

MagFass2Car (FASS 2.0) 使用指南

适用对象：第一次接触磁导航 + FASS 2.0 的调试/实施人员

插件：`StandardScene.Magnetic.dll`

车型：`MagFass2Car`

最后更新：2026-07

目录

1. [这套系统能做什么](#1-这套系统能做什么)
2. [5 分钟快速上手](#2-5-分钟快速上手)
3. [整体架构（新手必读）](#3-整体架构新手必读)
4. [场景与进程配置](#4-场景与进程配置)
5. [车辆参数说明](#5-车辆参数说明)
6. [通讯模式：UDP 与 TCP](#6-通讯模式udp-与-tcp)
7. [三种移动方式对比](#7-三种移动方式对比)
8. [地图配置：站点与路径](#8-地图配置站点与路径)
 - [8.1 节点块结构](#81-协议节点块25-字节结构)
 - [8.2 站点 fields 详细说明](#82-站点-fields-详细说明)
 - [8.3 路径 Track fields 详细说明](#83-路径-track-fields-详细说明)
 - [8.4 组包优先级与默认规则](#84-组包优先级与默认规则)
 - [8.5 典型场景配置示例](#85-典型场景配置示例)
1. [LoopMission 环线任务](#9-loopmission-环线任务)
2. [任务状态机与协议下发](#10-任务状态机与协议下发)
3. [状态查看与手动操作](#11-状态查看与手动操作)
4. [常见问题排查](#12-常见问题排查)
5. [参数推荐与检查清单](#13-参数推荐与检查清单)
6. [术语表](#14-术语表)

1. 这套系统能做什么

MagFass2Car 是 StandardScene **磁导航平台插件**里的 FASS 2.0 车型，主要能力：

能力	说明
路径下发	通过 0xB1 下发站点序列（每帧最多 10 站）
动作下发	通过 0xA1 补发站点动作（顶升、充电等）
状态接收	UDP 模式：车体主动上报 100B，调度回 0x10
站点业务	从地图 fields 读取启停、分叉、顶升等，拼进任务
环线配合	与 Core LoopMission 配合，自动跑 goalSite
任务闭环	状态机判断到站 + 动作是否完成，再发下一段

同一插件内还有 **MagCar（FASS 1.0 TCP）**，两者可并存，注意车型和 Mission 不要配混。

2. 5 分钟快速上手

步骤 1：加载插件

确认场景已加载 scene.mag（StandardScene.Magnetic.scene.json），提供：

- 车型：MagCar、MagFass2Car
- 进程：EventCarMission、MagFass2LoopMission

步骤 2：添加一辆车

1. 在场景中添加车型 **MagFass2Car**
2. 填写基本参数（UDP 模式示例）：

参数	示例值	含义
address	192.168.1.100	车体 IP
Port	5000	下发命令的目标端口
ListenPort	20103	调度监听车体上报的端口
VehicleCode	1	车体编号，须与上报帧一致
CommMode	Udp	PCB/UDP 模式（推荐）

步骤 3：确认在线

- 车体开始 UDP 上报后，车辆状态应显示「上线」
- `keepAlive` 日志出现 `keepAlive udp ok`
- 枚举里 `Protocol = FASS2.0/UDP`

步骤 4：手动走一步（验证通讯）

在车辆方法里点 **「查询状态」**，再点 **「启动」**，确认有日志且无报错。

步骤 5：跑一段路径

方式 A — 手动 Go（Web/调度下发）

用系统自带的「去某站」功能，内部会编译脚本，例如：

```
agv.MagFass2Go(1, 2, 0.2);
agv.MagFass2Go(2, 5, 0.2);
```

方式 B — LoopMission 环线

1. 添加 **`LoopMission`** 或 **磁导航FASS2环线**
2. 配置 `tasklist.json`
3. 启动进程
4. 车到站后自动分配 `goalSite` 并导航

