Software Engineering and Applications 软件工程与应用, 2025, 14(6), 1209-1218
Published Online December 2025 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2025.146107

Hans汉斯

基于PLC1200的一种轻量级G代码数据交互方法

于凯圳，王　海

沈阳理工大学机械工程学院，辽宁 沈阳

收稿日期：2025年10月29日；录用日期：2025年12月5日；发布日期：2025年12月16日

摘　要

针对西门子S7-1200PLC在工业场景中G代码传输效率低、人工操作误差大、专用组态软件成本高的问题，本文提出一种基于Qt与S7协议的轻量级G代码数据交互方法。该方法以Qt为开发平台，依托Snap7库封装S7通信协议基础模块，构建包含PLC连接区、DB写入区、Bool监控区、日志区及工具栏的模块化上位机界面，实现G代码文件的加载解析、PLC数据块的三行联动写入及控制位自动监控功能。实现了G代码与PLC数据块的高效、可靠交互。该方法无需依赖昂贵组态软件，降低了中小制造企业产线数字化升级的成本与门槛，为车间层设备数据集成提供了实用技术方案。

关键词

S7-1200 PLC，G代码，数据交互

A Lightweight G-Code Data Exchange Method Based on PLC1200

Kaizhen Yu, Hai Wang

School of Mechanical Engineering, Shenyang University of Technology, Shenyang Liaoning

Received: October 29, 2025; accepted: December 5, 2025; published: December 16, 2025

Abstract

Addressing the challenges of low G-code transmission efficiency, high manual operation error rates, and costly specialized configuration software for Siemens S7-1200 PLCs in industrial settings, this paper proposes a lightweight G-code data interaction method based on Qt and the S7 protocol. Utilizing Qt as the development platform and leveraging the Snap7 library to encapsulate the S7 communication protocol base module (S7_BASE class), this method constructs a modular host-computer

文章引用: 于凯圳, 王海. 基于 PLC1200 的一种轻量级 G 代码数据交互方法[J]. 软件工程与应用, 2025, 14(6): 1209-1218.
DOI: 10.12677/sea.2025.146107

interface. This interface includes PLC connection, DB write, Bool monitoring, log, and toolbar sections, enabling G-code file loading and parsing, three-line synchronized writing of PLC data blocks, and automatic control bit monitoring. It achieves efficient and reliable interaction between G-code and PLC data blocks. This approach eliminates reliance on costly configuration software, reducing the cost and barriers for small and medium-sized manufacturing enterprises to upgrade their production lines digitally. It provides a practical technical solution for integrating equipment data at the shop floor level.

Keywords

S7-1200 PLC, G-Code, Data Interaction

1. 引言

随着工业 4.0 与智能制造理念的深度渗透，制造业正加速向数字化、网络化、智能化转型，生产全流程的数据贯通已成为突破柔性生产瓶颈、实现精益化管控的核心前提[1]。西门子 S7-1200 系列 PLC 凭借其高可靠性、灵活扩展性及适配中小型自动化场景的性价比优势，已广泛应用于电子制造、机械加工、新能源等领域的产线控制中[2]。然而，在实际工业场景中，传统的数据交互方式仍存在显著局限：一方面，依赖人工通过 PLC 编程软件手动输入参数，不仅效率低下，单次仅能处理少量数据，更易因人为操作误差导致生产异常；另一方面，专用组态软件虽能实现部分自动化，但高昂的授权成本与复杂的二次开发门槛，对中小型制造企业而言经济性不足，且难以灵活适配多行有序文本的流式写入需求[3]。

目前工业场景中 PLC 与 G 代码的交互方案主要包括三类：一是基于 Python、Java 等语言的开源库，这类方案需手动编写大量通信底层代码，开发门槛较高，且跨平台兼容性易受依赖库限制；二是专业测控软件的 PLC 通信模块，虽功能强大，但软件授权费用高昂，且集成 G 代码文件处理需复杂的模块搭建；三是主流 SCADA/HMI 软件的脚本功能，这类软件需与特定品牌 PLC 绑定，二次开发灵活性不足，且整体部署成本对中小企业不友好[4] [5]。上述方案或存在开发门槛高、成本高、兼容性差等问题，难以满足中小制造企业低成本、轻量化的数字化升级需求。

