bachmann.
SolutionCenter

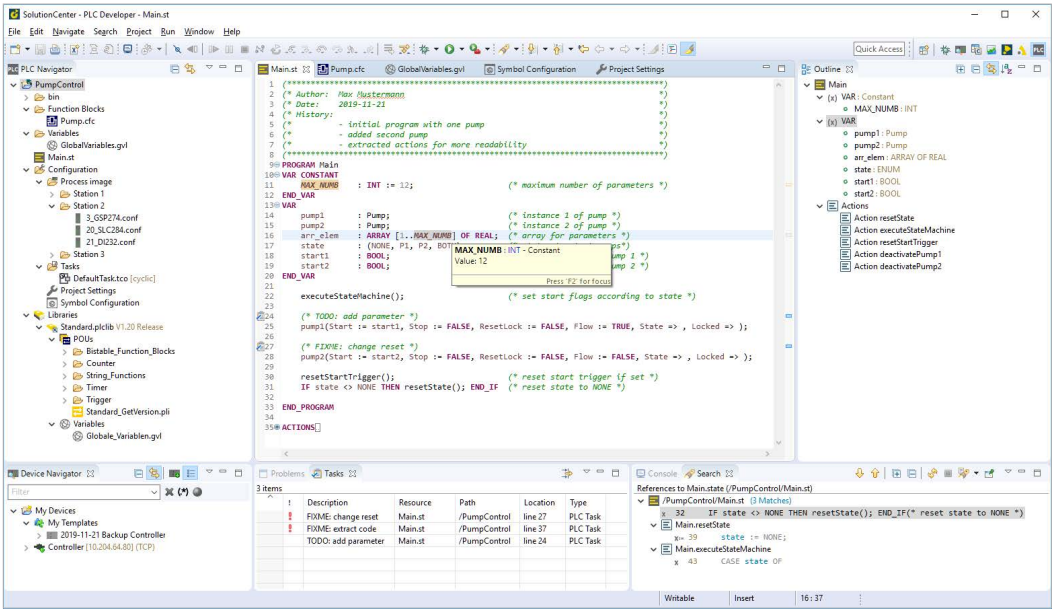
PLC 编程器 (PLC Developer)

如今，自动化解决的复杂问题，远远超出了单一的设备控制范围。IEC 61131-3 编程语言结构化文本 (ST) 中的编程程序非常适用于这种复杂的任务设置。使用这种语言可以实现高效的编程程序。通过 PLC 开发器可以得到一个在软件开发的所有阶段，为设备程序员提供全面支持的工具。

SolutionCenter 协同效应

PLC 编程器与 SolutionCenter 的无缝集成可从由此产生的协同效应中提供附加值：

- 通过将解决方案项目中的 PLC 项目与设备配置和模板相关联，可以将其组合成一个整体的解决方案，并作为一个整体进行管理。
- 通过 GIT 和 SVN 的版本管理功能，用户可以方便地跟踪代码修改，并为有效的发布管理提供依据。
- 以插件形式实现的扩展（还可直接从 Eclipse Marketplace 安装）简化了工程和编程。
- SolutionCenter 脚本 API 支持使用最新软件开发方法，包括持续集成、模型驱动开发和测试驱动开发等。



高效的编程和管理

清晰直观的编辑器可实现先进高效的操作：

- 用户定义的语法高亮显示增加了结构化文本（ST）中的代码可读性。
- 连续功能图（CFC）编程所需的用户界面的先进清晰设计还允许在大型程序中提供良好的概述。
- 得益于完全可用的代码导航和调用层次结构，复杂应用中的相互关系也可以轻松识别。
- 内容相关的自动完成功能和预定义和可扩展的代码和文件模板能够显著提升编程效率。
- 一个工作空间中的项目管理使源文件的所有更改都能够存储在本地历史记录中。无论项目是否进行版本管理，这些变更都可以后续再次调用。
- 程序进行编辑时，广泛的验证功能会即时告知程序员编程错误。快速修复通过自动添加变量声明等程序部分来纠正缺失的部分。
- 不需要离开编辑器，就可以通过工具提示调用和显示接口信息和函数块或变量的文档。
- 项目范围内可以轻松彻底地更改标识符。
- 输入和保存程序时，关键字和标识符的表示法会自动修正。
- CFC 编辑器还可以完全通过键盘操作。这可以提高 CFC 图的创建效率。

