数值无输出途径，不进入 `tlm`、参数表或 CAN，从外部看相当于未接。车速取自驱动板的 `0x212`
CN3 / CN4	TIM1_CH1 / CH2（PA8 / PE11）	PWM 已启动，但占空比恒为 0。行走电机由驱动板经 CAN 控制，本板不输出 PWM
CN7 / CN10	TIM1_CH4 / CH3（PE14 / PE13）	未配置为 PWM 通道

连接器	MCU 资源	状态
CN13 / CN14	TIM2_CH1 / CH2（PA0 / PA1）	捕获通道未启用
CN24 / CN25 / CN26 / CN29 / CN31	ADC1 另外五路	不采样。踏板固定使用 CN30，改用其他通道须重新编译
CN8 / CN9 / CN11 / CN12 / CN17 / CN18 / CN21	PE4 / PE7 / PE8 / PE9 / PE10 / PE12 / PE15	数字输出，上电置低后不再变化
CN15 / CN16 / CN23 / CN27 / CN28	PD13 / PD12 / PD11 / PD9 / PD8	数字输入，固件不读取
H2 · RC5 / RC6	PD14 / PD15，TIM4_CH3 / CH4	捕获通道未启用

1.3.4 接线注意事项

一、CN5 / CN6 在规格书中列为数字输出，出厂固件将其用作输入。 规格书 §8.1 将 PE2 / PE3 列入九路数字输出，这是硬件层面的描述：两个引脚经 100 Ω 串阻引到插座，可用作输出，也可用作输入。出厂固件在初始化的最后一步将其重新配置为内部上拉输入，用于连接刹车开关，踩下时接 GND。

> 注意：接线以本节为准。按数字输出接入负载时，该引脚为上拉输入，负载不会动作，且刹车开关无处连接。如需将其用作输出，须修改固件。

二、CN35 的波特率为 9600。 触摸屏通路的波特率与两个 USB 口（115200）不同。用串口工具监测该通路时须按 9600 打开，否则显示乱码，现象与触摸屏未连接相同。

---

2. 连接控制台

物理口	**USB1**（USART1）
串口参数	**115200 8N1，CR/LF**
内容	仅 ASCII 文本：输入命令，输出回复

2.1 打开串口前的注意事项

本板的 USB 转串口电路带有交叉耦合的复位电路。单一静态电平不会引起复位，特定的电平组合会使板子进入 ROM BootLoader：

操作	结果
`{DTR=1, RTS=0}` 打开串口	应用正常运行
`{DTR=0, RTS=0}` 打开	不复位
`{DTR=1, RTS=1}` 打开	不复位
RTS 为低时 DTR 跳变	不复位
RTS 为高时 DTR 跳变	**应用停止运行，板子进入 ROM BootLoader**

推荐以 `{DTR=1, RTS=0}` 打开串口。常见串口工具的默认设置不会使本板复位；RTS 拉高后再改变 DTR 会使本板进入 BootLoader。

警告：装车使用时，本板进入 BootLoader 后停止发送行驶请求，驱动器 300 ms 后超时并停止输出力矩。现场表现为车辆停止，无法从外部看出板子已进入 BootLoader。

恢复步骤（须严格按以下顺序执行，否则不会产生复位）：

```
{DTR=1, RTS=1}   保持 50 ms
{DTR=0, RTS=1}   保持 150 ms     ← 按住复位，RTS 必须还是高的
{DTR=1, RTS=0}   保持 300 ms     ← 松开复位，此时 BOOT0 已经是低
```

注意：不得先将电平恢复为安全值再执行复位。RTS 为低时 DTR 跳变不产生复位，板子将保持原状态，易被误判为硬件损坏。

2.2 上电后的基本查询

```
ver          固件版本、构建标识、镜像校验、MCU 唯一 ID
tlm          一次遥测快照（六行，机器可读）
WHY?         现在为什么不允许驱动
ST?          驱动器链路状态、回环三态、各类计数器、复位原因
```

`ver` 的输出示例（0.1.1 起为 Release 构建，镜像已盖章）：

```
=tlm 1 build debug=0 stamped=1 crc=c77a92cb major=0 minor=1 patch=1 board=dlaboard_b scm=45568e0 cmdhash=...
DlaBoard_B 0.1.1 45568e0 built ...
firmware : 0.1.1 45568e0, crc ok
uid      : <24 位十六进制>
```

`id` 以一行文字说明板卡身份，供上位机自动识别。

2.3 遥测快照 `tlm`

共七行，以 `=tlm end` 结束，全部为 `键=值` 格式，单位包含在键名中：

行	内容
`build`	版本、构建、盖章、命令表校验
`sys`	运行毫秒、复位原因、看门狗、HardFault 记录
`veh`	模式、档位、整车使能、刹车意图、踏板‰、速度档%、阻止位图
`spd`	车速 0.1 km/h、车速是否有效、行驶方向、母线电压 mV、母线电流 cA
`can`	链路三态（alive / frozen / believes）、健康字节、收发计数、发送失败、回显不符、短帧、FIFO 溢出、bus-off、刹车驱动位、拓扑
`steer`	转向执行器侧（0.1.4 起）：`0x231` 收帧数、链路是否在线、帧龄、回显序列号、当前位置、状态位、故障码
`rc`	遥控链路三态、串口丢字节计数、转向拦截计数。注：`rc3=` 字段默认报告 iBUS 链路

> 注：每个主循环周期输出一行。上位机读取完整快照时，超时须设为 3 秒以上。

---

3. 命令一览

板上无 `help` 命令。命令表随发布包提供（`COMMANDS.json`，43 条），上位机据此生成界面。板子在 `=tlm build cmdhash=` 中报告该表的 CRC32，与发布包一致时方可使用。

组	命令
认板 / 遥测	`ver`（`VER?`）、`id`、`tlm`（`TLM?`）、`IMU?`、`JET?`

组	命令
诊断	`WHY?` 、 `ST?` 、 `FOC ST?` 、 `GEAR?` 、 `RC?` 、 `BRK?` 、 `app` 、 `fault` 、 `fault codes`
参数	`param list` （ `PARAM?` ） 、 `param get <名>` 、 `param set <名> <值>` 、 `param save` 、 `param load`
标定	`calib list` 、 `calib status` 、 `calib start pedal` 、 `calib confirm` 、 `calib abort`
自检	`selftest run` 、 `selftest run all`
安全	`estop` 、 `clear`
台架驱动	`FOC D <%> [转向%]` 、 `FOC R <%> [转向%]` 、 `FOC RAW ...` 、 `FOC STOP` 、 `FOC OFF` 、 `BRAKE 1` 、 `BRAKE 0` 、 `BRAKE OFF`
台架注入	`FOC RX211 <8 个十六进制字节>`
台架转向	`STEER <位置 mm> <速度 1-100>` 、 `STEER OFF`

> 警告： `FOC D` / `FOC R` / `FOC RAW` 会使车辆行驶。这些命令由控制台接管 `0x210` ，并在 `WHY?` 中置位 `cli_test` 。台架命令 300 ms 未刷新即自动失效。
> 警告： `STEER` 会使转向执行器上电并移动齿条，发送前须确认齿条行程内无人员和障碍物。

`STEER <位置 mm> <速度 1-100>` 通过 `0x230` 向转向执行器发送目标位置和限速，例如 `STEER -12.5 30` 。

- 位置单位为 mm，可带一位小数，与 `tlm` 的 `steer` 行 `mm10` 同一基准（ `mm10` = mm × 10）。超出标定行程时由执行器限位。
- 速度为执行器配置速度上限（ `st_vmax_mm_s` ）的百分比，必须填写，范围 1–100。
- 仅在 MANUAL 模式下生效。其他模式的转向目标由遥控或车载计算机给出，命令被拒绝。
- 一条命令保持 5 s，需要持续保持时在 5 s 内重发。到期后齿条释放。
- 切换模式、断开整车使能、链路丢失闩锁或急停时立即释放，条件恢复后不会自动继续，须重新发送。
- `STEER OFF` 立即释放。
- 成功回复 `=ok steer ...` ；无法执行时回复 `=err steer <原因>` ，原因包括急停、模式、整车使能、闩锁、执行器无响应、执行器故障和未标定。