3. 整体架构（新手必读）

业务层：LoopMission / 手动 Go / 脚本
（分配 `goalSite` 或编译多行 `MagFass2Go`）



MagFass2Car 执行层

- 脚本拦截（Loop + `goalSite`）→ 一次全程任务
- `Fass2TaskBuilder`：路径 + `fields` → 节点序列
- `Fass2TaskStateMachine`：滑动窗口下发 + 动作闭环



协议层：Fass2Protocol + Fass2UdpHub

- 收 100B 状态 → 回 0x10
- 发 0x01/0xB1/0xA1 到 address:Port

AGV 车体

****关键理解：****

- ****LoopMission (Core) **** 只负责「给谁分配哪个目标站」（写 goalSite 标签）
- ****MagFass2Car**** 负责「怎么走过去」（拼包、下发、等到站）
- ****阶段 4**** 后，Loop 不再逐行执行多段脚本，而是****拦截为一次全程任务****（可关）

4. 场景与进程配置

4.1 插件清单

文件：StandardScene.Magnetic.scene.json

```
{
  "carTypes": [ "MagCar", "MagFass2Car" ],
  "missionTypes": [ "EventCarMission", "MagFass2LoopMission" ]
}
```

4.2 选哪个 Mission?

Mission	来源	适用
LoopMission	Core	通用环线，MagFass2Car 同样可用
磁导航FASS2环线 (MagFass2LoopMission)	Magnetic 插件	逻辑与 LoopMission 相同，名称标识磁导航 FASS2 场景
FASS车辆事件进程 (EventCarMission)	Magnetic 插件	**仅 MagCar (FASS 1.0) ** ，不驱动 MagFass2Car

新手建议：MagFass2Car 环线用 ****`LoopMission`**** 或 ****`磁导航FASS2环线`**** 均可。

4.3 tasklist.json 最小示例

```
{
```

```
"tasks": [
  {
    "id": "loop-1-2",
    "startType": "AutoLoop",
    "kind": "Loop",
    "currentStationId": 1,
    "targetStationId": 2,
    "priority": 1,
    "trafficControl": 2
  },
  {
    "id": "loop-2-1",
    "startType": "AutoLoop",
    "kind": "Loop",
    "currentStationId": 2,
    "targetStationId": 1,
    "priority": 1,
    "trafficControl": 2
  }
]
```

含义：车到站 1 → 自动分配目标站 2；到站 2 → 自动分配目标站 1，形成环线。

5. 车辆参数说明

在场景编辑器中选中 `MagFass2Car`，常用字段如下。

5.1 通讯

字段	默认	说明
<code>CommMode</code>	<code>Udp</code>	<code>Udp</code> =PCB 被动收状态； <code>Tcp</code> =PLC 主动查询
<code>address</code>	—	车体 IP
<code>Port</code>	<code>5000</code>	命令下发端口
<code>ListenPort</code>	<code>20103</code>	调度 UDP 监听端口
<code>VehicleCode</code>	<code>0</code>	必须与车体上报的 <code>Car</code> 字段一致
<code>UdpOfflineTimeoutMs</code>	<code>15000</code>	超过此时间未收到上报视为离线

5.2 任务与状态机

字段	默认	说明
EnableTaskStateMachine	true	开启任务状态机（推荐）
EnableLoopTaskDrive	true	Loop 场景拦截多段脚本，改全程任务
LockCount	4	每次 0xB1 滑动窗口站数（≤10）
TaskResendIntervalMs	500	周期重发 0xB1 间隔
ActionRetryIntervalMs	1000	动作未完成时 0xA1 重试间隔
MoveTimeoutSeconds	120	单次移动超时（秒）
StartBeforeMove	true	移动前发 0x01 启动