通过模块化结构实现先进操作

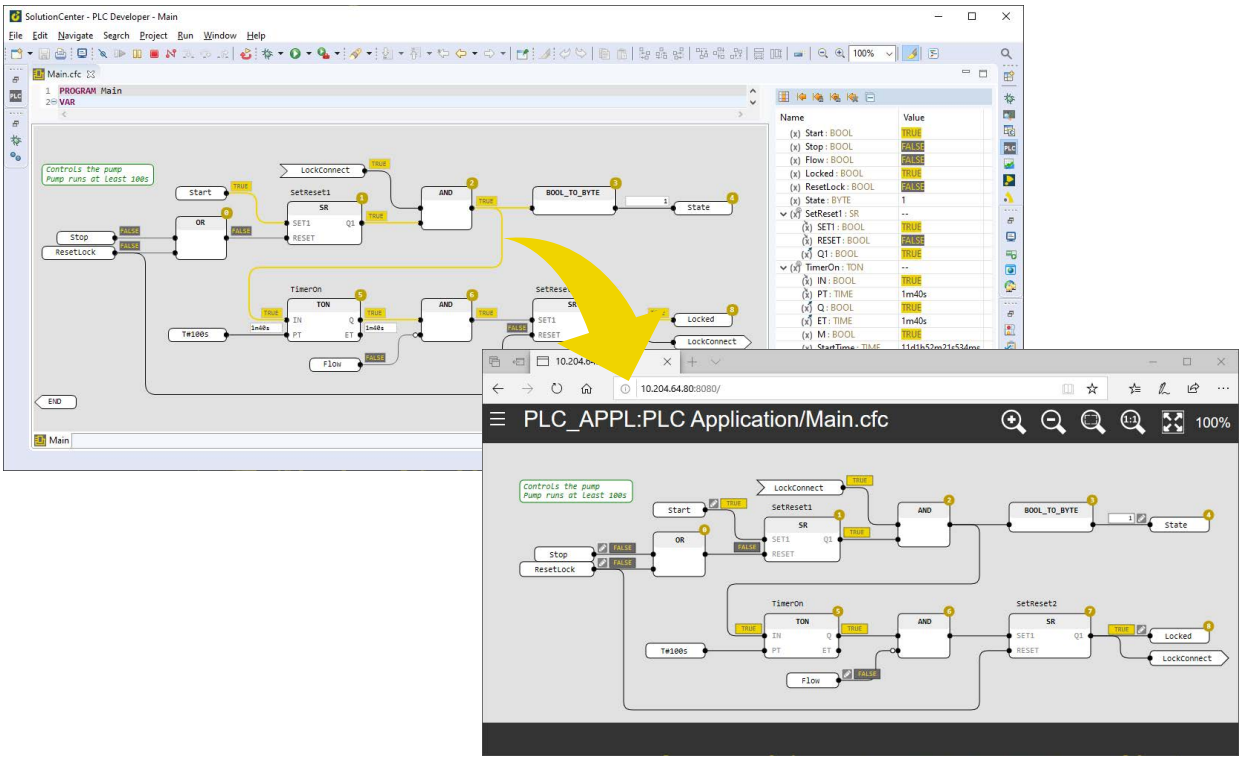
PLC 编程器采用开放、灵活和透明的设计，从而将工程工作推向了一个新的水平：

- 源代码的所有部分都通过定义的文件格式以文本形式保存，这为用户生成模块和配置提供了各种可能性。因此，可以自动执行重复操作步骤，并防止出现复制/粘贴错误。
- 可以根据需要选择程序部分的结构，从而使应用具有模块化设计。
- 为了确保构建的可再现性，将使用的库保存在项目中。可通过更新清晰显示新版本中的库。
- ST、CFC、C 和 C++ 用户定义库的创建和调试可以贯穿始终，特别是支持其他项目中代码的可重用性。
- 编程的执行与目标平台无关，只需要在执行时选择目标平台即可。编译之后，创建的应用可以直接传输到 M1 控制器并运行。
- 外部可见变量可以在符号配置中清晰配置。

通过快速故障排查提升品质

PLC 编程器的调试框架非常实用，特别是在开发新应用和进行故障排查时：

- 启动调试会话后，可以在不停止应用的情况下监控实际变量值。这可以通过 ST 中明确设计的监控表或 CFC 中的在线监控来实现。应用程序的各个变量和结构还可以组合在变量列表中，进行联合监控，并绘制成图表和集。
- 启用流量控制可以在不停止应用的情况下跟踪程序流。
- 应用可以通过断点在任何位置停止，然后分步执行。可以联合管理、激活或完全停用断点。
- 应用停止时将显示堆栈框架，通过该框架还可以导航到要调用的功能。
- 变量发生变化时，可以通过突出显示变量值来更好地评估变量的动态。
- 布尔变量的值用颜色突出显示，以更好地识别应用的当前状态。
- 代码修改可以在线执行，而不需要停止应用运行。PLC 编程器在传输过程中检测是否可以执行在线更改。
- 还可以同时调试多个应用程序。当带有多个任务的应用程序被调试时，可以在调试会话期间连接并更改活动调试任务。
- PLC Insight 是基于所选 CFC 块自动生成的 webMI 可视化。它为用户提供对应用的访问权限，无需使用任何工程工具。可视化与应用程序一起安装在目标控制器上。除信号流之外，还可监控和更改输出变量的值。