> 注：执行器有到位死区，齿条停止后的位置与目标最多可相差约 1 mm，此时 `onpos` 仍为 1。本命令已在接有转向执行器和前轮的台架上验证（车轮悬空，小幅移动），尚未在车轮着地的条件下验证。

3.1 `app` ： 应用层（0.1.9 起）

自 0.1.9 起，出厂固件分为核心层和应用层。用户可通过修改应用层改变整车驾驶行为，无需改动安全相关部分。

层	内容	可否修改
核心层	输入采集、安全判断（允许驱动与强制刹车的条件）、CAN 链路、看门狗、参数存储	否
应用层 `App/apps/<名字>/`	各路输入到油门、转角、速度档的换算；模式切换；多节点编排	是，供用户修改

源码位于发布包的 `source.zip` 中（出厂应用为 `App/apps/a1_diff2/` ，附 README）。修改后无需连接车辆即可验证：

```
sh tests/sim/run.sh          喂一串输入，断言目标随时间怎么走
sh tests/run.sh              单元测试
```

在板上查询应用当前的计算结果：

```
> app
APP name=a1_diff2 api=1 brake=0 act=1 band=60 out=60 clamp=0 L=350/0* R=350/0* steer=0/0*
```

字段	含义
`name` / `api`	当前编入的应用名称，及其对应的接口版本
`brake` / `act`	应用当前的刹车意图 / 应用判定是否应当行驶。该值仅为应用的主张，是否允许驱动由核心层决定
`band`	应用主张的速度档；`0` 表示不主张，使用核心层的计算值
`out`	实际发送到 CAN 上的速度档，见下
`clamp`	应用输出越界值的次数。持续不为 0 表示应用的计算与量程不匹配
`L=` / `R=` / `steer=`	各节点的 `牵引‰/转角‰`，`*` 表示本周期应用未主张

> 注意：当前版本中，应用层只有速度档一个字段实际输出到总线（0.1.10 起）。`out=` 即本帧 `0x210` 发出的速度档字节。油门、挡位、转向仍由核心层计算，`app` 中对应的值仅为应用的主张。修改应用可以改变速度档，不能改变油门。
> 注：控制台接管期间（`FOC RAW` / `FOC D` / `FOC R`）不经过应用层。这些命令绕过整车仲裁，直接发送指定内容。

规则：应用输出 0 时，使用核心层的计算值。出厂应用将核心层给出的速度档原样返回；将其改为其他值后，总线上的速度档随之改变。

3.2 基于开发模板的二次开发（0.1.11 起）

§3.1 说明如何修改本车的驾驶行为。本节说明如何将本板用于其他项目。

本板是一块通用整车控制器：一颗 STM32F407、一路 CAN、三路串口、一组数字量输入和继电器输出、板载 EEPROM 和一颗六轴 IMU。出厂固件是本板的一个典型应用示例。开发其他设备时，可从开发模板开始。

3.2.1 模板内容

发布包的 `source.zip` 为同一套源码树，提供两种编译配置：

配置	编译结果
出厂固件（默认）	本书 §4 起描述的全部功能：整车仲裁、档位、遥控、触摸屏、驱动器和转向相关帧
开发模板	仅完成板级初始化：时钟外设、串口控制台、认板、参数存储、独立看门狗、镜像自述头。主循环由用户实现，不含整车逻辑

模板包含：

- 串口控制台，以及 `ver` / `id` 两条认板命令（上位机据此识别板卡）
- 参数存储：板载 EEPROM，双区轮换 + CRC32 + 回读自验（§8）
- 独立看门狗（2 秒），以及上一次复位原因的记录
- 镜像自述头（上位机据此识别镜像）
- `param list` / `get` / `set` / `save` / `load` 五条命令，输出与出厂固件完全一致
- CAN 外设已初始化，收发由用户实现

模板不包含：整车仲裁、档位解码、遥控、触摸屏、六轴 IMU、轮速，以及驱动器和转向执行器的通信帧。需要其中某项时，可从出厂固件源码中移植对应文件。

3.2.2 编译与烧录

```
cmake --preset Release-minimal
cmake --build --preset Release-minimal
```

生成可直接烧录的 `.bin` / `.hex`（已盖章）：

```
DLA_PROFILE=minimal DLA_CONFIG=Release sh scripts/build.sh
```

产物输出到 `Release-minimal/`。

> 注意：模板产物与出厂固件产物须分开存放，不得放在同一目录。两份不同的镜像同名放在一起时，烧错不会产生任何报错，板子将运行另一份固件。

用户代码写在 `App/core/app_minimal.c` 的两个函数中：`DlaCart_AppInit()`（上电执行一次）和 `DlaCart_AppLoopIteration()`（主循环中反复调用）。

> 注意：不得在主循环中阻塞。看门狗超时为 2 秒，喂狗须等上述函数返回后才会执行。需要按固定周期执行的任务，应记录时间戳进行判断，不使用延时函数。

板上引脚，以及出厂固件未使用的引脚，见随包的 `pins.yaml`。

注意：`pins.yaml` 中的「未使用」仅指出厂固件未使用，不代表可任意使用。调试接口和晶振相关引脚被改动后，将无法烧录固件。

3.2.3 不可删除的内容

以下三项是模板与上位机配合的前提。删除其中任一项，板卡将无法被上位机识别，且不会有任何提示指出原因：

项目	删除后果
看门狗	程序跑飞后，输出保持在跑飞时的状态继续运行，且不会报错。这是本板最严重的一种失效
`ver` / `id` 两条命令	上位机按这两条命令的输出文字识别板卡。改动一个字，板卡即被识别为已擦除的空板：面板上无板名，型号选择无法保存，烧录和遥测功能随之不可用
镜像自述头	上位机按自述头识别镜像。相差一个字节即无法识别，且无提示：上位机改用整文件校验，校验显示通过，损坏的镜像也可被烧入

3.2.4 受保护文件清单

随包的 `template.yaml` 列出修改前须谨慎评估的文件，分为两组，两组的影响程度不同：

组	含义	示例
板级安全	修改后板卡可能出现故障，或故障无法被发现	看门狗；处理器故障后、复位前的挂点；参数存储的读写和记录布局
工具链契约	修改后板卡在上位机工具链中无法识别	`ver` / `id` 两条命令；build 记录行；镜像自述头；参数表格式；板号；编译预设名称

名单以外的文件默认可以修改。列入名单的依据是修改错误后不会被立即发现；修改错误会当场报错的文件不列入名单。

> 警告：处理器故障（程序跑飞、访问非法地址）发生后、板子复位前，固件会调用一个挂点函数，供用户停止执行机构。模板中该函数的默认实现为空。用户驱动电机、继电器、气阀或其他板卡时，须实现该函数：在任一源文件中定义同名函数即

> 可。未实现时不会报错，板子照常复位，但从故障发生到复位期间，输出保持故障前的状态。实现方法见 `App/Inc/board_isr_hooks.h` 文件头。

---

4. 整车控制逻辑

4.1 输入

输入	连接器	说明
踏板	**CN30**（ADC1_IN8）	端点由 `calib start pedal` 标定（§9），输出 0..1000‰
档位	**U7** 档位开关（默认四位置保持型；自复位型见 §4.2）	**脚 2=P、脚 3=R、脚 4=N、脚 5=D**（脚 1=GND）
整车使能	**CN22**（低有效）	断开时不允许驱动
刹车开关	**CN5 / CN6**（低有效，两路）	任一路闭合即为刹车意图
遥控	**H2**（默认仅使用 RC3，iBUS）	iBUS：ch1 转向、ch2 油门、ch5/ch6 两个限速旋钮、ch7 急停。PWM 模式：RC1/RC2/RC4
速度档位	触摸屏（**CN35**）	0..100%，随 `0x210` 下发
车载计算机	**USB2** 二进制帧	FOLLOW / AUTO 模式