5.3 地图节点映射

字段	默认	说明
UseTagValueAsNode	false	true 时用站点 TagValue 作为协议 Node 号

5.4 调试

字段	默认	说明
EnableDetailLog	true	详细日志
LogRawFrame	true	打印收发十六进制

6. 通讯模式：UDP 与 TCP

6.1 UDP/PCB（推荐，现场 FASS 2.0 新车默认）

车体 —UDP→ 调度 ListenPort（100B 状态）
调度 —UDP→ 车体 address:Port（0x10 应答）
调度 —UDP→ 车体 address:Port（0xB1/0xA1/0x01，独立发送）

特点：

- 调度**不主动轮询**，靠车体上报驱动状态机
- 每收到一帧状态，**必须回 0x10 **（已由 Fass2UdpHub 自动处理）
- 下发命令**不等待同步应答**（异步）

6.2 TCP/PLC（调试/兼容）

- 调度主动发 0x00 查询，同步收 100B
- 适合仿真器或无 UDP 上报的环境
- 设置 CommMode = Tcp，配置 ConnectTimeoutMs 等

6.3 端口核对清单

检查项	操作
车体上报发到哪	确认发往调度机 ListenPort
调度命令发到哪	确认 address:Port 与车体监听一致
编号一致	VehicleCode = 上报帧内 Car 字段
防火墙	放行 UDP 双向

7. 三种移动方式对比

方式 1：多段脚本（传统）

****触发****：手动 Go、EnableLoopTaskDrive=false 时的 Loop、任意编译脚本。

****流程****：

```
FindRoute → Compile → 多行脚本：
  agv.MagFass2Go(1,2,speed);
  agv.MagFass2Go(2,3,speed);
  ...
→ SelfEvaluating 逐行执行
→ 每行：锁终点 → 发 0xB1 → 等到站 → 释放起点
```

****特点****：每**条边**一段任务，多次 taskId，与交管「逐段锁点」一致。

方式 2：Loop 全程任务（阶段 4，默认）

****触发****：goalSite 标签 + 编译脚本含 MagFass2Go + EnableLoopTaskDrive=true。

****流程****：

```
LoopMission 写 goalSite
```

- GoSite 仍 Compile 多行脚本 (Core 不变)
- actualSendScript 拦截, 不逐行执行
- ExecuteGoalTaskAsync(当前站 → goalSite)
- 状态机一次规划全程, 滑动窗口发 0xB1

****特点****: 业务上一次任务, 协议上仍遵守每帧 ≤10 站。

方式 3: 单段 MagFass2Go (脚本一行)

****触发****: 脚本里只有一行, 或无 goalSite 的测试。

****流程****: 与方式 1 的单行相同, 状态机只跑 src → dst 一段。

对比表

维度	多段脚本	Loop 全程任务
规划范围	每条边	当前站 → goalSite
taskId	每段一个	全程一个
交通锁	每段锁终点	只锁 goalSite
适用	手动调试、兼容旧逻辑	环线自动跑
开关	EnableLoopTaskDrive=false	EnableLoopTaskDrive=true (默认)

8. 地图配置：站点与路径

任务组包时, 调度读取地图中站点 (Site) 和路径 (Track) 的 fields 键值, 合并写入 FASS 2.0 的 0xB1 节点块 (每站 25 字节)。

****配置位置****: 地图编辑器 → 选中站点或路径 → 属性/fields 面板 → 添加键值对。

****命名约定****:

- 业务动作字段统一以 Fass2_ 开头 (代码常量见 Fass2SiteFields.cs)
- 值为**十进制整数**字符串, 例如 Fass2_Lift = 1
- 布尔字段写 true / false 或 1 / 0

8.1 协议节点块 (25 字节结构)

每个站点在 0xB1 中对应一个节点, 主要字段如下 (与 Fass2NodeMessage 一致):