PLC 编程器	
总述	
集成	• 在一个工作空间内管理多个 PLC 项目 • 在一个工作空间内管理所有设备和 PLC 项目 • PLC 项目在解决方案项目中关联，用于联合管理
显示	• 为处理 PLC 项目预定义的透视图 • 视图和编辑器的可自由选择且清晰的布置
版本管理	• GIT（本地和远程） • SVN（本地和远程）
库	• 通过版本比较将库集成为副本 • 版本显示 • 加入标准库 • 加入 PLC 库 • 加入外部 PLC 库（C/C++，MATLAB® / Simulink®）
自动化	• SolutionCenter 中脚本的执行（JavaScript 和 Python） • 用于创建和传输应用的预定义模块
可扩展性	• 可通过 Eclipse Marketplace 安装的扩展（插件） • 使用用户创建的 Eclipse 插件
可扩展性	• 独立于目标平台创建应用 • 所有 M200-CPU 上的传输和执行
结构	• 工作集中的项目组合 • 可在项目中自由选择的文件夹结构 • 可全面访问所有源代码文件
兼容性	• 兼容 M-PLC（编译器和运行时间） • 可导入 M-PLC 项目
编辑器	
语言	• 结构化文本（ST） • 连续功能图（CFC） • 可根据需要进行组合
代码导航	• （在项目中）查找引用 • 调用层次结构（直到任务） • 调用者层次结构（可扩展的层次结构） • 开放声明（局部和全局）
自动完成	• 内容相关（声明、调用和程序部分） • 预定义的源代码模板 • 用户定义的源代码模板
自动校正	• 识别符校正 • 关键字符号校正
语法高亮度显示	• 可自由配置 • 分别用于结构化文本、全局变量列表、数据类型和硬件配置
快速修复	• 变量和功能块实例声明 • 创建缺失的元素（连接标记、跳转和跳转标签） • 功能块接口的更新
版本控制	• 与本地存储版本的比较（本地历史记录） • 与 SVN 或 GIT 的源代码比较
变更	• 标识符的全项目重命名
信息	• 标识符工具提示中的内容相关信息（ST） • 功用块和针脚上的内容相关信息（CFC）
键盘操作	• 使用键盘快捷键执行最常用的操作 • 还可通过键盘在 CFC 中编程
验证	• 对输入的源代码进行语法和语义检查 • 在源代码和错误视图中直接显示问题

PLC 编程器	
配置	
符号配置	• 局部和全局变量的可见性 • 对局部和全局变量的访问 • 集合、结构和数组条目的导出 • 可继承的配置
全局变量列表	• 全程变量声明 • 全局常量声明 • 全局保留变量声明
变量配置	• 将变量分配给物理硬件地址
任务配置	• 最多可配置 16 个任务 • 被调用程序的配置 • 任务类型的配置（循环、同步、事件/错误中断和自由轮转） • PTP 同步 • 看门狗超时
过程映像	• 在线和离线设备的硬件导入（选择性） • 硬件模块的手动添加
项目设置	• 版本号 • 模块名称（m 文件） • 容错度 • 多核适用性 • 保留变量的冲突处理 • 存储器布置 • 标记设置
在线操作	
安装	• 代码更改的在线更改 • 目标控制器永久或动态可选 • 临时或永久安装在目标控制器上 • 模块实例的名称可自由选择
初始化	• 启动时可选择的调试任务（带多任务处理项目）
部署配置	• 每个项目可以有多个配置 • 可以组合执行多项配置（启动组） • 收藏夹管理
调试 - 故障排查	
按步处理	• 通过断点停止应用 • 步入/超过/返回
监控	• 编辑器中监控表内的变量值（ST/CFC） • 内联变量值（仅限 CFC） • 所选变量的变量列表 • 通过流量控制显示已执行代码
显示	• 更改INT值的格式（dec、hex 或 bin） • 应用停止时的当前堆栈帧
库	• 通过整合源代码（链接资源）实现内部库 • 通过连接源编码（C/C++）实现外部库
操作	• 可以设置变量值（一次性写入） • 可强制使用可变值（每次执行前写入）
可视化	• 从 CFC 块生成完整的 webMI 可视化（PLC Insight） • 将可视化与应用一起传输到目标 • 通过可视化监控和更改导出的变量 • 根据功能块之间的连接跟踪信号流