本研究针对特定场景下的数据集成问题，提供了一种经过实践验证的低成本高效益解决方案，降低了自动化系统数据配置的复杂度与门槛。通过将储存大量 G 代码的 txt 文件直接与 PLC 的 DB 块进行简洁、轻量级数据交互，稳定实现 G 代码文本数据向 S7-1200PLC 的自动化传输，为中小制造企业的产线数字化升级提供实用的技术参考与落地案例。

2. 开发工具及平台

Qt 跨平台界面应用开发基于 C++面向对象编程语言设计图形用户界面，其平台项目开发可以使 GUI 图形界面程序、控制台程序以及服务器相关程序，使用元对象编译器以及一些宏定义来筑造面向对象的框架，其易扩展性以及组件编程的功能深受开发者青睐。Qt 兼容性非常好，该平台能编译的源码在很多其他不同操作系统上也能直接编译运行，不需要过多修改，并且会根据不同系统本身的特性而生成该平台特有的显示效果[6]。Qt 特有的信号与槽机制打破了传统的 callback 信息传递机制，能够保证参数传递的有效性与正确性。经过长时间的维护与升级，Qt 有着非常丰富的基于 C++的图形库，其集成封装库里

有数据库、脚本库、XML 库以及 OpenGL 库等，这些封装库可多线程操作使得 Qt 平台有着非常强大的功能，具备开发大型工程项目的能力[7]。

在功能支持上，Qt 提供了完善的工具类库，涵盖字符串处理(QString)、文件操作(QFile)、定时器(QTimer)、对话框(QMessageBox)等，能够满足工业控制场景中数据交互、用户交互、流程控制等核心需求。同时，Qt 的跨平台特性确保了软件可在 Windows、Linux 等操作系统上无缝移植，为后续系统的扩展部署提供了便利[8]。

3. 程序开发设计

3.1. 界面开发设计

上位机界面是操作人员与系统交互的核心载体，其设计需遵循“功能清晰、操作简便、反馈及时”的原则。本设计的界面采用模块化布局，通过 setupUI 函数实现组件创建与排布，窗口尺寸设为 1000 × 800 像素，标题为“G 代码解析 V1.0”，确保操作人员可快速定位所需功能。界面布局采用横向分区 + 纵向整合的方式，分为 PLC 连接区、DB 写入区、Bool 监控区、日志区及工具栏五大功能模块，各模块通过 QHBoxLayout (横向)与 QVBoxLayout (纵向)管理器实现自适应排布，具体结构如图 1 所示。

Figure 1. Interface structure diagram
图 1. 界面结构图

(1) PLC 连接区

PLC 连接区位于界面顶部，是实现上位机与西门子 PLC 通信的入口，包含参数输入与操作按钮两类组件。布局上，采用 QHBoxLayout 横向排列组件，通过 addWidget 依次添加标签(“IP:”“机架:”“插槽:”)、输入框和按钮，并使用 addSpacing(20)在按钮前增加间隔，addStretch()填充右侧空白，使布局紧凑且美观。该区域的核心作用是简化 PLC 连接配置流程，操作人员无需编写代码，即可通过界面输入参数完成通信建立。

(2) DB 写入区

DB 写入区用于配置 G 代码写入 PLC 数据块的存储参数，位于 PLC 连接区下方。布局同样采用 QHBoxLayout，通过标签(“上一行 DB:”“起始字节:”等)明确参数含义，各组参数间通过 addSpacing(10)分隔，按钮前增加 addSpacing(20)突出操作入口。该区域的设计考虑了工业现场的实际需求——G 代码执行需依赖“上一行 - 当前行 - 下一行”的上下文关联，因此需同时配置三组存储地址，确保 PLC 能按顺序解析代码。

(3) Bool 监控区

Bool 监控区位于 DB 写入区下方，用于配置 PLC 控制位的监控参数，实现 G 代码的自动写入触发。布局采用 QHBoxLayout，标签与组件一一对应，monitorCheck 通过 addSpacing (20)与左侧参数区分隔，突出功能开关的视觉权重。该区域的设计体现了“PLC 主动触发”的工业控制逻辑——当 PLC 的特定布尔位为 true 时，上位机自动写入 G 代码，无需人工干预，适用于自动化生产节拍控制。