偏移	字段	类型	来源
0-1	Node	ushort	站点 TagValue 或 site.id
2-3	Distance	ushort	上一站到本站的边长 (mm 量级)
4	StartStop	byte	站点 Fass2_StartStop / 默认规则
5	Direction	byte	站点或边 fields
6	Orientation	byte	边朝向计算或站点覆盖
7	Byroad	byte	分叉选择, 站点或边
8-9	Speed	ushort	边 Speed 换算 (0.1 m/min 单位)
10	Obstacle	byte	站点
11	Audio	byte	站点
12	Light	byte	站点
13	Charge	byte	站点
14	Rest	byte	站点
15	Lift	byte	站点
16	Clamp	byte	站点
17	Tray	byte	站点
18	Roll	byte	站点
19	Shutdown	byte	站点

状态机到站后，会比对车体上报状态帧中**同一节点块**的字段是否达到期望值；未达标则补发 0xA1。

8.2 站点 fields 详细说明

8.2.1 节点与调度类

`TagValue` (int, 可选)

项目	说明
映射协议字段	Node （节点号）
何时配置	磁导航地标号与地图 site.id **不一致**时必须配置
车辆开关	MagFass2Car.UseTagValueAsNode = true
示例	TagValue = 10086
注意	未配置且 UseTagValueAsNode=false 时，直接用 site.id 作为节点号

`goSite` (int, 可选)

项目	说明
映射协议字段	无（不参与 0xB1 组包）
用途	LoopMission 在该站无任务匹配时，读取此字段作为默认下一目标
读取方	Core AbstractLoopMission.LoopStartAction
示例	站 3 配 goSite = 5，车到站 3 且无 tasklist 匹配时，自动写 goalSite=5

`Fass2\_WaitMode` (string, 预留)

项目	说明
映射协议字段	无（当前版本不直接写入节点块）
规划值	None / Button / Plc / Api
当前状态	字段已定义，**等待放行联动逻辑待扩展**；可通过修改 Fass2_StartStop 或外部改 fields 触发重发

8.2.2 `Fass2\_StartStop` — 启停模式 (byte)

映射协议字段：StartStop （节点块第 5 字节）

值	含义 (对齐 FASS 2.0 字典)	典型场景
1	过站 (不停)	路径中间站、通道站
2	普通停车	工位、等待点、装卸点
11	管控启动	管控区入口, 需授权后启动
12	管控停止	管控区停车等待
22	精准停止	装配、对接, 精度要求高

****默认规则**** (未显式配置 `Fass2_StartStop` 时, `Fass2TaskBuilder` 自动决定) :

站点位置	默认 StartStop
中间站	1 (过站)
末站 (goalSite)	2 (停车)
末站 + <code>Fass2_PrecisionStop=true</code>	22 (精准停)

****动作完成判定****: 若配置了非过站值, 状态机要求上报 `Node.StartStop` 与期望值一致, 且车体非运行态 (`State≠1`), 才推进下一段。

****与 `Fass2_PrecisionStop` 关系****:

- 可直接写 `Fass2_StartStop = 22`
- 或末站写 `Fass2_PrecisionStop = true` , 由组包器自动设为 22

8.2.3 `Fass2\_PrecisionStop` — 精准停 (bool)

项目	说明
类型	<code>true / false / 1 / 0</code>
效果	仅对 **路径末站** 生效, 等价于 <code>StartStop = 22</code>
优先级	低于显式 <code>Fass2_StartStop</code> (若同时配置, 以 <code>Fass2_StartStop</code> 为准)
示例	装配工位末站: <code>Fass2_PrecisionStop = true</code>

8.2.4 `Fass2\_Byroad` — 分叉选择 (byte)

映射协议字段: `Byroad`

项目	说明
取值范围	通常 0 – 3，表示磁分叉/岔路选择序号
站点 vs 边	站点显式配置优先；未配置时继承**进入该站的边** Fass2_Byroad 或 magSelect
典型场景	Y 型路口、多磁条岔口
完成判定	若期望值非 0，上报 Node.Byroad 须一致

8.2.5 `Fass2\_Direction` — 车头旋转 (byte)

