

磁条交管逻辑与策略说明

MagCar 管控区排队、兼容组并行与扫车放行

适用对象：现场实施、调度配置、软件维护

配置文件：SimpleLite 工作目录 `Config/Signal/mag-control-areas.json`

进程名称：磁条交管进程（MagTrafficMission）

文档日期：2026-08-18

1. 文档目的与适用范围

本文说明 FASS 2.0 磁导航场景中，**磁条交管进程**如何根据管控区配置表对 **MagCar**（FASS 1.0 磁条车）进行停车等待、排队和放行。读者按本文即可完成十字路口、汇合点、单入口限流的配表，并理解扫车每一拍在做什么。

1.1 适用

- 车型仅 **MagCar**。类型名必须恰好为 MagCar。
- 车在配置的**管控触发站点**停稳后，由本进程发 FASS 1.0 启动指令（`StartMagCar`）。
- 配置表由迷穀数据中心「磁条管控区」或进程内「打开管控区列表」维护，保存到上述 JSON。

1.2 不适用

- **Mag2Car** 不走本进程。Mag2 仍使用地图字段 `Fass2_ControlArea` / `Fass2_ControlCapacity`，以及任务报文 `StartStop=12→11`，本进程不改 Mag2 报文。
- 内核 `TrafficControl.TryLock` 点锁、交通管制进程 `traffic.json`、环线任务的流量字段，与本表独立。
- 不负责让车在触发点停车。车必须先由任务/站点策略停在触发点，本进程只负责「满足条件后发启动」。

2. 总体策略

交管策略可以概括为四句话：

1. **按区绑定**：同一「管控区名称」的多行表示同一个物理路口（或汇合区）。
2. **按线占用**：每一行有自己的内部站点列表和容量，占用只统计本行站点，不把整个路口并成一份。
3. **冲突才互斥**：同区里，兼容组相同，或两行内部站点有交集，视为冲突，不能同时放；否则可并行。
4. **冲突才排队**：优先级、到达时刻、等待时长只在冲突线路之间生效。不冲突的线路互不插队、互不占名额。

兼容组留空且同区多行都为空时，视为同一组，**整区互斥**，与早期「同区容量 1 只放一辆」的行为兼容。旧 JSON 不填兼容组即可沿用。

3. 配置表（交管策略字段）

主键是**管控触发站点**。同一触发点多条启用记录只保留第一条。未启用（`IsUse=false`）的行保留在文件中但不参与判定。

字段	JSON 键	含义与配法
管控触发站点	TriggerSit	MagCar 必须停稳在此站点后才进入本行排队。应对齐现场停车点，不要把路口内部行驶点当成触发点。
管控区名称	AreaName	同名多行绑成一个路口，共用等待时长（取最长）。空则该触发点自成一区，相当于单入口限流。
兼容组	CompatGroup	同区里互不干扰的线路填 不同组名 （如东西 EW、南北 NS）。会交叉、会抢同一段路的填 同组 。空字符串与其他空组视为同组。
优先级	Priority	整数，越大越先放。干道高、支路低。只在冲突线路之间比较。
等待时长	WaitSeconds	秒。同区多行不一致时取 最长 。 ≤ 0 表示不做插队。用于防止支路被干道一直堵住。
管控区站点列表	ControlArea	本线路穿过路口的 内部站点 ，逗号分隔，如 10,11。不要把各方向排队触发点写进来。空则回退为触发点本身。
区内最多几辆	ControlCapacity	本行容量。小于 1 按 1 处理。统计范围：占用本行内部站点的车 + 与本行冲突、已发启动但未进区的在途车。
放行后启停字节	ControlReleaseStartStop	兼容字段。MagCar 放行发 FASS 1.0 启动指令，不使用该字节。
启用	IsUse	false 时本行不生效。

3.1 运行态（不入库）

名称	说明
到达时刻	进程第一次扫到该 MagCar 停在该触发点的时间。开走后再回来重新计时。不写入 JSON。
在途占额	已对该车发过启动，但尚未占用本行内部站点。避免「刚放行、车还在触发点」时下一拍又放一辆冲突车。

3.2 进程参数

参数	默认	说明
TickMs	1000	扫车周期（毫秒），下限按 200ms 夹紧。
PendingTimeoutSeconds	8	在途超时（秒）。超时后取消占额，防止启动失败把名额卡死。
AreasPath	Config\Signal\mag-control-areas.json	相对路径只解析 SimpleLite 工作目录，不读 plugins 目录。

