基于 Modbus/TCP 的地铁 ISCS 与 ATS 接口设计与实现

周 崇[1]
ZHOU Chong

摘 要 针对地铁综合监控系统（ISCS）与列车自动监控系统（ATS）间的数据交互需求，文章设计并实现了一套基于 Modbus/TCP 的接口方案。明确了网络架构、通信协议及寄存器映射规则，开发了相应的数据解析程序。通过实际报文测试，验证了系统能够准确获取列车状态与阻塞信息，并成功触发环境与设备监控系统（BAS）联动，为地铁线路升级改造中的系统集成提供了可靠参考。

关键词 综合监控系统；自动列车监控系统；Modbus/TCP；接口集成；数据解析

doi：10.3969/j.issn.1672-9528.2026.05.036

0 引言

地铁轨道交通综合监控系统（integration supervisory control system, ISCS）通过接口协议，将电力监控系统（PSCADA）、环境与设备监控系统（BAS）、火灾报警系统（FAS）、闭路电视系统（CCTV）、广播系统（PA）、乘客信息系统（PIS）、站台门系统（PSD）、信号自动列车监控系统 ATS 等系统集成到一个软件平台，从而实现统一管理和运营调度[1]。而 ATS 是地铁信号控制系统中重要的组成部分，负责监控和指挥全线所有列车的运行。它通过自动调整和控制，确保列车能按计划安全、高效地行驶。

随着城市轨道交通技术的不断发展，地铁 ISCS 与 ATS 之间的融合深度，地铁建设和运营方的要求发生了变化。根据融合的深度主要分为两种（暂不考虑 ATS 集成 ISCS），第一种为接口互联，即综合监控系统与 ATS 利用通信接口实现信息交互。目前这种技术在国内地铁中较为普遍，比如上海轨道交通 10 号线，中央级综合监控系统与 ATS 进行列车的实时信息、阻塞及火警信息的交互[2]。青岛地铁 11 号线综合监控系统获取 ATS 的列车位置、区间阻塞信息以及列车组号信息，而 ATS 获取综合监控系统接触网带电信息[3]。第二种为综合监控系统深度集成 ATS，即综合监控系统与 ATS 共用一套硬件，综合监控系统提供对外接口与 ATS 系统交互，综合监控系统完全或部分集成子系统 ATS 的功能，例如北京地铁 6 号线、燕房线、大兴机场线、北京地铁 17 号线、太原地铁 2 号线等线路采用以行车指挥为核心的综合自动化系统，即综合监控系统与 ATS 深度集成[4]。

目前以上两种方式都在目前地铁项目中广泛应用。以接口集成方案最为成熟，因为两个系统硬件分别独立部署，耦合性相对较低，只要对接口定义准确且满足彼此需求就能很好地发挥两个系统的作用。而综合监控系统集成 ATS 的方案虽然是城市轨道交通发展的趋势，但是需要综合监控系统与 ATS 供应商之间紧密配合，更关键信号专业涉及行车安全，其安全等级更高，而综合监控系统子专业众多，调试难度加大，并且集成 ATS 后需要取得 SIL2 级的安全认证[5]。可见这种方式对于实施难度更大，挑战更高。 综上所述，本文采用接口互联的方式更加符合对城市轨道交通项目升级、改造和延线设计。

1 系统网络接口

综合监控系统与 ATS 之间接口主要在控制中心和车站，在控制中心两个专业通过网络通信接口互相连接，从而实现信息传递的目的。在车站中，ATS 主要与综合监控系统的 IBP 进行连接，而且其连接方式为硬线点位接线方式。所以不再描述车站部分的内容。那么，综合监控系统与 ATS 的中心网络接口界面如图 1 所示。

综合监控系统与 ATS 之间采用以太网接口 RJ45 接口，ATS 网线进入综合监控设备室内机柜配线架，然后通过冗余的前置机通信 FEP 接入综合监控 AB 网交换机系统，从而实现数据传输到综合监控的实时服务器。同一时刻，FEP 只有一个在工作，当检测到 FEP 故障或者宕机后，另一台备用 FEP 切换成主机模式，开始负责对来自其他系统的数据进行通信和数据处理。