映射协议字段：Direction

项目	说明
含义	到站后车头旋转指令（具体角度定义以车体协议为准）
站点 vs 边	站点配置覆盖边配置
完成判定	上报 Node.Direction 与期望值一致

8.2.6 `Fass2\_Orientation` — 行驶朝向 (byte)

映射协议字段：Orientation

项目	说明
含义	进出该站时的前进/后退/横移朝向（0–255，对应 0°–360° 量化）
默认	由边几何计算：atan2(Δy , Δx) + Reverse + CarDirectionBias
覆盖	站点或边显式配置 Fass2_Orientation 时直接使用
完成判定	非 0 期望值时须上报一致

8.2.7 `Fass2\_Lift` — 顶升/举升 (byte)

映射协议字段：Lift

值	含义（对齐 backend 字典）
1	上升到位
2	下降到位
3	上升中
4	下降中

项目	说明
典型场景	顶升搬运、装卸高度切换
配置建议	工位停车（StartStop=2）+ Fass2_Lift=1 或 2
完成判定	上报 Node.Lift 等于配置值；未完成自动 0xA1 补发
注意	一般配**到位值**（1/2），不要配中间态（3/4）除非明确需要

8.2.8 `Fass2\_Roll` — 滚筒 (byte)

映射协议字段：Roll

项目	说明
用途	滚筒输送启停/方向
取值	以现场车体协议为准（写入节点块第 19 字节）
完成判定	期望值非 0 时须上报一致

8.2.9 `Fass2\_Charge` — 充电 (byte)

映射协议字段：Charge

值	含义
1	停止充电
2	开始充电

项目	说明
典型场景	充电站、LoopMission 充电触发任务
配置建议	充电工位： Fass2_StartStop=2 + Fass2_Charge=2
离站	下一任务或离站前可配 Fass2_Charge=1

8.2.10 `Fass2\_Rest` — 休眠 (byte)

映射协议字段： Rest

值	含义
1	停止睡眠
2	开始睡眠

项目	说明
典型场景	长时间待机工位、低功耗停车区

8.2.11 `Fass2\_Obstacle` — 避障区 (byte)

映射协议字段： Obstacle

项目	说明
用途	切换车体避障区域/模式
取值	以车体协议定义为准（区域编号）
典型场景	进入狭窄通道、人机混行区

8.2.12 `Fass2\_Audio` / `Fass2\_Light` — 声光 (byte)

字段	协议字段	用途
Fass2_Audio	Audio	提示音、播报
Fass2_Light	Light	警示灯、状态灯

取值以车体协议为准，一般为模式/曲目/灯色编号。

8.2.13 `Fass2\_Clamp` — 夹抱 (byte)

映射协议字段： Clamp

项目	说明
用途	夹紧/松开机构
典型场景	夹抱式 AGV、治具锁紧

8.2.14 `Fass2\_Tray` — 托盘 (byte)

映射协议字段： Tray

项目	说明
用途	托盘旋转/伸缩等机构
典型场景	多工位转台、托盘调角

8.2.15 `Fass2\_Shutdown` — 关机 (byte)