4. 交管逻辑（扫车每一拍）

进程启动或点「重新加载配置」后，读取启用规则。此后约每秒执行一次 `ScanAndRelease`。

单拍流程

- 快照全场车辆：站点 (`siteID / GetLastSite`)、占用站点 (`siteID + holdingLocks`)、是否 `MagCar`、是否停稳。
- 启用规则按管控区名称分组（空名称则按「`#sit:触发点`」自成一区）。同区等待时长取各行之最长。
- 清理到达时刻：已离开原触发点的车重新计时。
- 清理在途：车消失、超时、或已经踩到本行内部站点 → 去掉在途标记（进区后改由占用统计）。
- 收集本区停在各触发点上的停稳 `MagCar`，记为等待队列；已在途的不再重复发启动。
- 按第 6 节规则排序，然后**从队头到队尾贪心**：本行有空位则发启动并登记在途；无空位则跳过，继续看后面不冲突的车。
- 放行动作：先尝试停充，再 `StartMagCar`（`FASS 1.0` 启动）。

4.1 何谓停稳

状态不是「运行中 / `Running`」。下列视为可排队：停止中、充电中、`WaitPass`，以及没有 `pendingLocks`、没有 `occupied` 标签的非运行状态。充电中的车放行前会尝试 `StopCharge`。

4.2 占用怎么数

对某一等待车辆，只统计占用**该行 `ControlArea`** 的他车。排队触发点上尚未放行的 `MagCar` 不计入占用（它们还在区外等候）。本车不计。

占用站点来源：车辆当前 `siteID`，以及 `status.holdingLocks`。任一命中列表即算 1 辆。

4.3 可放行数（空位）

空位 = 本行容量 - 占用本行站点的车 - 与本行冲突的在途车

空位 ≥ 1 才对该车发启动。容量 1 表示：本线路内部（含冲突在途）最多一辆。

在途车若已经进入本行内部站点，只算「占用」，不再重复算「在途」，避免一名额扣两次。

5. 冲突判定（能否并行）

只比较同一管控区名称下的两行（或一行与一条在途记录）。不同区互不影响。

满足下面任一条件即冲突（互斥）：

- 1. 兼容组相同（不区分大小写，去首尾空格）。两边都为空也算相同 → 整区互斥。
- 2. 两行的管控区站点列表有交集。

两组都不同、且站点列表不相交 → 不冲突，可同时放行。

场景	兼容组	内部站点	结果
旧表、未填兼容组	都空	任意	同区全部互斥
东西对向直行	都填 EW	可不相交	同组，互斥（对向若也要并行，应拆成不同组且站点不相交）
东西 / 南北互不交叉	EW 与 NS	10,11 与 20,21	不冲突，可并行
组名不同但抢同一段	EW 与 NS	都含 15	站点相交，仍互斥
左转与对向直行交叉	建议同组，或站点列出交叉段	有交集或同组	互斥

若四个方向都复制同一份内部站点（例如都写 10,11,12,13），则即使兼容组不同也会因站点相交而全部互斥。要对向并行，必须按方向拆开内部点，且不要把触发点写进列表。

6. 排队规则（仅冲突线路）

同一管控区内，先按下列比较函数排序，再贪心放行。不冲突的车排序先后不影响能否同时过，只影响扫描顺序。

6.1 插队（等待时长）

当同区等待时长 $W > 0$ ，且两车优先级不同：

若 **高优先级到达时刻 - 低优先级到达时刻 $> W$** ，则低优先级先走。

含义：低优先级更早到达，且已经比高优先级早到超过 W 秒，避免支路饿死。差值未超过 W 时，仍让优先级高的先走。

使用「 $>$ 」而不是「 $\geq$ 」。例如 $W=10$ ，恰好早到 10 秒时仍按优先级。

6.2 默认顺序

1. 未触发插队时：优先级数值大的在前。
2. 优先级相同：到达时刻早的在前。
3. 仍相同：车号小的在前（稳定次序）。

6.3 贪心，而不是只放队头

排序后从前往后看。队头因本线已满或冲突在途没有空位时，**跳过队头**，继续检查后面的车。这样南北已占用时，队头若是南北，不会挡住后面可以走的东西向车辆。

7. 放行与在途生命周期

1. 判定可放 $\rightarrow$ StopCharge（若有） $\rightarrow$ StartMagCar。
2. 登记在途：记录区名、触发点、兼容组、本行内部站点、放行时刻。
3. 下一拍该车仍停在触发点：不再重复启动，但作为冲突在途占额。
4. 车进入本行内部站点：清除在途，改为占用统计。
5. 超过 PendingTimeoutSeconds（默认 8 秒）仍未进区：清除在途，避免卡死。车若仍停在触发点，下一拍可再次进入排队。
6. 车离线或从系统消失：清除在途与到达时刻。