(4) 日志区与工具栏

日志区是界面的核心反馈区域，位于中下部，采用 QPlainTextEdit 组件(logEdit)，设置为只读模式(setReadOnly (true))，用于实时显示系统运行状态。日志内容包括：连接状态(如“[PLC]连接成功”)、文件操作(如“[系统]文件加载完成，共 20 行”)、数据写入结果(如“[PLC]成功写入三行：DB8.610 = ‘G01 X100’”)等，每条日志自动换行，便于操作人员追溯系统行为。

工具栏位于窗口顶部菜单栏下方，通过 addToolBar 创建，包含两个 QAction：“打开 txt 文件”关联 openFile 函数，用于加载 G 代码文本；“重新解析”关联 parseFile 函数，用于重新处理已加载的文件，同时工具栏的设计也简化了高频操作的入口。

3.2. S7 通信协议基础模块设计与实现

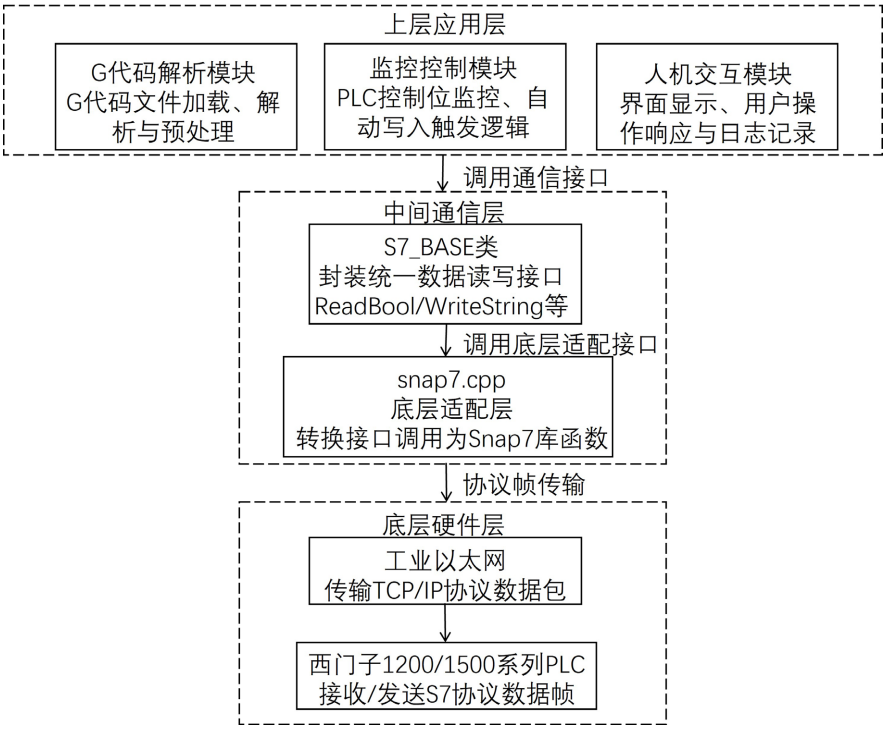


Figure 2. S7 communication module positioning diagram
图 2. S7 通信模块位置关系图

在工业自动化控制系统中，上位机与 PLC 之间的可靠通信是实现数据交互与控制指令传输的核心环节。本系统针对西门子 1200/1500 系列 PLC 的通信需求，设计了基于 Snap7 库的 S7 通信协议基础模块(S7_BASE 类)，并通过 snap7.cpp 文件封装底层接口，形成了一套完整的通信解决方案。

S7_BASE 模块的核心设计目标是依托 Snap7 库的底层能力，构建一个适配西门子 PLC 的通用通信层，其功能定位体现在三个方面：一是通过 snap7.cpp 对 Snap7 库的核心函数(如连接管理、数据读写)进行封装，将 C 语言风格的库接口转换为符合 Qt 框架规范的 C++类方法；二是支持工业控制中常用的数据类型(BOOL、INT、FLOAT、STRING、CHAR)与存储区域(DB、I、Q、M 等)的操作，满足不同控制场景的需求[9]；三是处理协议交互中的细节(如字节序转换、数据格式校验)，降低上层应用的开发复杂度。该模块在系统架构中处于中间层位置，上接上位机的业务逻辑(如 G 代码写入、控制位监控)，下通过 snap7.cpp 调用 Snap7 库与 PLC 硬件交互，形成“应用层–通信层–硬件层”的三层架构，其中 snap7.cpp 作为桥梁，负责将 S7_BASE 类的抽象接口转换为 Snap7 库的具体函数调用，其位置关系如图 2 所示。