映射协议字段： Shutdown

项目	说明
用途	到站后车体软关机/深度待机
注意	与全局 0x06 关机指令不同；此为**任务节点内**动作
风险	配置后车体可能下线，需人工或远程重启

8.2.16 站点 fields 速查总表

字段键	类型	协议字段	是否参与动作闭环
TagValue	int	Node	否（仅节点映射）
goSite	int	—	否（Loop 调度用）
Fass2_StartStop	byte	StartStop	是
Fass2_PrecisionStop	bool	StartStop →22	是
Fass2_Byroad	byte	Byroad	是（非 0 时）
Fass2_Direction	byte	Direction	是（非 0 时）
Fass2_Orientation	byte	Orientation	是（非 0 时）
Fass2_Lift	byte	Lift	是（非 0 时）
Fass2_Roll	byte	Roll	是（非 0 时）
Fass2_Charge	byte	Charge	是（非 0 时）
Fass2_Obstacle	byte	Obstacle	是（非 0 时）
Fass2_Audio	byte	Audio	是（非 0 时）
Fass2_Light	byte	Light	是（非 0 时）
Fass2_Rest	byte	Rest	是（非 0 时）
Fass2_Clamp	byte	Clamp	是（非 0 时）
Fass2_Tray	byte	Tray	是（非 0 时）
Fass2_Shutdown	byte	Shutdown	是（非 0 时）
Fass2_WaitMode	string	—	预留

8.3 路径 Track fields 详细说明

路径 fields 主要影响**进入该站的那段边**（距离、速度、朝向、分叉）。

字段键	类型	映射	详细说明
Speed	float	Speed	边速度。≤1 按 **m/s**（协议值 = speed×600）；>1 按 **m/min**（协议值 = speed×10）。未配置默认 0.2 m/s
Reverse	bool	Orientation	true 时朝向 +180°
CarDirectionBias	float	Orientation	朝向偏置（度），叠加到几何朝向
EnableCarAbsoluteDirection	bool	Orientation	true 时用 CarAbsoluteDirection 代替几何计算
CarAbsoluteDirection	float	Orientation	绝对朝向（度，0–360）
Fass2_Byroad	byte	Byroad	边级分叉；站点未配 Fass2_Byroad 时生效
magSelect	byte	Byroad	磁导航 coder 分叉序号；Fass2_Byroad 为空时作为 Byroad 备选
Fass2_Direction	byte	Direction	边级车头旋转；站点未覆盖时生效
Fass2_Orientation	byte	Orientation	边级朝向；站点未覆盖时生效
Magnet	bool	—	true 启用 MagneticTrackCoder（agv.MagGo 等），与 MagFass2Go 模板独立
ReverseDst	int	—	倒车目的地站 ID；磁 coder 跳过该边

距离 Distance：由地图坐标自动计算（上一站与本站欧氏距离，mm 取整），一般无需手配。

8.4 组包优先级与默认规则

当同一协议字段在站点和边都有配置时：

站点 Fass2_* > 边 Fass2_* > 边通用字段（Speed/Reverse/magSelect）> 系统默认

** StartStop 决定逻辑**：

- 1. 若站点有 Fass2_StartStop → 直接用
- 2. 否则若是路径末站且 Fass2_PrecisionStop=true → 22
- 3. 否则若是路径末站 → 2
- 4. 否则 → 1（过站）

** Byroad 决定逻辑**：

1. 站点 Fass2_Byroad
2. 边 Fass2_Byroad
3. 边 magSelect
4. 0

****fields 变化重发****：任务执行中，若**尚未到达的站点** Fass2_* 签名变化，状态机会从当前站重新 BuildTaskPlan 并分配新 taskId。

8.5 典型场景配置示例

场景 A：通道过站（无动作）

站点 fields：**无需配置**（中间站默认 StartStop=1）

场景 B：普通停车工位

```
Fass2_StartStop = 2
```

场景 C：精准对接工位（末站）

```
Fass2_PrecisionStop = true
```

或

```
Fass2_StartStop = 22
```

场景 D：顶升装卸工位

```
Fass2_StartStop = 2
Fass2_Lift = 1      # 到站升起到位
```

离站下一任务可在目标站配 Fass2_Lift = 2 下降，或通过后续路径站点 fields 覆盖。

场景 E：充电工位

```
Fass2_StartStop = 2  
Fass2_Charge = 2      # 开始充电
```

场景 F：Y 型磁分叉口

****站点****（推荐在岔口站配）：

```
Fass2_Byroad = 1      # 选择磁条 1
```

或在该站****进入边****上配：

```
magSelect = 1  
# 或  
Fass2_Byroad = 1
```

场景 G：磁地标与站 ID 不同

****站点****：

```
TagValue = 10086
```

****车辆****：

```
UseTagValueAsNode = true
```

场景 H：Loop 默认下一跳

****站点 3****（车到 3 后若 tasklist 无匹配，自动去 5）：

```
goSite = 5
```

场景 I：完整工位（停车 + 顶升 + 精准）

```
TagValue = 2001
Fass2_StartStop = 22
Fass2_Lift = 1
Fass2_Obstacle = 2
```