> 各路输入的信号、格式及断线行为见 §1.3.2；固件不使用的接口见 §1.3.3。

4.2 档位解码

四路输入在每个周期解码为一个档位。多路同时为低（短路或先通后断）在两种开关类型下都判为无效。开关类型由参数 `gear_sw_type`（0.1.3 起，默认 0）选择，两种的断线失效方式相反——这是选型的唯一判据，不是手感问题：

- `gear_sw_type = 0`（默认，**保持型**四位置开关）：按电平解码。恰好一路为低即该档；四路全高解不出档位，**落 P** 并计 `gear_none_cnt`；多路同时为低**落 P** 并计 `gear_ambig_cnt`。线路断开 ⇒ 落 P，故障看得见，方向安全。
- `gear_sw_type = 1`（**自复位 / 点动**挡杆）：按边沿闩锁，点到哪一档就保持哪一档。杆弹回后四路全高是常态——不换挡、不计故障（`gear_none_cnt` 不再增长）；多路同时为低**不换挡**，仍计 `gear_ambig_cnt`。⚠ 线路断开不产生任何边沿 ⇒ **档位停在原地**（例如停在 D），没有任何故障指示。自复位开关物理上不携带「我还在这一档」这个信息，这个代价消不掉，选它就是接受这一点。

解码结果稳定 `gear_debounce`（默认 50 ms）后生效，上电后 `gear_boot_ignore`（默认 500 ms）内不解码。`GEAR?` 分别报告四路原始电平和解码出的档位，用于接线检查。

4.3 刹车意图合并

物理刹车开关、遥控急停通道、档位在 P、整车使能断开，任一路请求刹车即刹车。各路之间为「或」关系，不设优先级。

> 注意：本板在总线上请求刹车时，驱动器侧的控制台无法解除刹车（驱动器对刹车请求同样取「或」）。检验刹车执行器极性前，须先将本板从 CAN 总线上断开。

4.4 未接入与已断开的输入源

固件区分「从未接入」和「接入后断开」两种状态。从未接入的输入源不参与仲裁；接入后断开的输入源视为失效，按其安全值处理。

- 遥控从未接入：不产生任何主张，车辆不会因未接遥控而保持刹车；
- 遥控曾在线、现已断开：摇杆按中位处理，REMOTE / COLLECT 下强制刹车。

4.5 指令源丢失：档位回 P，两块板一起撤销使能（0.1.4 起）

曾在线的指令源断开后，本板除清除其主张外，还进入锁存状态：

```
档位 → P（空挡）        0x210 的 allow_drive → 0        0x230 的使能位 → 0
```

链路恢复后，车辆不会自动继续行驶，须由驾驶员重新拨一次档解除锁存。锁存生效期间，`tlm veh` 的阻止位图中有对应位指示。

该行为不可通过参数调整。

判定为断开的条件：遥控 / 采集模式，接收机曾有信号后 200 ms 无帧；自动模式，上位机指令帧超过 300 ms；跟随模式，跟随帧超过 200 ms。手动 / 训练模式无远端指令源，不适用。遥控急停通道单独处理。

4.6 驾驶员清除故障的三种操作（0.1.7 起）

驱动板或转向执行器锁存故障后，驾驶员无需连接串口，执行以下任一操作即可清除故障：

操作	判据
连拨两次 P	1.5 秒内两次回到 P
长按 N	在 N 挡保持 2 秒（每次保持只计一次）
屏幕按钮	屏幕向地址 `0x0620` 写 1

一次操作清除整车故障：本板的急停锁存、转向执行器（`0x230` 的清故障位置位一帧）、驱动板（`0x210` 挡位字节 bit7 的上升沿）。`tlm veh` 末尾的 `clrs=` 为累计清除次数。

> 注：出厂触摸屏界面尚未提供清除故障按钮。固件已支持向地址 `0x0620` 写 1 的清除操作。

---

5. CAN 总线接口（整车侧）

本章内容完整独立：按本章接线和解析即可与本板通信，无需参阅其他章节。

> 字节布局以随源码发布的 `App/Inc/foc_can.h`（转向两帧为 `App/Inc/steer_can.h`）为准。本章与其不一致时，以头文件为准。

5.1 总线层

项目	规格
协议	**经典 CAN（非 CAN FD）**
波特率	**1 Mbps**
帧格式	**标准帧（11 位 ID）**，全部 **DLC = 8**
字节序	**小端**（多字节字段低字节在前）
物理层	ISO 11898-2，3.3 V 收发器
终端电阻	板载 120 Ω，经拨动开关投切。总线仅在两端各投入一个，中间节点不投入
接收滤波	本板 mask = 0，全部接收，再按 ID 分发
DLC 检查	本板接收的每一帧均要求 DLC = 8，短帧计数后丢弃（`ST?` 的 `shortrx`）

5.2 ID 分配

ID	方向	周期	内容	本章
`0x210`	控制板 → 广播	**20 ms**	行驶请求。所有驱动板均接收	§5.4
`0x211`	驱动板 → 控制板	20 ms	状态（双电机板；或拓扑 B 的左轮板）	§5.5
`0x212`	驱动板 → 控制板	100 ms	母线遥测 + 车速	§5.6
`0x213`	右轮板 → 控制板	20 ms	状态（仅拓扑 B）	§5.7
`0x214`	右轮板 → 控制板	100 ms	遥测（仅拓扑 B）	§5.7
`0x230`	控制板 → 转向执行器	**20 ms**	转向指令（0.1.4 起）	§5.10
`0x231`	转向执行器 → 控制板	20 ms	转向状态（0.1.4 起）	§5.10

> 注意：总线上新增节点时，不得占用 `0x213` / `0x214`。这两个 ID 保留给拓扑 B 的右轮板，当前未接入不代表今后不接入。占用后接入拓扑 B 时，将出现两个节点发送同一 ID、内容不同的冲突。

两种驱动拓扑，帧格式完全一致：

拓扑	组成	发送的帧
A	一块双电机驱动板	`0x211`（同时填写左右两个轮速槽）+ `0x212`
B	两块单电机驱动板	左板 `0x211`（只填写左轮槽）+ `0x212`；右板 `0x213` + `0x214`

本板将拓扑 B 两块板的上行数据合并为与双电机板等价的视图（轮速分槽、健康位各管半边、母线电流求和）。拓扑按收到的帧 ID 判定，不按某个字段值是否为 0 判定，注意事项见 §5.7。

5.3 速查

```
0x210  控制板发  20 ms
  [0:1] u16 油门 ‰ 0..1000        [2:3] i16 转向 ‰ -1000..1000（本板恒发 0）
  [4]   标志：b0 允许驱动 b1 刹车 b2-7 速度档（2% 一档，0=不声明）
  [5]   挡位 0=N 1=前进 2=后退    [6] 序列号（每帧自增）
  [7]   超时 x10 ms（本板发 30 = 300 ms）

0x211  驱动板发  20 ms
  [0]   健康位（全部 1 = 好）      [1] 驱动板序列号
  [2:3] i16 左轮车架转速 rpm       [4:5] i16 右轮车架转速 rpm
  [6]   回显它解出的整个标志字节   [7] b7 命令有效 / b6 本地锁 / b0-3 采信的挡位

0x212  驱动板发  100 ms
  [0:1] u16 母线电压 mV            [2:3] i16 母线电流 10 mA/LSB
  [4]   序列号                     [5:6] u16 车速 0.1 km/h（绝对值）
  [7]   保留 0

0x230  控制板发  20 ms      转向指令（0.1.4 起）
  [0:1] i16 目标：‰ -1000..1000，或 mm x10（当 [2] b1 置位）
  [2]   标志：b0 使能 b1 单位=mm b2 回中 b3 清故障（上升沿触发）
  [3]   速度限制 0..255 = 对端配置上限的 0..100%（0 = 用对端默认）
  [4]   序列号（每帧自增，状态帧里回显）
  [5]   指令超时 x10 ms（0 = 用对端默认 300 ms）   [6][7] 保留 0

0x231  转向执行器发  20 ms
  [0:1] i16 当前位置 ‰             [2:3] i16 当前位置 mm x10
  [4]   状态位 b0 已标定 b1 已使能 b2 到位 b3 在负限位 b4 在正限位
        b5 转向故障（码在 [5]） b6 助力 b7 驱动侧故障锁存（桥关了）
  [5]   转向故障码                 [6] i8 电机电流 Iq A x10
  [7]   序列号回显
```