2 通信协议

本项目综合监控系统与 ATS 采用 Modbus/TCP 通信协议，该协议是 Modbus 协议基于 TCP/IP 网络的实现形式，具备开

1. 沈阳新松机器人自动化股份有限公司 辽宁沈阳 110169

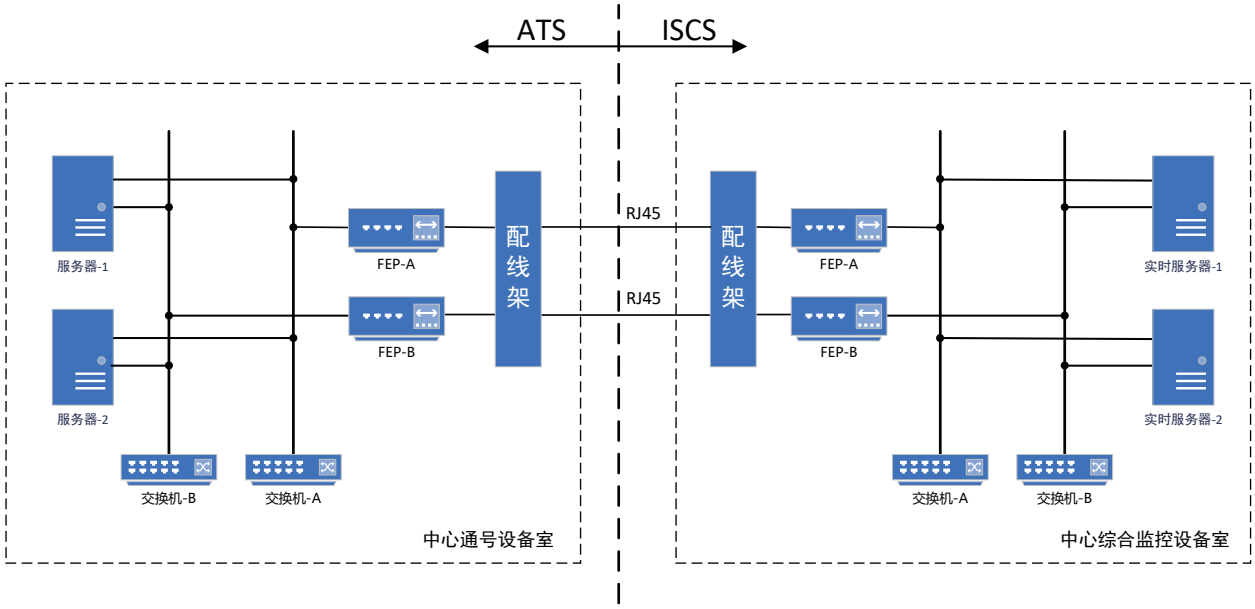

图 1 控制中心综合监控系统与 ATS 网络接口界面图

放、结构简洁、支持高并发通信等特点。其采用客户端 - 服务器模式，在传统协议数据单元前添加 7 字节 MBAP 报文头，利用 TCP/IP 协议栈实现可靠传输，无需额外校验字段 [6]。该协议支持 Ethernet II 与 802.3 帧格式，并使用 502 端口进行通信。在实际工业系统中，Modbus/TCP 广泛用于数据采集、设备监控与系统集成，如在城市轨道交通综合监控系统中实现中心与车站级的数据可靠交互 [7]。

在本设计中，综合监控系统作为 Modbus 通信主站（即客户端），ATS 作为 Modbus 通信从站（即服务端），端口号为 502。综合监控系统通过读输入寄存器（04H）从 ATS 系统中请求数据，通过写多个保持寄存器（10H）向 ATS 系统发送数据。Modbus/TCP 协议的数据传输单元由 MBAP 报文头和协议数据单元（PDU）构成，具体定义格式如表 1~2 所示。

表 1 ISCS 向 ATS 轮询服务器状态报文（功能码 04H）

描述		长度 / 字节	定义
报头	事务标识	2	TCP 顺序号
	协议标识	2	0x0000
	长度	2	0x0006
	单元标识	1	0xFF
Modbus 数据	功能码	1	0x04
	起始地址	2	0x0000（根据具体寄存器定）
	寄存器个数	2	0xXXXX（根据具体寄存器定）