配置禁忌（新手易错）：

错误	后果	建议
中间站误配 StartStop=2	路径中间不必要的停车	仅工位/末站停车
Lift=3/4（运动中）作为目标	动作可能永不完成	用到位值 1 / 2
TagValue 重复	节点映射错乱	全图唯一
只配站点不配 UseTagValueAsNode	下发节点号仍是 site.id	成对配置

9. LoopMission 环线任务

9.1 运行流程

```
1. 车到站 N，LoopMission 检测可用
2. 匹配 tasklist: currentStationId = N → 分配 targetStationId
3. 写 tags: goalSite = 目标站 ID
4. LoopStartAction 调用 GoSite
5. MagFass2Car 拦截 → 全程任务到 goalSite
6. 到达后 GoSite 清理 goalSite / occupied / loopAssigned
7. 下一轮任务匹配
```

9.2 相关 tags

标签	含义
goalSite	当前要去的目标站 ID
loopAssigned	分配时间戳
occupied	车辆占用中，值如 go5

9.3 与 Core 的关系

- ****不需要改 Core****; `LoopMission` 逻辑完全沿用
- `磁导航FASS2环线` 仅为场景标识, 行为与 `LoopMission` 相同
- 真正区别在 ****车型**** 是否 `MagFass2Car` 且 `EnableLoopTaskDrive=true`

9.4 关闭 Loop 拦截 (恢复多段脚本)

```
EnableLoopTaskDrive = false
```

Loop 仍走 `Compile` 多行 `MagFass2Go`, 每行一段边。

10. 任务状态机与协议下发

10.1 状态流转

```
Idle → Planning → Moving → AtStation → SegmentDone → ... → Complete
                                     ↓
                               Fault (急停/故障/超时)
```

10.2 单帧最多 10 站 — 不冲突

- ****规划****: 全程可能有 20+ 站, 存在内存 `Plan` 里
- ****下发****: 每次只发滑动窗口 `min(LockCount, 剩余, 10)` 站
- ****推进****: 当前站动作完成 → `CurrentIndex++` → 再发下一段
- ****同一 taskId****: 全程复用, 周期重发当前窗口

示例 (15 站路径, `LockCount=4`) :

```
第1次 0xB1: 站 1,2,3,4
到站 1 后: 站 2,3,4,5
到站 2 后: 站 3,4,5,6
...直到 goalSite
```

若车体需要首帧更多预发, 可将 `LockCount` 调至 ****10**** (不超过协议上限)。

10.3 动作闭环

到站后比对上报 25B 节点与期望:

场景	完成条件
过站	到达节点且非运行态（State≠1）
停车	StartStop 匹配
顶升/充电等	对应字段在状态帧中达标
未完成	自动补发 0xA1，按 ActionRetryIntervalMs 节流

10.4 车辆状态枚举（调试看）

枚举键	含义
TaskPhase	任务阶段（Idle/Moving/AtStation/...）
TaskGoal	目标站 ID
TaskIndex	当前站点索引
State	车体状态文字
CurrentNode	当前节点号
Task	车体上报的任务号

10.5 重启恢复（可选）

任务上下文会写入 car.tags（Fass2Task_* 前缀）。重启后 keepAlive 尝试恢复，需等车体再次 UDP 上报后状态机继续推进。

11. 状态查看与手动操作

车辆右键方法（界面中）：

方法	协议	用途
查询状态	0x00	TCP 模式主动查；UDP 模式看缓存
启动	0x01	启动车体
停止	0x02	停止
急停	0x03	急停
重置UDP监听	—	重新注册 UDP 会话
重置TCP连接	—	关闭长连接（TCP 模式）