5.4 `0x210` 行驶请求（控制板 → 驱动板）

周期 20 ms，持续发送。 本帧为周期帧，安全状态由帧内容表达，不以停发表达。

字节	类型	字段	取值
`[0:1]`	u16 LE	油门	0..1000（‰）
`[2:3]`	i16 LE	转向	−1000..+1000（‰）。本板恒发 0，见 §5.10
`[4]`	位域	标志	见下表
`[5]`	u8	挡位	0 = N / 1 = 前进 / 2 = 后退（>2 一律按 0 处理）
`[6]`	u8	序列号	每帧自增，用于表明发送方在线
`[7]`	u8	超时	×10 ms。0 = 使用驱动器默认值；本板发送 **30**（300 ms）

`[4]` 标志字节：

位	含义
bit0	`allow_drive`：允许驱动

位	含义
bit1	`brake`：请求刹车
bit2 – bit7	速度档，u6，**2% 一档**，0..50 → 0..100%

速度档规则：

① `0` 表示不声明，不表示档位为零。 驱动器收到 0 时不更新该字段，沿用上一次收到的档位。发送方不得将 0 作为有效值发送。

计算方式：`档位步数 = ceil(百分比 / 2)`，写入 `[4] >> 2`。反向解码：`百分比 = 步数 × 2`。

② 字段值为百分比而非档位序号，接收端不得再次缩放。 速度档限制车辆的最高车速，不限制出力。驱动器只将其用于转速上限，不缩放扭矩：最低档与最高档扭矩相同，仅最高车速不同。接收端如将其再乘入油门，将造成重复缩放，表现为低速档带不动负载。

③ 量化误差。 四个常用档 35 / 60 / 80 / 100 中只有 35% 无法精确表示，驱动器收到的是 36%（在 250 rpm 上限下相差 2.5 rpm，约 0.12 km/h）。取整方向为向上。

完整示例（油门 350‰、不转向、允许驱动、不刹车、速度档 60%、前进、序列号 0x2A、超时 300 ms）：

```
ID 0x210  DLC 8
5E 01 00 00 79 01 2A 1E
|__|__|  |__|__|  |  |  |
 350‰    转向 0    |  |  +-- 30 x 10 ms = 300 ms
                  |  |  +----- 序列号 0x2A
                  |  +-------- 挡位 1 = 前进
                  +----------- 标志 0x79 = 0b01111001
                               b0=1 允许驱动, b1=0 不刹车,
                               b2-7 = 0b011110 = 30 步 x 2% = 60%
```

5.5 `0x211` 驱动板状态（驱动板 → 控制板）

字节	内容
`[0]`	健康位，见下表
`[1]`	驱动板自身的序列号，用于判断驱动板是否在线
`[2:3]`	i16 LE 左轮车架机械转速 rpm（饱和处理）
`[4:5]`	i16 LE 右轮，同上
`[6]`	回显驱动板最后一次解出的整个标志字节（含速度档六位）
`[7]`	b7 = 驱动器判定控制板链路有效；b6 = 本地锁；b0 – b3 = 驱动器最后采信的挡位

`[0]` 健康字节（各位均为「1 = 好」）：

位	含义
bit0	电机 1 运行中
bit1	电机 2 运行中
bit2	电机 1 霍尔新鲜
bit3	电机 2 霍尔新鲜
bit4	电机 1 栅驱无故障

位	含义
bit5	电机 2 栅驱无故障
bit6	最近收到过命令
bit7	**刹车驱动已使能**

> 注：「1 = 好」为统一约定，与引脚电平无关。驱动器上低有效的故障信号在发送前已转换为 `_ok` 形式，解析该字节时不得再按引脚极性取反。

`[7]` 字节（不适用「1 = 好」约定）：

位	含义
bit7	命令有效：驱动器判定控制板链路有效
bit6	本地锁：1 = 驱动器处于本地模式，照常接收和回复，但不执行总线指令
bit3 – bit0	驱动器最后采信的挡位

`[6]` / `[7]` 为回环字段。 控制板据此区分三种外部表现相近的失效：

现象	含义	排查方向
帧不再到达	对端未收到本板的帧	线缆、接头、对端是否上电
帧在到达，`[1]` 序列号不变	对端卡死（应用挂死，CAN 邮箱仍在重发）	对端固件
帧在到达，`[7]` bit7 = 0	对端收到了帧，但判定本板链路无效	本板发送侧，序列号是否在变化

本板对三种情况均进行判定（`ST?` 的 `alive` / `frozen` / `believes`）。

> 注：回显 `[6]` 只比对 `allow_drive` 和 `brake` 两位。速度档六位在改档帧与其回显之间必然相差一个周期，不参与比对；这两位承载安全语义。

5.6 `0x212` 母线遥测（驱动板 → 控制板）

字节	内容
`[0:1]`	u16 LE 母线电压，**mV**
`[2:3]`	i16 LE 母线电流，**10 mA/LSB**（拓扑 B 下为两块板之和）
`[4]`	序列号
`[5:6]`	u16 LE 车速，**0.1 km/h**，绝对值，上限 9999
`[7]`	保留，0

车速由驱动板按其自身存储的轮周长换算后发送。更换车轮时，在驱动板上修改轮周长参数即可，无需重新编译，消费端也无需自行换算。

> 注意：`[5:6]` 读数为 0 时有两种可能：车辆静止，或线上没有车速数据（旧版驱动器固件不填写该字段）。本板以三项条件判定车速是否有效：遥测帧是否新鲜；车速报 0 时轮子是否在转；读数量级是否合理（超过 60 km/h 的读数判为坏帧）。任一项不满足，车速反馈判为无效，自动驾驶不使能。

行驶方向不在本帧中。 `0x212` 的车速为绝对值，行驶方向由 `0x211` 中带符号的轮速确定。

5.6b 上行两帧的解码示例

同一时刻收到的两帧，逐字节解析如下：

```
ID 0x211  DLC 8
70 3C 78 00 88 FF 79 81

[0] = 0x70 = 0b01110000
        bit4 电机1 栅驱正常 = 1     bit5 电机2 栅驱正常 = 1
        bit6 最近收到过命令 = 1     bit7 刹车驱动已使能 = 0  <- 见 §5.9 ①
        bit0/1 两路都没在跑，bit2/3 霍尔不新鲜（车停着，正常）
[1] = 0x3C  驱动板序列号 60。与上一帧比对：变了 = 它没卡住
[2:3] = 78 00 (小端) -> 0x0078 = +120 rpm   左轮（车架坐标系，正 = 前进）
[4:5] = 88 FF (小端) -> 0xFF88 = -120 rpm   右轮
        注：一正一负为正常，双轮车必有一侧反装。两侧同号时须检查
[6] = 0x79  回显本板上一帧的标志字节：与本板发出的 0x79 一致
        注：只比对 bit0/bit1：速度档六位相差一个周期为正常
[7] = 0x81 = 0b10000001
        bit7 命令有效 = 1  驱动器判定本板链路有效
        bit6 本地锁  = 0  注：0 不等于「未锁定」，见 §5.9 ②
        bit0-3 采信的挡位 = 1（前进）

ID 0x212  DLC 8
7A 94 FA 00 11 2D 00 00

[0:1] = 7A 94 (小端) -> 0x947A = 38010 mV = 38.0 V   母线电压
[2:3] = FA 00 (小端) -> 0x00FA = 250 x 10 mA = 2.50 A 母线电流
[4]   = 0x11  遥测序列号
[5:6] = 2D 00 (小端) -> 0x002D = 45 x 0.1 = 4.5 km/h  车速（绝对值）
[7]   = 0x00  保留
```