通过两个系统预先定义好的通信协议地址，ATS 收到来自 ISCS 的状态轮询请求以后，将对应的地址信息反馈给 ISCS 系统，反馈的数据也是写在了协议中 Modbus 数据位中。

表 2 ISCS 向 ATS 发送状态信息（功能码 10H）

描述		长度 / 字节	含义
报头	事务标识	2	TCP 顺序号
	协议标识	2	0x0000
	长度	2	0x0013（根据具体寄存器数量计算）
	单元标识	1	0xFF
Modbus 数据	功能码	1	0x10
	Address 写起始地址	2	0x0258（根据具体寄存器分配定）
	寄存器个数	2	0x0006（同上）
	字节个数	1	0x0a 寄存器个数 ×2
	数据	12	0xXX…XX

同样当 ATS 收到来自 ISCS 系统的状态信息以后，也会馈给 ISCS，包括 ISCS 发送的起始地址和寄存器个数信息。

3 接口需求与功能

根据综合监控系统和 ATS 系统之间的运行和系统联动需求，ATS 系统传输给综合监控系统车辆信息和区间阻塞信号等信息，当综合监控系统接收到来自 ATS 的区间阻塞信息时，会通知 BAS 系统启动对应的通风和照明模式，同时，还有可能涉及与 CCTV、PA、PIS 的联动。此外，由于 ATS 与供电系统没有接口，则 ATS 系统需要向综合监控系统获取区间接触网供电信息。具体接口功能如表 3 所示。

表 3 综合监控系统（ISCS）与 ATS 互联信息表

序号	传输方向	传输内容
ISCS.ATS.01	ATS->ISCS	（1）列车在线数量 （2）列车编号 （3）列车类型 （4）列车方向 （5）区间信息 （6）是否阻塞
ISCS.ATS.02	ISCS->ATS	（1）区间信息 （2）是否带电

4 Modbus 协议寄存器分配原则

寄存器分配原则是针对 Modbus 协议的数据字段与接口功能要求所设计，旨在约束双方数据交互信息的地址定义，最终作为数据点表的生成依据。建立详细的寄存器分配原则能够节省双方接口开发时间，提高双方调试效率。具体分配原则如表 4 所示。

在表 4 中，从数据传输方向来看，分为综合监控系统从 ATS 上获取数据，另外一种是综合监控系统提供给 ATS 轨行供电分段分区，这部分用到的寄存器是 1001~1010，每个寄存器的 16 个位每 2 个位表示一个供电分区，其表示意义已在表中陈述。表中重点是 ATS 提供综合监控的列车数据，寄存器地址从 1 到 516，这些地址可以设定 100 列列车，能够满足大多数地铁每条线路（30~60 列列车的需求）。地址 1 表示在线列车数，随后每五个地址代表一列列车的信息，其信息包括列车组号（b）、列车 CBTC 状态（c）、列车编组号（d）、列车位置（e）、阻塞信息（f）、列车方向（g），代表意义也已经在表中给与描述。最后 ATS 还提供时间戳信息，以保证数据的准确性和可追溯性。

5 接口系统设计

5.1 系统业务流程

本系统采用 MODBUS/TCP 协议实现 ISCS 与 ATS 间的数据通信。如图 2 所示，ISCS 作为 MODBUS 主站通过 502 端口与 ATS 从站建立 TCP 连接，采用定时轮询机制每 1 s 发送读寄存器请求（功能码 04H）获取列车运行状态，解析数据后存储并检测阻塞以触发 BAS 的阻塞通风等模式。同时，当供电分区状态变更时，ISCS 主动发送写寄存器请求（功能码 10H）更新 ATS 系统数据。通信异常时自动重试，确保数据传输的实时性与可靠性，满足地铁运营监控需求。