3.3. 数据读写功能实现

S7_BASE 模块的数据读写功能采用基础字节操作 + 类型封装的分层设计，底层通过 ReadBytes 与 WriteBytes 方法调用 snap7.cpp 中的 Cli_ReadArea 与 Cli_WriteArea 函数，上层基于此封装各类数据类型的专用接口。ReadBytes 与 WriteBytes 方法是所有数据操作的基础，支持任意存储区域的字节流读写，通过 area 参数指定存储区类型(如 S7AreaDB 表示数据块)，dbNumber、startByte 与 size 参数确定操作地址与长度。这两个方法通过 snap7.cpp 与 Snap7 库交互，将上层的抽象数据请求转换为具体的协议帧发送，同时处理底层返回的状态码，确保数据传输的可靠性。

针对工业控制中常用的数据类型，S7_BASE 类封装了专用读写接口，其实现逻辑均依赖 snap7.cpp 的底层支持：BOOL 类型通过位运算从 1 字节数据中提取或修改目标位，适用于控制信号的读写；INT 类型通过 qFromBigEndian/qToBigEndian 函数处理西门子 PLC 的大端字节序，确保整数数据的正确解析；FLOAT 类型通过指针类型转换实现 4 字节数据与浮点值的映射，适用于模拟量数据；STRING 类型需处理前 2 字节的长度标识(最大长度与当前长度)，通过 snap7.cpp 的字节操作函数完成字符串的截取与填充；CHAR 类型直接读写 1 字节数据，适用于单个字符传输[10]。各类数据类型的读写特性如表 1 所示。

Table 1. Read/write characteristics of various data types
表 1. 各类数据类型的读写特性表

数据类型	读取方法	写入方法	数据长度	关键处理逻辑
Bool	ReadBool	WriteBool	1 位	位运算提取/修改目标位
Int	ReadInt	WriteInt	2 字节	大端/小端字节序转换
Float	ReadFloat	WriteFloat	4 字节	字节序转换 + 指针类型转换
String	ReadString	WriteString	可变长度	长度字段处理 + 字符截断
Char	ReadChar	WriteChar	1 字节	直接字节转换

3.4. G 代码文件处理模块

G 代码文件处理通过 openFile 与 parseFile 函数实现文件加载，解析，标准化的全流程，为写入 PLC

提供统一格式的指令数据。openFile 函数的设计充分考虑工业场景下的文件多样性，通过 QFileDialog 实现可视化文件选择，过滤条件严格限定为“文本文件 (.txt); 所有文件(.*)”，既保证仅加载合法格式的 G 代码文件，又保留对特殊格式文件的兼容能力。文件选择后，函数通过 QFile 以只读+文本模式打开文件，借助 QTextStream 的逐行读取机制将内容载入 QStringList lines 容器——这种线性存储结构既便于按行索引，又能直接映射 PLC 对 G 代码的逐条执行逻辑。读取过程中，针对可能出现的文件损坏、权限不足等异常，函数通过 QMessageBox 弹出明确警告(如“无法打开文件！”)，同时在日志区记录错误详情，便于操作人员快速定位问题；若读取成功，则输出“[系统]文件加载完成，共 X 行”(X 为 lines. Size ())，并将 currentLine 初始化为 0，形成加载–计数–定位的连贯反馈。