解析结论： 链路正常（序列号在变化，驱动器判定本板链路有效）；车辆以 4.5 km/h 前进；母线 38.0 V / 2.5 A；刹车驱动未使能（`0x211[0]` bit7 = 0）。该位为 0 时含义不唯一，不得直接报告为刹车故障。

> 注意：两个协议中的挡位编码不同，不得相互套用：
>
> | 位置 | 编码 |
> |---|---|
> | CAN `0x210[5]` 与 `0x211[7]` 低四位 | **0 = N，1 = 前进，2 = 后退** |
> | 车载计算机串口帧（§6） | **0 = P，1 = R，2 = N，3 = D** |
>
> 两种编码分别由各自协议规定，均不可更改。网关实现须显式转换，不得直接透传该字节。

5.7 拓扑 B：`0x213` / `0x214`

格式与 `0x211` / `0x212` 完全一致，发送方为右轮板，只填写右轮一侧的数据。

本板的合并规则：

量	合并方式
轮速	各填各的槽
健康位 bit0 – bit5	每侧管自己的三位
健康位 bit6 / bit7	两块板均写入（属于整车级判断），两帧的原始字节分别保存，按固件规则合并；刹车驱动位取「或」
母线电流	求和

> 注意：本板连续收到 3 帧 `0x213` 后才锁定为拓扑 B（锁定后仅断电可复位）。因此接入两块单电机板时，前 60 ms 内 `0x211` 仍按拓扑 A 解释，而拓扑 B 的左板不填写右轮槽。首次搭建拓扑 B 时，须确认这 60 ms 内是否产生错误读数。

5.8 时序、超时与失效行为

链路	判据	超时	到期后
`0x210` 发送	—	20 ms 周期	持续发送；安全状态由帧内容表达
`0x211` 到达	最后一帧的时刻	**500 ms**	判为不新鲜，阻止驱动
`0x211` 序列号	`[1]` 是否变化	**300 ms**（= 15 帧）	判为对端卡死，阻止驱动
`0x212` 到达	最后一帧的时刻	**500 ms**	车速反馈无效，自动驾驶撤销使能
车速合理性	`[5:6]` ≤ 600（60 km/h）	每帧	超出即判为坏帧
车载计算机 AUTO `0x11`	帧新鲜度	**300 ms**	刹车 + 速度 0 + 转向回中
车载计算机 FOLLOW `0x10`	帧新鲜度	**200 ms**	刹车 + 油门 0
遥控	帧新鲜度	**200 ms**	摇杆按中位处理；REMOTE / COLLECT 下强制刹车
遥控急停通道	帧新鲜度	200 ms	曾在线后断开：刹车；从未接入：不主张
触摸屏	心跳	8 s	仅影响指示灯，不影响控制

本板故障时： `0x210` 停发，驱动器 300 ms 后超时并停止输出力矩。由于刹车意图位于该帧的一个位中，此时驱动器无法获知是否需要刹车。本板配有 2 秒独立看门狗，程序挂死时自动复位；发生 HardFault 时立即锁定急停并复位。

5.9 状态位的解读限制

① `0x211[0]` bit7 = 刹车驱动已使能。

该位表示驱动器中保存的一项使能设置，不是测量值，不能证明刹车执行器已实际连接。

表述	可推出的结论	是否已验证
刹车驱动使能已开启	本板发出的刹车位会传到 H 桥	是（即该标志本身）
BRK 口上连接并驱动着执行器	车辆会实际停止	否

执行器脱落、线路断开或卡死时，该位仍为 1。该位不能作为车辆可以停止的依据。

该位为 0 时有两种含义（车辆未装刹车执行器 / 已装但未使能），总线上无法区分，不得直接显示为「刹车故障」。早期驱动器固件中该位恒为 0，与「未使能」无法区分。

② `0x211[7]` bit6 = 本地锁，值为 0 时含义不唯一。

驱动器自 0.4.8 起支持本地锁，但 0.4.8 / 0.4.9 / 0.4.10 无论是否锁定均发送 0，自 0.4.11 起如实上报。总线上没有任何字段携带对端固件版本。

因此，0 表示未锁定，或对端固件早于 0.4.11。总线上存在早于 0.4.11 的驱动器时，不得将 0 判定为未锁定。本板只在读到 1 时动作：不支持本地锁的固件只发送 0，读到 1 时结论可靠；读到 0 时不作判断。

③ 本地锁只能由 bit6 判定。

处于本地模式的驱动板仍照常发送 `0x211`、报告命令有效、回显标志字节并置位「最近收到过命令」，这些位与正常执行总线指令的驱动板完全相同。线路断开的驱动板会停发 `0x211`，可以从帧是否到达判断；驱动板是否执行总线指令，只能依据 bit6 判断。

5.10 转向：`0x230` / `0x231`（0.1.4 起接通）

`0x210[2:3]` 转向字段恒发 0。转向使用独立的 `0x230` 指令帧和 `0x231` 状态帧，转向执行器不接收行驶请求帧，行驶请求中的任何位都不会使转向执行器使能。

本板发送 `0x230` 的规则：

- 每 20 ms 无条件发送一帧。链路是否在线由对端根据帧是否到达判断。
- 使能位由发送函数内部生成，不由调用方传入：整车使能（PD10）为真、已设定过目标、且本帧不在清故障，三个条件同时满足时置位。
- 未设定过目标之前，使能位恒为 0：帧照常发送，执行器不上电。
- 手动 / 训练两个模式下，本板显式释放转向（`SteerCan_Release`），由驾驶员手动转向。转向使能不会因一段时间未设定目标而自动撤销。
- 五个使用转向的模式统一使用 ‰ 目标（±1000 对应对端的软限位），不发送 mm。
- `[3]` 速度限制、`[5]` 超时的覆盖值只作用于本帧，不写入对端的持久化配置。

警告：`[2]` b3 清故障按该位自身的上升沿触发，不按电平触发。`[4]` 序列号每帧自增，以序列号变化作为边沿依据的检测方式等同于电平检测：持续置位 b3 时每 20 ms 清除一次故障，而越程故障在齿条静止后不会重新锁存，翻圈保护因此失效，且无任何提示。同一帧中同时置位使能和清故障不会使执行器上电；须先清除 b3，再在下一帧发送使能。

接收 `0x231` 后的处理： 位置写入 `tlm` 的 `steer` 行。`[4]` b7（驱动侧故障锁存）和 b5（转向故障）含义不同：b7 表示 H 桥已关断，齿条不会动作；b5 表示执行器无法确认自身位置。

警告：不得将牵引驱动板固件烧入转向执行器，反之亦然。两者源自同一套代码，烧错后执行器可正常启动、正常响应串口并通过全部校验，但齿条没有任何限位，且会无条件发送 `0x211` / `0x212`，与驱动板发生 ID 冲突。选择固件时以发布包 `MANIFEST.txt` 中的 `name:` 为准，不按目录名或版本新旧选择。

执行器反装（0.1.8 起）： 参数 `steer_invert`，默认 0。本转向系统的安装方式决定了绝大多数情况下方向固定（执行器正方向 = 车辆右转），装车流程中无需设置左右。仅当执行器反向安装时将其置 1：置 1 后 `0x230` 的目标在总线上取反，`0x231` 返回的位置和左右限位作镜像处理，本板内部仍一律按「正 = 右转」处理。执行器侧无需任何修改。

5.11 第三方接入清单

实现驱动器（接收本板的行驶请求）：

1. 接收 `0x210`，DLC = 8，按 §5.4 解析；
2. 用 `[6]` 序列号判断发送方是否在线，不得只判断帧是否到达；
3. 用 `[7]` 的超时值作为自身看门狗（本板发送 300 ms）；
4. `[4]` bit2-7 为 0 时沿用上一次的速度档，不得按 0% 处理；
5. 速度档只作用于转速上限，不得乘入扭矩；
6. 周期发送 `0x211`（20 ms）与 `0x212`（100 ms），并如实填写 `[6]` / `[7]` 回环字段，控制板依据这两个字段区分三种失效；
7. 健康字节按「1 = 好」统一，未安装的部件如实报 0。