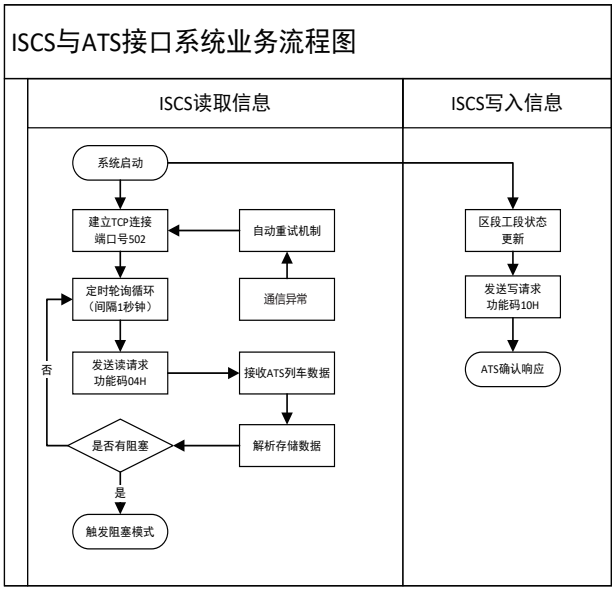

图 2 ISCS 与 ATS 接口系统业务流程图

表 4 综合监控系统（ISCS）与 ATS 互联信息表

传输方向	寄存器地址	高字节								低字节							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATS->ISCS (04H)	1	（a）在线列车数															
	2	（b）第 1 列列车车组号（例如：车组号为 123，对应 0x007B）															
	3	（c）第 1 列列车 CBTC 状态，bit(0)=1 为 CBTC 列车，bit(0)=0 非 CBTC 列车															
	4	（d）第 1 列列车编号车组号（例如：车组号为 021，对应 0x0015）															
	5	（e）第 1 列列车位置（2 字节，例如 0x0101，需要对照表找到 0x0101 对应区段）															
	6	（g）列车方向 bit(8)=0 未知，bit(8)=1 左行，bit(8)=2 右行								（f）阻塞信息 bit(0)=1 列车停车，bit(0)=0 列车正常							
	7~501	第 2 列到第 100 列的列车信息															
	502~510	预留位															
	511~516	时间戳地址（精确到秒），例如，2024 年 12 月 15 日 14 时 30 分 45 秒															
ISCS->ATS (10H)	1001	(i-8)		(i-7)		(i-6)		(i-5)		(i-4)		(i-3)		(i-2)		(i-1)	
	1002~1009	2 个位用二进制表示一个轨行区，共计最大 80 个轨行区；01 －区域带电；10 －区域未带电；00 和 11 －未知状态															
	1010	(i-80)		(i-79)		(i-78)		(i-77)		(i-76)		(i-75)		(i-74)		(i-73)	

5.2 软件代码实现

本设计采用的是 C# 高级编程语言，该语言是由微软开发的面向对象编程语言，具备类型安全、自动内存管理和丰富类库支持等特性，由于本系统中用于实现 MODBUS/TCP 通信协议解析和实时数据处理，其高效的位运算能力和多线程支持确保了列车运行数据处理的准确性与系统响应的实时性，满足了工业监控系统对可靠性和稳定性的要求。下图是关于解析 ATS 数据的代码函数。

通过图 3 可以看出，该函数依据表 4 的寄存器地址映射规范，将 MODBUS/TCP 协议传输的原始二进制寄存器数据转换为结构化的列车运行信息。通过位运算提取车组号、CBTC 状态、位置编码及阻塞状态等关键字段，实现列车实时运行状态的精确解析。

该解析模块为综合监控系统提供准确的列车位置和阻塞状态信息，当检测到列车阻塞时，立即触发 BAS 系统启动相应的阻塞模式，如隧道通风系统调整，确保运营环境的安全与稳定。