parseFile 函数则聚焦于 G 代码的标准化处理，通过遍历 lines 容器实现批量解析。其核心逻辑依赖 processLine 函数完成文本净化：首先通过 trimmed ()去除首尾空格与换行符，避免因格式冗余导致 PLC 解析异常；再通过 toUpper ()将所有字符转换为大写，统一指令格式(如将“g01 x100 y200”规范为“G01 X100 Y200”)，消除大小写不一致对 PLC 识别的干扰。对于空行或注释行(以“;”开头)，函数会自动标记为“[空行/注释]”，既不影响解析流程，又在日志中明确标识，便于追溯原始文件结构。解析过程中，函数通过“[行 X]处理后文本”的格式实时输出日志(X 为行号 +1)，并调用 QCoreApplication:: processEvents ()强制刷新界面，确保在处理数千行 G 代码时日志不卡顿。遍历结束后，函数重置 currentLine=0，为写入模块提供清晰的起始基准。该模块通过标准化处理确保 G 代码格式统一(如“g01”转为“G01”)，适配 PLC 解析需求；实时日志输出便于操作人员追溯处理过程，批量解析能力满足工业场景中大量 G 代码的预处理需求。其流程如图 3 所示。

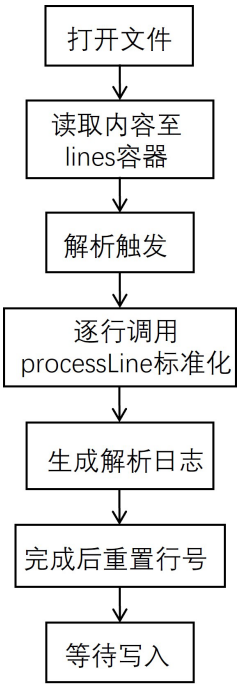


Figure 3. G-code file processing flowchart
图 3. G 代码文件处理流程图

3.5. G 代码写入与 PLC 监控逻辑

G 代码写入与 PLC 监控是数据交互的核心，通过 writeCurrentLine (手动)与 monitorBool (自动)函数

实现，依托 S7_BASE 模块完成数据传输。writeCurrentLine 函数的写入逻辑以“三行联动”为核心，充分考虑 PLC 执行 G 代码时对上下文关联性的依赖。函数首先进行双重前置校验：若 PLC 未连接(plcConnected 为 false)，立即弹出“请先连接 PLC！”的警告，避免无效通信；若 currentLine ≥ lines.size () (所有行已写入)，则提示“全部解析完成！”，防止越界操作。校验通过后，函数读取 DB 写入区的参数(上一行/当前行/下一行的 DB 号与起始字节)，通过 toInt (&ok)进行有效性判断——对非数字输入自动补全默认值(DB 号 1、起始字节 0)，确保写入地址合法。这种设计使 PLC 在执行当前指令时可提前获取上下文，避免因信息不完整导致的执行中断。写入过程中，函数通过 S7_BASE:: WriteString 接口将文本写入指定 DB 区域，严格限制字符串最大长度为 20 字节，并根据返回值判断写入结果：若三行均成功，日志输出“[PLC]成功写入三行：DB%1.%2 = ‘%3’, ...”，同时 currentLine 递增 1；若任一失败，输出“[PLC]写入失败”并弹窗警告，确保操作人员及时干预。

PLC 监控逻辑则通过 monitorTimer 定时器(500 ms 间隔)与 monitorBool 函数实现自动化触发，核心是对 PLC 控制位的实时监测与响应[11]。monitorTimer 的启停由“启动 plc 控制解析”复选框控制——勾选时定时器启动，周期性调用 monitorBool；取消勾选时定时器停止，兼顾自动化与手动控制需求。monitorBool 函数首先检查 PLC 连接状态，未连接则直接返回；连接正常时，读取 Bool 监控区参数(控制位 DB 号、起始字节、位位置)，同样对无效参数补全默认值。随后调用 S7_BASE:ReadBool 读取指定布尔位状态，若为 true (PLC 发出触发信号)，立即执行三项操作：调用 writeCurrentLine 写入当前行 G 代码、调用 S7_BASE:WriteBool 将控制位复位为 false (防止重复触发)、日志输出“[PLC]监控触发，布尔已复位 DB%1.%2.%3”(含具体位地址)，其逻辑如图 4 所示。