监听总线（仪表、记录仪或诊断工具）：

1. 只接收不发送，不投入终端电阻（位于总线物理末端时除外）；
2. 车速取 `0x212[5:6]`，不得用 `0x211` 的转速自行乘以轮周长换算；
3. 方向取 `0x211` 轮速的符号；
4. 用序列号判断链路状态，不用帧计数；
5. `0x211[0]` bit7 与 `0x211[7]` bit6 的值为 0 时均不作为结论（§5.9）。

禁止事项：

- 不得占用 `0x213` / `0x214`；

- 不得单方面在 `0x210` 中增加或修改位定义。CAN 字节位置由控制板与驱动板共同约定，任何增加或修改须两侧同步进行；
- 总线上不得存在第二个 `0x210` 发送方。

5.12 无整车测试

台架上无驱动器时，可用以下三条命令测试 CAN 侧：

```
FOC RAW <油门‰> <转向> <标志> <挡位> [超时x10ms]      让控制台接管 0x210 发一帧指定内容
FOC RX211 <b0> <b1> ... <b7>                          往本板注入一帧 0x211（十六进制）
ST?                                                    看解析结果、回环三态与全部计数器
```

`FOC RX211` 用于验证本板的解析功能。该命令只验证解析侧；驱动器是否实际发送相应位，须将两块板接在同一总线上验证。

`ST?` 中的 `tx210` 行输出本板最后一帧实际发出的内容。总线上无驱动器时，可据此确认急停对发出帧的覆盖作用。

---

6. 与车载计算机的串口协议

物理口	**USB2**（UART5），115200 8N1
帧	**22 字节定长**，小端
帧头	`AA 55`，版本 `01`
校验	**CRC16-MODBUS**，覆盖偏移 2 – 19，位于帧尾

类型	方向	用途
`0x10`	计算机 → 板	跟随模式指令
`0x11`	计算机 → 板	**自动驾驶指令**
`0x02`	板 → 计算机	跟随就绪 / 状态
`0x03`	板 → 计算机	**自动驾驶就绪 / 状态**，约 10 Hz
`0x01` / `0x04` / `0x05` / `0x06` / `0x07`	板 → 计算机	控制、心跳、手动、遥控、采集上报

`0x11` 自动驾驶指令的载荷：

字段	类型	说明
目标转角	i16，0.1°	左负右正
目标车速	i16，mm/s	+ 前进 / − 后退 / 0 停
刹车	u8	1 = 主动刹车，优先于目标车速
挡位	u8	0=P 1=R 2=N 3=D，回显 / 校验用
标志	u8	bit0 = 请求使能

> 注：未使能时，计算机也须持续发送 `0x11` 心跳（速度 0、刹车 1 等安全值，并置位请求使能位），板子结合自身安全条件决定是否使能。300 ms 内未收到有效帧时，板子刹车、速度归零并清除积分。

`0x03` 状态帧的载荷：

字段	类型	说明
实测车速	i16，mm/s	+ 前进 / − 后退
实测转向	i16，0.1°	
状态位	u8	bit0 已使能 / bit1 驱动器状态新鲜 / bit2 整车使能 / bit3 急停或接管 / bit4 处于自动模式
挡位	u8	
阻止位图	u16	§7 的 `drive_block` 位图，0 = 无任何阻止条件
固件版本	u8 + u8	高 4 位主版本 / 低 4 位次版本；再一字节修订号

> 注：阻止位图与版本字节占用原先固定为 0 的四个保留字节，未改变其他字段的含义。将其视为保留字节的消费端可继续忽略，不受影响。
> 注意：这四个字节的内容尚未在整车上核对（见 §13），接入时须先核对一帧实际数据。

使能条件（须全部满足）： 屏上选中自动模式 + 驱动器状态新鲜 + 整车使能闭合 + 档位 D + `0x11` 心跳在 300 ms 内 + 计算机请求使能 + 无急停 / 无接管。任一条件不满足即不使能，并强制刹车。

---

7. 驱动阻止原因

整车不行驶的原因可能位于使能开关、档位、刹车、遥控、CAN 对端等任一环节，外部现象相同。`WHY?` 将各项原因分开报告。

```
WHY?          位图 + 短标识符，例如：block=0x0003 ven,gear
fault         当前生效的条目，带一句「这是什么」和一句「下一步查哪儿」
fault codes   完整的十四条表
```

位	标识符	含义	性质
bit0	`ven`	整车使能开关未闭合	活值
bit1	`gear`	档位不在 D / R（自动模式下要求 D）	活值
bit2	`brake_sw`	刹车踏板开关按下	活值
bit3	`rc_brake`	遥控急停通道在刹车位	活值
bit4	`rc_lost`	遥控链路曾在线、现已断开（REMOTE / COLLECT 下）	活值
bit5	`foc_stale`	驱动器状态帧不新鲜（> 500 ms）	活值
bit6	`speed_invalid`	车速反馈无效（无数据 / 字段为空 / 量级不合理）	活值
bit7	`host_stale`	车载计算机指令帧不新鲜	活值
bit8	`host_disable`	车载计算机未请求使能	活值
bit9	`cli_test`	台架命令正在接管行驶请求	活值
bit10	`foc_frozen`	帧在到达但序列号不变，驱动器应用卡死	活值
bit11	`foc_disbelief`	驱动器回报已判定本板链路无效	活值
bit12	`foc_locked`	驱动器报告处于本地模式（只对 1 响应，见 §5.9）	活值
bit13	`estop`	**急停锁存**	**锁存**，须执行 `clear`

> 注：位号只增加、不复用。上位机按位号显示说明文字和警告。

> 注意：bit12（驱动器本地锁）在任何模式下均生效。驱动器处于本地锁时，驾驶员踩踏板也不会使其执行。

---

8. 参数

```
param list              整张表：名字、单位、范围、默认值、当前值、可读可写、要不要存
param get <名>
param set <名> <值>
param save              写进板载 EEPROM，并立刻按下次开机的方式回读自验
param load              读回
```

参数表分两类：

类	举例	可写
设置（id 1 起）	自动驾驶限速与增益、各类超时、踏板端点与死区、遥控阈值、CAN 周期与新鲜度门限、档位去抖	可写、可保存
活值（id 60 起）	模式、档位、整车使能、刹车意图、踏板‰、速度档、阻止位图、车速、母线电压电流、链路三态、各类计数器、复位原因	只读

使用规则：

一、行数以帧中的 `count=` 为准。 客户端不得写死行数，参数表可能增加行。

二、`param set` 回显成功不代表设置已生效，每次修改后须读回确认。 标记为「可写」的参数，其执行代码均读取运行时值。

三、参数存储在 MCU 外部的 EEPROM 中，重新烧录固件不会擦除参数。 存储格式为双区轮换 + CRC32：两个区交替写入，读取时两区均读，取校验通过且序号较大的记录。任何时刻至少保留一条完整的旧记录，写入过程中掉电可以恢复。记录只允许追加字段，旧记录按自身长度校验，缺少的字段取默认值。

四、每条记录标明所属的参数表（0.1.11 起）。 由于第三条，参数记录在烧录其他固件后仍保留，即使新固件的参数表完全不同。参数编号由各参数表独立定义：出厂固件的 1 号参数为自动模式限速（0 – 1500 mm/s），开发模板（§3.2）的 1 号参数为示例参数（−1000 – 1000）。

记录头中包含所属参数表的标识。标识与当前固件不符时，整条记录不予装载，参数全部保持默认值，`param load` 输出相应提示。

> 注意：参数表标识可防止误读，不能防止覆盖。两个备份区轮换写入，在另一份固件上执行两次保存后，两个区均被覆盖，原有参数丢失。在两份固件之间切换烧录前，须先用 `param list` 记录重要参数值（特别是已标定的踏板端点）。