6 测试与验证

系统设计完成后利用抓取报文的方式来验证系统通信的准确性和有效性，由于 ISCS 发送数据较为单一，这里只验证收 ISCS 请求的数据。从日志文件中，获取到 ISCS send 请求报文为：00 01 00 00 00 06 FF 04 00 00 02 04。这里请求了 516 个寄存器地址的数据。ISCS rececive 接收部分报文为：00 01 00 00 04 0B FF 04 04 08 00 02 00 7B 00 01 00 15 01 01 02 00 00 80 00 00 00 16 01 02 01 01 … … 07 E8 00 0C 00 0F 00 0E 00 1E 00 2D（省略号为：第 12 到第 510 个寄存器，共 499 个寄存器，全部为 0000）。那么通过接收到的报文解析看出目前在线有 2 趟列车。其基本信息如下：第 1 列为非 CBTC 列车，右行，且未阻塞，第 2 列为 CBTC 列车，左行阻塞。上述测试结果表明，系统请求与响应报文完全符合表 1、表 4 所定义的协议规范，程序能够正确解析出在线列车数、列车状态及阻塞信息，验证了通信链路与数据解析逻辑的正确性。

```
/// <summary>
/// 解析ATS系统传输的列车数据
/// </summary>
public static List<TrainInfo> ParseTrainData(ushort[] rawRegisters)
{
    var trainList = new List<TrainInfo>();

    if (rawRegisters == null || rawRegisters.Length < 1)
        return trainList;

    // 寄存器地址1：在线列车数量
    int onlineTrainCount = rawRegisters[0];

    // 每列列车占用5个寄存器（地址2-501）
    for (int i = 0; i < onlineTrainCount; i++)
    {
        int baseIndex = 1 + i * 5;

        if (baseIndex + 4 >= rawRegisters.Length)
            break;

        // 解析单列列车信息
        TrainInfo train = new TrainInfo
        {
            Index = i + 1,
            TrainGroupNumber = rawRegisters[baseIndex],                    // 车组号
            IsCBTCTrain = (rawRegisters[baseIndex + 1] & 0x0001) == 0x0001, // CBTC状态
            TrainNumber = rawRegisters[baseIndex + 2],                     // 列车编号
            PositionCode = rawRegisters[baseIndex + 3],                    // 位置编码
            IsBlocked = (rawRegisters[baseIndex + 4] & 0x0001) == 0x0001,  // 阻塞状态
            Direction = ParseDirection((ushort)((rawRegisters[baseIndex + 4] >> 8) & 0x0003)) // 运行方向
        };

        trainList.Add(train);
    }

    return trainList;
}
```

图 3 解析 ATS 函数数据代码示例

7 结论与展望

本文设计并实现了一套基于 MODBUS/TCP 协议的 ISCS 与 ATS 接口系统，通过清晰的协议定义与寄存器映射，实现了列车状态与供电信息的可靠交互。测试验证了接口能够准确解析 ATS 数据并触发 BAS 联动，证明了该方案在接口互联场景下的可行性与稳定性，为地铁线路升级改造提供了实用参考。

与此同时，当列车阻塞时不仅触发 BAS 系统，还可联动 PIS、PA、CCTV 等更多专业子系统。这需要 ATS 提供更丰富的列车数据（如目的地、载客量等）作为支撑。同时，探索更高性能的通信协议以适应更复杂的数据交互需求，也是后续研究的重要方向。

参考文献：

[1] 郑宁，裴加富，林立，等．城市轨道交通综合监控系统与 CCTV 集成方案研究 [J]. 铁道通信信号 ,2020,56(11):79-81.

[2] 陈晨，钱存元，李慕君．全自动无人驾驶系统信号子系统的接口 [J]. 城市轨道交通研究 ,2019,22(9):146-148.

[3] 高云超．城市轨道交通 ATS 系统外部接口日志分析 [J]. 铁路通信信号工程技术 ,2021,18(1):76-79.

[4] 张红欣，王亚涛．轨道交通全自动运行线路综合监控和 ATS 集成与互联分析 [J]. 隧道与轨道交通 ,2019(4):10-12.

[5] 张华刚．综合监控与信号 ATS 融合方案分析及建议 [J]. 科技创新与应用，2023, 13(18): 153-156.

[6] 周树桥，于晖，黄晓津．基于 Modbus 协议的通信设计及调试方法研究 [J]. 自动化仪表，2023, 44(S1):158-164.

[7] 高秀兰．基于 Modbus TCP/IP 通讯综合监控系统的实现 [J]. 仪表技术与传感器，2015(10): 104-106.

【作者简介】

周崇（1983—），男，辽宁铁岭人，硕士，工程师，研究方向：城市轨道交通综合监控系统、数字化工厂等。

（收稿日期：2025-12-18 修回日期：2026-04-30）