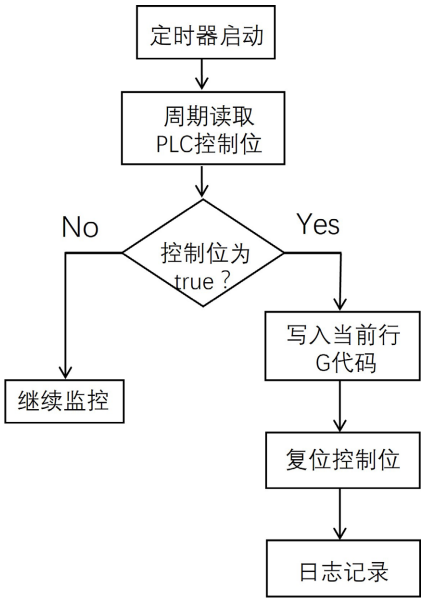


Figure 4. PLC monitoring logic flowchart
图 4. PLC 监控逻辑流程图

4. 实现与验证

虚拟 PLC 的搭建基于 S7-PLC SIM Advanced V3.0 软件实现，为满足系统软件在环调试需求，需将在线访问模式配置为 PLC SIM 模式。当“1 Active PLC Instance (s)”下方的实例指示灯显示为绿色时，表明虚拟 PLC 已进入运行状态。虚拟机 PLC 设置如图 5 所示。

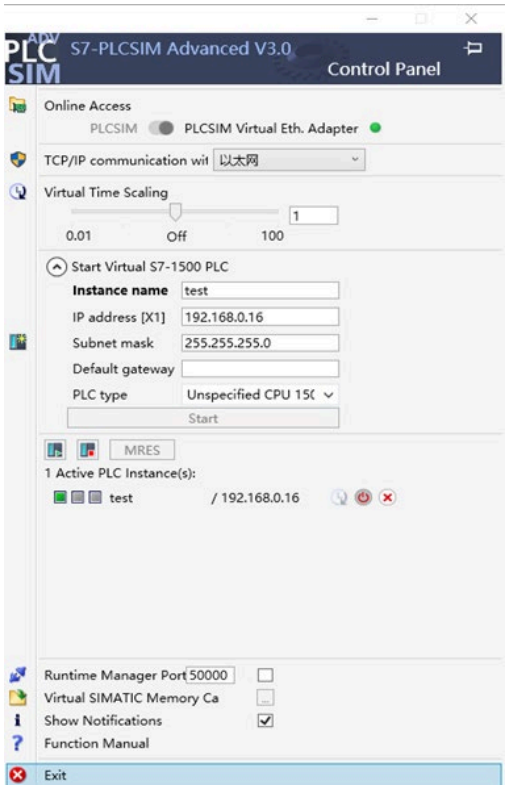


Figure 5. Virtual PLC configuration
图 5. 虚拟 PLC 设置

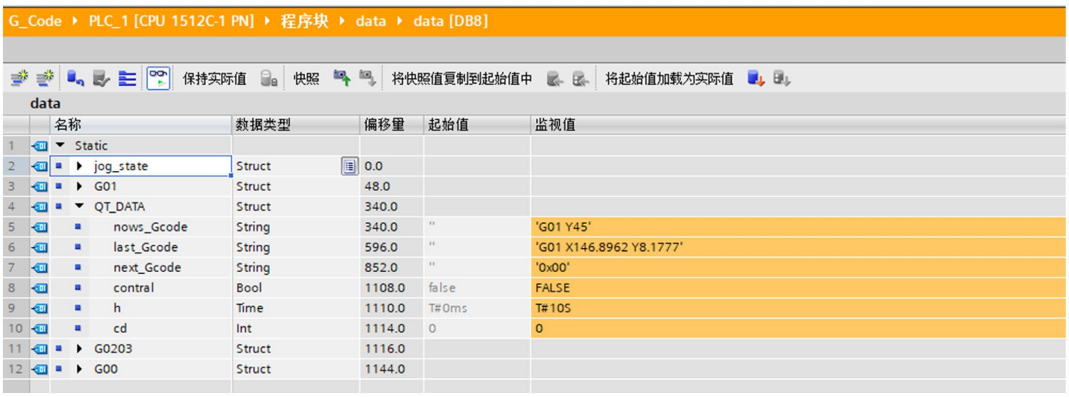
Figure 6. Initial state
图 6. 初始化状态

根据虚拟 PLC 的地址完成该上位机与 PLC 的通讯连接，此时上位机日志区会显示[PLC]连接成功，然后在工具栏打开 txt 文件，本文使用的 G 代码文本是由 CAM 导出的一个凸台的 G 代码文本，以这个文本进行测试，将这个文件在上位机中打开，打开后日志区显示“[系统]文件加载完成，共 530 行”，此时启动 PLC 控制解析，初始化完成，此时初始化完成状态如图 6 所示。