8.1 参数装载失败的提示（0.1.12 起）

`param load` 报告装载 0 条时，原因有四种，排查方法各不相同。自 0.1.12 起，每种原因输出不同的提示：

板子输出	含义	处理
`nothing has ever been saved on this board`	从未保存过参数	出厂状态，非故障
`a record is there but does not check out`	有记录，但校验不通过	重新执行 `param set` + `param save`
`the stored record belongs to a DIFFERENT parameter table`	本板曾烧录其他固件，存储的参数不属于当前参数表	见上文第四条
`no storage chip answered on I2C1`	存储芯片无应答	先断电再重新上电，见下

> 注意：板子之前工作正常、之后报告无存储芯片应答时，须断电后重新上电。处理器可能在任意时刻复位（烧录、看门狗、复位按键），包括正在从存储芯片读取数据的过程中。此时芯片保持数据线为低电平，复位后处理器不会释放该状态，I²C 总线无法通信，现象与未焊接存储芯片相同。自 0.1.12 起，固件上电时先释放总线，该情况基本不再出现；如仍出现，断电重新上电即可恢复。

---

9. 踏板标定

```
calib list          这块板有哪些标定项
calib start pedal   开始踏板端点标定，板子一步步给提示
calib confirm       确认当前这一步（松开 / 踩到底）
calib status        现在走到第几步
calib abort         放弃
```

标定对象为踏板的两个端点（松开位与踩到底位），结果写入参数表并通过 `param save` 保存。上位机按 `calib status` 逐步引导用户操作，每一步均有机器可读的回执。

---

10. 自检

```
selftest run        八项，逐条给结论
```

输出为机器可读格式：`=st 1 result item=<项> verdict=<pass|fail|info|skip> ...`，最后一行为 `=st 1 summary total= pass= fail=`。

项	检查内容	说明
`image`	镜像自述头与校验	
`eeprom_present`	I²C 上是否有 EEPROM 应答	器件是否存在
`eeprom_rw`	对隔离地址的读写	不涉及参数存储的两个区
`param_record`	存储中是否有一条有效记录	`info`：出厂板卡本身无记录，非故障
`imu`	惯性传感器是否存在	
`pedal_adc`	踏板 ADC 当前读数	`info`：此项无标准值
`watchdog_bit`	看门狗是否曾触发复位	`info`：表示过去曾发生复位，不表示板子当前异常
`can_loopback`	—	`skip`，并输出原因：环回测试会使本板脱离总线并停发行驶请求

> 注：自检只读取已有状态，以及对隔离资源进行读写，不重新配置正在工作的外设，不会中断本板正在进行的通信。

---

11. 急停与安全申报

11.1 急停

```
estop       立即闩死：行驶请求被压成「停 + 刹车」。任何状态下都可调
clear       显式解除（边沿触发，一条命令 = 一次事件）
```

0.1.16 起，控制台 `clear` 清除整车：本板急停、转向执行器故障（`0x230` 清故障位）和两块驱动板的锁存故障（`0x210` 字节 5 bit7，单路驱动板 0.6.3 起识别），与驾驶员清除故障的操作（§4.6）做同一件事。各板只在故障原因已消失时清除，清除后以 `tlm` / `ST?` 中各板的健康位为准。

急停在行驶请求发送函数内部实现，对所有行驶请求出口（台架命令与正常仲裁）均生效。

11.2 安全申报（必读）

问题	本板的答案
是否有不经过 CPU 的关断通路?	**无。** 本板无功率级、无栅极驱动，急停由软件实现：锁定一个标志，使每一帧行驶请求变为「停 + 刹车」。
是否有温度保护?	**无。** 本板不测量温度。母线电压和电流来自驱动器的遥测帧，由驱动器测量，本板仅转发。
载人 / 载重场景是否需要外部物理断路器?	**需要，且为强制要求。** 本板的急停依赖 MCU 运行、CAN 通信正常、驱动器响应。任何失控可能造成伤害的场合，须另配独立的物理断路器，直接切断驱动器的动力回路，不经过本板。

警告：本板的急停由软件实现，不具备不经过 CPU 的关断通路。载人、载重或任何失控可能造成伤害的车辆，须另配独立的物理断路器，直接切断驱动器的动力回路。

其他安全相关实现：

- **HardFault**：锁定急停，在不清零的 RAM 中记录故障信息，然后立即复位。故障记录可通过 `tlm` 与 `fault` 查询。注：该记录在复位后保留，断电后丢失。
- **独立看门狗 2 秒**：程序挂死时自动复位。

---

12. 升级与发布包

引导方式	本板无 bootloader，也无 A/B 槽，只有一个镜像，起始地址 `0x08000000`
升级通道	MCU 的 ROM BootLoader（ISP），经 USB1
进入方式	§2.1 的 RTS / DTR 时序
镜像自述头	位于镜像偏移 `0x200`：长度、CRC32、版本、构建标识、是否 Debug 构建

发布包目录包含：`MANIFEST.txt`、`.bin`、`.hex`、`SHA256SUMS`、`COMMANDS.json`。

发布包另附 `source.zip`：完整源码树，出厂固件和开发模板（§3.2）均由其编译，包含主机测试和仿真。

0.1.17 的 `MANIFEST.txt` 摘要（字节数、CRC32 和构建时间以包内实际文件为准）：

```
name: DlaBoard_B
version: 0.1.17
commit: 5eaa56e
built: <构建时间>
config: Release
application: DlaBoard_B.bin <字节数> bytes, crc32 <8 位十六进制>, load 0x08000000
source: source.zip
```

> 注：本板无镜像槽，`MANIFEST.txt` 中的角色字段为 `application:`。
> 警告：升级期间本板停发行驶请求。装车使用时，须确保车辆处于安全状态后再升级。

升级后须完成以下三项检查：

1. `ver`：确认版本号和 `crc ok`；

2. `param list`：确认参数仍在（参数存储在 MCU 外部的 EEPROM 中，烧录不会改动）。如本板曾烧录其他固件，须先阅读 §8 第四条；
3. 逐一检查本板连接的外设：触摸屏（`app` 的 `out=` 应与屏上所选速度档一致）、驱动器（`tlm can` 的 `alive=1`）、存储芯片（`selftest run`）。判定标准为每一路均输出预期值，仅确认板子运行不够。烧录会复位本板，但不会复位与其连接的外设，烧录完成后应立即检查各路连接。

---

13. 验证状态

注意：下表将已验证与未验证的项目逐行分开列出，未验证的项目不构成保证。

已在板上验证：

项	说明
安全项、看门狗、构建标识	含反向验证：人为挂死后由看门狗复位
`WHY?` 阻止位图	报出 `ven,gear`（使能断开 + 档位在 P）
CAN 回环三态（alive / frozen / believes）	
参数表读取、写入、保存与回读自验	
`0x211[0]` bit7 刹车驱动使能	两个方向均由接收端观测：置位读到 1，清除读到 0
健康位合并（拓扑 A）	
串口 DTR / RTS 的复位组合与恢复步骤	§2.1 全部为实测结果
发布包烧录流程	探测 → 全片擦除 → 写入 → 回读校验 → 复位到应用 → `ver` 有回应
转向链路 `0x230` / `0x231`	已在整车上验证（固件 0.1.4 起），见 §5.10