日志前缀： [MagFass2Car:车名(id)]，任务状态机： task-sm。

12. 常见问题排查

Q1：车显示离线（UDP）

可能原因	处理
车体未上报	查车体是否发往 ListenPort
VehicleCode 不匹配	对照上报帧 Car 字段
超时	调大 UdpOfflineTimeoutMs 或查网络
IP/端口错误	核对 address、Port

Q2：发了 0xB1 车不动

可能原因	处理
未启动	确认 StartBeforeMove 或手动点启动
节点号不对	检查 UseTagValueAsNode 与 TagValue
TCP/UDP 模式搞反	确认 CommMode
状态机 Fault	看 TaskPhase、TaskFault

Q3：Loop 不分配任务

可能原因	处理
进程未启动	启动 LoopMission / 磁导航FASS2环线
tasklist 不匹配	currentStationId 要等于车所在站
车不可用	Commons.SelectCar 条件
已有 goalSite	等上一次完成

Q4: 到站但不继续

可能原因	处理
动作未完成	看 <code>Lift / StartStop</code> 等是否与配置一致
状态机 <code>AtStation</code>	等 <code>0xA1</code> 或检查车体动作
超时	调大 <code>MoveTimeoutSeconds</code>

Q5: Loop 和多段脚本哪个在生效?

看日志:

- `loop goal drive intercept` → 全程任务 (阶段 4)
- `script begin` + 多行 `MagFass2Go` → 多段脚本

Q6: 与 MagCar 混用注意事项

- `EventCarMission` **只控制 MagCar**
- 同场景两种车型时, `tasklist` 和车辆类型要对应
- FASS 1.0 用 `MagCar`, FASS 2.0 用 `MagFass2Car`

13. 参数推荐与检查清单

13.1 首次联调推荐配置

```
CommMode           = Udp
EnableTaskStateMachine = true
EnableLoopTaskDrive  = true
LockCount           = 4          (车体需更多预发可改 10)
StartBeforeMove      = true
EnableDetailLog       = true
LogRawFrame          = true
MoveTimeoutSeconds   = 120
```

13.2 上线前检查清单

- [] 插件 `StandardScene.Magnetic.dll` 已加载
- [] 车型为 `MagFass2Car` (不是 `MagCar`)
- [] `VehicleCode`、IP、端口与车体一致
- [] UDP 能收到 100B, 日志有 `0x10` 应答
- [] 关键站点 `TagValue` / `Fass2_*` 已配置

- [] LoopMission 已启动, tasklist.json 路径正确
- [] 单车手动 Go 一站成功后再开环线

14. 术语表

术语	解释
0xB1	路径/任务下发命令, 最多 10 个站点节点
0xA1	单站点动作下发
0x10	UDP 状态应答 (时间戳)
goalSite	LoopMission 给车分配的目标站标签
LockCount	每次下发的滑动窗口站数
TaskBuilder	将地图路径+fields 拼成节点序列
状态机	控制规划→下发→到站→推进的自动流程
PCB/UDP	车体主动上报, 调度被动接收
PLC/TCP	调度主动查询车体

附录：相关源码路径

模块	路径
车型	CarTypes/MagFass2Car.cs
协议	Protocol/Fass2Protocol.cs 、 Fass2UdpHub.cs
组包	Tasking/Fass2TaskBuilder.cs
状态机	Tasking/Fass2TaskStateMachine.cs
动作判定	Tasking/Fass2ActionResolver.cs
环线 Mission	Chained/MagFass2LoopMission.cs
站点字段常量	Tasking/Fass2SiteFields.cs

文档随代码演进, 若参数或行为有变更, 以源码为准。