通过读取指定 Bool 为的状态，PLC 触发信号就写入三行 G 代码后自动复位，不断地触发与复位，实现大量 G 代码的快速读写进 PLC 对应的 DB 块内，当所有 G 代码都传输完毕后，界面自动跳出全部解析完成，调试过程如图 7 所示。

(1) PLC 数据块数据交互；

(2) 上位机数据传输调试。



(1) PLC数据块数据交互

(2)上位机数据传输调试

Figure 7. Debugging process
图 7. 调试过程

5. 结论

本文针对西门子 S7-1200 系列 PLC 的 G 代码数据交互需求，设计实现了基于 Qt 与 S7 协议的轻量级交互系统，核心成果包括构建基于 Snap7 库的 S7_BASE 通信模块，封装 BOOL、INT 等多数据类型统一读写接口、集成 PLC 连接、G 代码解析与三行联动写入的模块化上位机界面，以及提出 PLC 控制位主动触发的自动化写入逻辑，测试中 530 行 G 代码传输成功率达 100%，较人工输入效率提升 80%以上，且无需昂贵组态软件，依托跨平台特性降低了中小制造企业数字化成本；但系统仍存在局限，仅适配西门子 S7 系列 PLC、缺乏 G 代码语法校验、日志无持久化存储且仅支持 txt 格式加载，未来将通过扩展 Modbus 等通用协议适配多品牌 PLC、新增正则表达式语法校验、集成 SQLite 数据库实现日志追溯、支持 nc/gcode 多格式加载及开发移动端监控界面进一步优化，该系统可广泛应用于金属加工、模具制造等场景，具备较强工程实用价值。

参考文献

[1] 王珍珍, 张庆磊, 王传刚, 等. 基于 Qt 的远程监测系统客户端设计与实现[J]. 电子科技, 2015, 28(8): 149-152.

[2] 吴树红, 沈柳柳, 谭枫, 等. 基于 PLC 及数据库运算的松套管纤膏流量计量控制系统[J]. 电线电缆, 2024, 67(6): 66-70.

[3] 焦敬波, 王昆, 谢荣灿, 等. 基于 Snap7 的盾构实时监控系统设计[J]. 电子技术与软件工程, 2020(9): 59-63.

[4] 罗光耀, 张龙刚, 俞瑞昕. LabVIEW 与 S7-1200 系列 PLC 基于 Modbus TCP/IP 协议的通信方法[J]. 塑料包装, 2015, 25(5): 26-29.

[5] Li, G., Li, W.J., Su, Q.Q., Sun, C.F. and Ge, Z.Y. (2014) Monitoring and Control System of Underground Coal Gasification Based on Industrial Ethernet and PLC. *Applied Mechanics and Materials*, **496**, 1376-1380. https://doi.org/10.4028/www.scientific.net/amm.496-500.1376

[6] 魏学舟, 刘涛. 基于 Snap7 的 PLC 上位机监控软件开发[J]. 设备管理与维修, 2018(14): 129-131.

[7] 李坡, 王丹. 基于 Snap7 的西门子 PLC 以太网客户端开发[J]. 江苏高职教育, 2019, 19(1): 56-59.

[8] 姜艳艳. 一种雷达终端软件的显示技术[J]. 电子世界, 2021(12): 41-42.

[9] 喻杰, 高俊. 基于 S7 协议和 Snap7 的直升机传动试验器通信设计[J]. 工业仪表与自动化装置, 2020(5): 66-70.

[10] 杜影, 郑义, 石家勇, 等. 基于 QT 的跨平台数据管理接口设计[J]. 计算机测量与控制, 2025, 33(3): 213-218.

[11] 刘海萍. 光电经纬仪数据交互终端国产化设计与实现[J]. 电子测试, 2019(15): 90-91+96.