尚未验证（不构成保证）：

项	说明
刹车驱动使能位的冷启动行为	已验证调试器复位后的行为，断电冷启动未验证
健康位合并（拓扑 B）	未验证
`0x211[7]` bit6 本地锁的线上行为	解码已验证（注入 `0→1→0` 三个方向均正确）；驱动器实际发送该位未验证
急停对已发出帧的覆盖作用	使能开关断开时，仲裁已将相同字段置为安全值，覆盖作用无法观测。验证须闭合使能开关（车轮须离地）
`0x03` 状态帧的阻止位图与版本字节	自动模式只能从触摸屏进入，尚未在接有触摸屏的条件下验证
档位接线	接线标准已确定，空载已验证（不接线时落 P，计数器不增加）；2026-09-07 台架接入过一根自复位挡杆、四路均可触发（`gear_sw_type` 由此新增），保持型开关接入与带参数换挡的完整验证未进行
刹车执行器极性	极性接反时，「刹车」命令实际为松开刹车。验证极性前须先将本板从 CAN 总线上断开（见 §4.3）
轮速脉冲输入（CN20 / CN19）	固件对脉冲计数，但计数值无输出途径，也未在车上标定（左右轮与两路输入的对应关系未确定）。当前无功能依赖该输入，车速取自 `0x212`，见 §1.3.3
转向正负号参数 `steer_invert`	参数及主机测试已具备（0.1.8），未在执行器反装的车辆上验证。默认 0，不改变行为，见 §5.10 末

---

14. 版本历史

14.1 固件

版本	日期	内容
0.1.17	2026-11-11	换向时照常发送 R 挡，不再改写为 N + 刹车（撤销 0.1.16 的这一项：刹车位使驱动板短路制动，高速换向时有撞击声）。换向由驱动板完成，单路驱动板须为 0.6.4 或更新版本。
0.1.16	2026-11-11	仅台架使用，未公开发布。控制台 `clear` 清除整车（§11.1）；换向时改发 N + 刹车（0.1.17 已撤销）。
0.1.15	2026-11-11	新增控制台命令 `STEER <位置 mm> <速度>` 和 `STEER OFF`（§3），仅连接 USB1 即可向转向执行器发送位置和速度。
0.1.14	2026-09-16	`build` 记录增加 `hw=` 字段，报告本镜像对应的硬件版本。
0.1.13	2026-09-11	修复 0.1.11 / 0.1.12 中触摸屏始终无法连接、屏上速度档不再下发的问题，装有触摸屏的车辆建议升级。
0.1.12	2026-09-11	修复源码包主机测试失败，`param load` 区分四种装载失败原因（§8.1），引脚核对输出在中文 Windows 上不再乱码。
0.1.11	2026-09-11	新增开发模板编译配置与 `template.yaml`（§3.2），参数记录标明所属参数表（§8 第四条），上电时释放存储芯片总线（§8.1）。
0.1.10	2026-09-10	应用层速度档字段输出到总线（§3.1）。
0.1.9	2026-09-10	出厂固件分为核心层与应用层（`App/apps/`），新增 `app` 命令与主机仿真 `sh tests/sim/run.sh`。
0.1.8	2026-09-10	新增参数 `steer_invert`（0/1，默认 0），用于执行器反装的车辆（§5.10 末）。
0.1.7	2026-09-10	新增驾驶员清除故障的三种操作（连拨两次 P / 长按 N 2 s / 屏幕按钮）及 `tlm veh` 的 `clrs=` 字段（§4.6）。
0.1.6	2026-09-10	发布包附带 `source.zip`，清单头增加 `commit:` / `built:`。
0.1.5	2026-09-10	四个处理器故障向量（HardFault / MemManage / BusFault / UsageFault）统一执行锁定急停、记录、复位，`fault` 故障码成帧输出，发布默认 Release 构建。
0.1.4	2026-09-10	转向接入 `0x230` / `0x231`（§5.10），新增指令源丢失锁存（§4.5），遥控改为 iBUS 并将两个限速旋钮分别用于油门上限和转向上限，移除 `servo_can` 模块，`tlm` 增加 `steer` 行，构建改为 CMake + Ninja。
0.1.3	2026-09-07	新增自复位挡杆支持（`gear_sw_type`）。
0.1.2	2026-09-07	挂 D / R 后前两帧油门置零，以满足驱动板挂挡互锁条件。
0.1.1	2026-09-01	新增 `id` 命令，修正发布包 `MANIFEST` 的 CRC32 算法。
0.1.0	2026-08-31	首个 Release 构建，包含镜像自述头与盖章、`SHA256SUMS`、命令表 id 唯一性检查、出厂自检、踏板标定、`estop` / `clear` 与 HardFault 处理、参数持久化（板载 EEPROM，双区轮换 + CRC32 + 回读自验）、`fault` / `fault codes`、`param get` / `set` / `save`、档位电平解码与接线标准、驱动器本地锁解码，以及控制台与车载计算机分用两个 USB 口。

14.2 本书

版本	日期	说明
Rev. 1.10	2026-11-11	对应固件 0.1.17，§11.1 增加控制台 `clear` 清除整车的说明，修订 §12、§14 及页脚。
Rev. 1.9	2026-11-11	对应固件 0.1.14 / 0.1.15，§3 增加 `STEER` 命令，修订 §12、§14 及页脚。
Rev. 1.8	2026-10-09	修订 §1.3.2、§4.2：自复位挡杆支持（参数 `gear_sw_type`，固件 0.1.3 起）此前只记在 §14.1 变更记录，§1.3.2「须使用四位置选择开关」与 §4.2「不按边沿锁存、断线落 P」仍按 0.1.3 之前的实现描述，与固件实现矛盾；本次按当前固件对齐，并分别写明两种开关的断线行为。
Rev. 1.7	2026-09-17	文字修订，内容不变。
Rev. 1.6	2026-09-11	对应固件 0.1.11 – 0.1.13，新增 §3.2、§8.1，修订 §8、§12、§15 及页脚。
Rev. 1.5	2026-09-10	对应固件 0.1.9 / 0.1.10，新增 §3.1，修订 §3 命令表。
Rev. 1.4	2026-09-10	修订 §13 验证状态与 §1.3.1 接口表。
Rev. 1.3	2026-09-10	对应固件 0.1.8，修订 §5.10 `steer_invert` 说明。
Rev. 1.2	2026-09-10	对应固件 0.1.2 – 0.1.7，改写 §5.10，补充 §2.3、§5.2、§5.3，新增 §4.5、§4.6。
Rev. 1.1	2026-09-04	新增 §1.3 接口功能定义，补充 §4.1 输入表的连接器。
Rev. 1.0	2026-09-02	首版，对应固件 0.1.1，含 §5 CAN 协议全文。

> 注：《产品规格书》Rev. 1.1 §17.3 中「参数只读、不可掉电保存」与「出厂为 Debug 构建」两项描述不适用于 0.1.0 及以后的固件（参数可写可存，出厂为 Release 构建并已盖章），以本书为准。

---

15. 已知限制（固件侧）

#	项	说明
1	**急停由软件实现**	本板无不经过 CPU 的关断通路。载人 / 载重场合须另配物理断路器，见 §11.2
2	**转向正负号未在反装车辆上验证**	参数 `steer_invert` 及主机测试已具备（0.1.8 起），未在执行器反装的车辆上验证。默认 0，不改变行为，见 §5.10 末
3	**无温度保护**	本板不测量温度；母线读数由驱动器测量，本板仅转发
4	**命令表的 `needs` 字段均为 `none`**	该字段的取值（桥 / 力矩 / 已标定）针对带功率级的板卡定义，均不适用于本板。本板命令的实际前提为使能开关闭合并挂 D 档，见 §7 阻止位图
5	**台架命令会接管行驶请求**	`FOC D` / `FOC R` / `FOC RAW` 执行期间 `WHY?` 显示 `cli_test`；命令 300 ms 未刷新即失效
6	**拓扑 B 切换有 60 ms 窗口**	连续收到 3 帧 `0x213` 后才锁定，其间按拓扑 A 解释，见 §5.7

#	项	说明
7	**在两份固件之间切换烧录可能丢失参数**	记录带有参数表标识，不会被误读；但备份区轮换写入，无法防止覆盖，在另一份固件上保存两次即覆盖两个区。切换固件前须先用 `param list` 记录参数，见 §8 第四条
8	**开发模板尚未在用户环境中完整验证**	§3.2 的流程已在公开下载的发布包上验证（建立工程 → 编译 → 主机测试 → 烧录 → 认板 → 参数存取经复位保持），尚未在用户环境中从零完整验证。如遇不一致之处，请联系 Daliang Auto

© Daliang Auto · 整车控制板 出厂固件说明书