重新加载配置会清空到达时刻和在途表，请避免在路口有车等待时频繁重载。

8. 配置示例

8.1 十字路口：东西与南北并行

区名均为「十字路口A」。东西兼容组 EW，南北 NS。内部站点按方向拆开，容量均为 1。东西对向若现场确实会在同一段交汇，应把对向也拆站点或改为同组。

```
[
  { "TriggerSit": 8, "AreaName": "十字路口A", "CompatGroup": "EW", "Priority": 10, "WaitSeconds": 15, "ControlArea": "10,11", "ControlCapacity": 1, "IsUse": true },
  { "TriggerSit": 18, "AreaName": "十字路口A", "CompatGroup": "EW", "Priority": 10, "WaitSeconds": 15, "ControlArea": "12,13", "ControlCapacity": 1, "IsUse": true },
  { "TriggerSit": 9, "AreaName": "十字路口A", "CompatGroup": "NS", "Priority": 5, "WaitSeconds": 15, "ControlArea": "20,21", "ControlCapacity": 1, "IsUse": true },
  { "TriggerSit": 19, "AreaName": "十字路口A", "CompatGroup": "NS", "Priority": 5, "WaitSeconds": 15, "ControlArea": "22,23", "ControlCapacity": 1, "IsUse": true }
]
```

效果：东西一辆与南北一辆可同时过；同一兼容组内容量 1，同向或同组第二辆要等；南北优先级较低，但若已等待超过 15 秒且比干道更早到达，可以插队。

8.2 单入口限流（旧用法）

只配一行：触发点 8，区名为空，兼容组空，ControlArea 写内部点，容量 1。该点自成一区，他车占用内部点时不放行。

8.3 支路防饿死

干道 Priority=10，支路 Priority=1，同区 WaitSeconds=20。支路先到、干道后到且时间差大于 20 秒时放支路；否则放干道。

9. 现场操作

1. 确认已加载信号插件 `StandardScene.Signal.dll`，进程栏中有「磁条交管进程」。
2. 将 JSON 放到 SimpleLite 工作目录 `Config\Signal\mag-control-areas.json`。可用迷穀数据中心「磁条管控区」编辑（含兼容组列，需较新的 MiGu.Server）。
3. 添加并**启动**磁条交管进程。改表后点「重新加载配置」，或进程内打开列表保存后重载。
4. 状态栏可见：规则条数、触发点等待数、在途数、扫车拍数。
5. 诊断通道 `MagTraffic` 会记录放行：区名、兼容组、触发点、本线占用/容量、冲突在途。

交管不会替任务系统停车。请保证 MagCar 到达触发点时处于停止，否则不会进入排队，也不会发启动。

10. 与 Mag2 / 其他交管边界

机制	对象	关系
本进程 mag-control-areas.json	MagCar	触发点停稳后发 StartMagCar
地图 Fass2_ControlArea 等	Mag2Car	StartStop=12 等待，放行改本车报文，本进程不介入
TrafficControl 点锁	MagCar / Mag2 路径执行	锁站点资源，与本表并行存在
交通管制 traffic.json	区域互斥钩子	需单独启动「交通管制」进程

11. 常见问题

十字路口空着，四个方向却一起冲出去？

兼容组未区分，或内部站点写成同一串。请按方向拆兼容组并拆开 ControlArea。

四个方向谁也不走？

把四个停车触发点写进了 ControlArea，容量又为 1，等待车被当成区内占用。触发点不要写入内部列表。

改了 JSON 不生效？

必须点「重新加载配置」或重启进程。路径必须是工作目录 Config/Signal，不是插件目录。

Mag2 在同一路口不停不等？

本策略不管 Mag2。Mag2 需在地图配置 Fass2_StartStop=12 及管控区字段。

容量填 0？

当前实现小于 1 按 1 处理，不会变成不限流。

对向也要并行，但写成了同一兼容组？

同组即互斥。对向要并行：不同组名，且内部站点不相交。

12. 逻辑对照（维护用）

内容	位置
表字段	StandardScene.Signal / MagControlAreaModel
冲突、占用、容量	MagControlAreaGate
扫车、排队、在途、贪心放行	MagControlRuntime.ScanAndRelease
进程与重载	MagTrafficMission
迷穀列定义	MiGu.Server / SignalDataStore（表 id: mag-control）

本文描述 2026-08-18 代码行为。若后续调整兼容组或容量语义，以源码为准并同步修订本文。