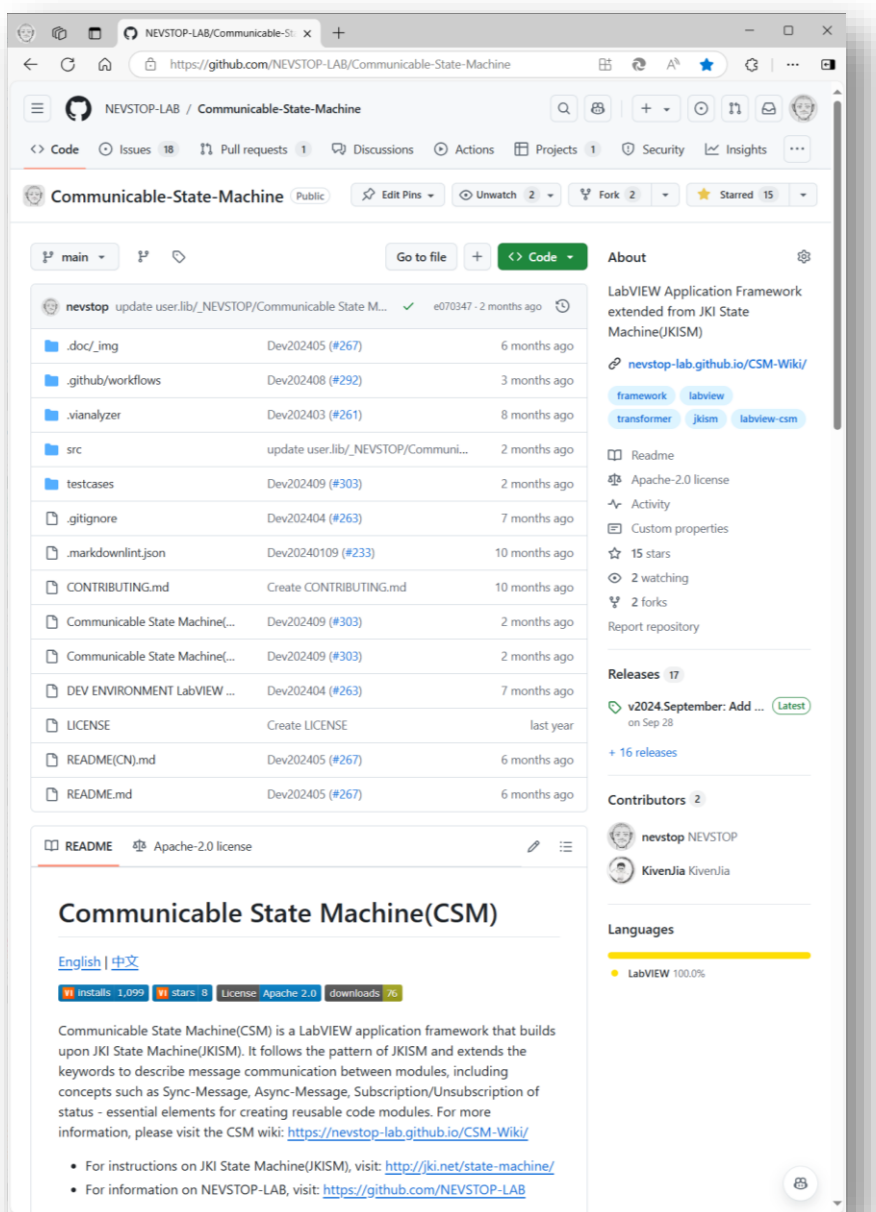


CSM

Communicable State Machine

可通讯状态机



可通讯状态机框架 (Communicable State Machine)

Communicable State Machine(CSM) 是一个基于 JKI State Machine (JKISM) 扩展的开源 LabVIEW 应用开发框架。它遵循 JKISM 的字符串状态描述模式, 扩展了关键词, 用于描述模块之间的消息通信, 包括同步消息、异步消息、状态订阅/取消订阅等概念.

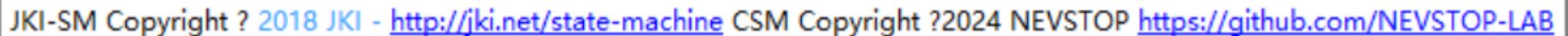
1. 完善且独特的 LabVIEW 程序框架 (vs DQMH/SMO/AF ...)

- 纯文本的流程控制
- 隐形的“通讯总线”
- VI 即模块
- 内置超级详细的检查接口
- 拓展式的设计

2. 全开源项目 ([GitHub](#) | [gitee](#) | [vipm.io](#)), MIT License

3. 中文技术支持

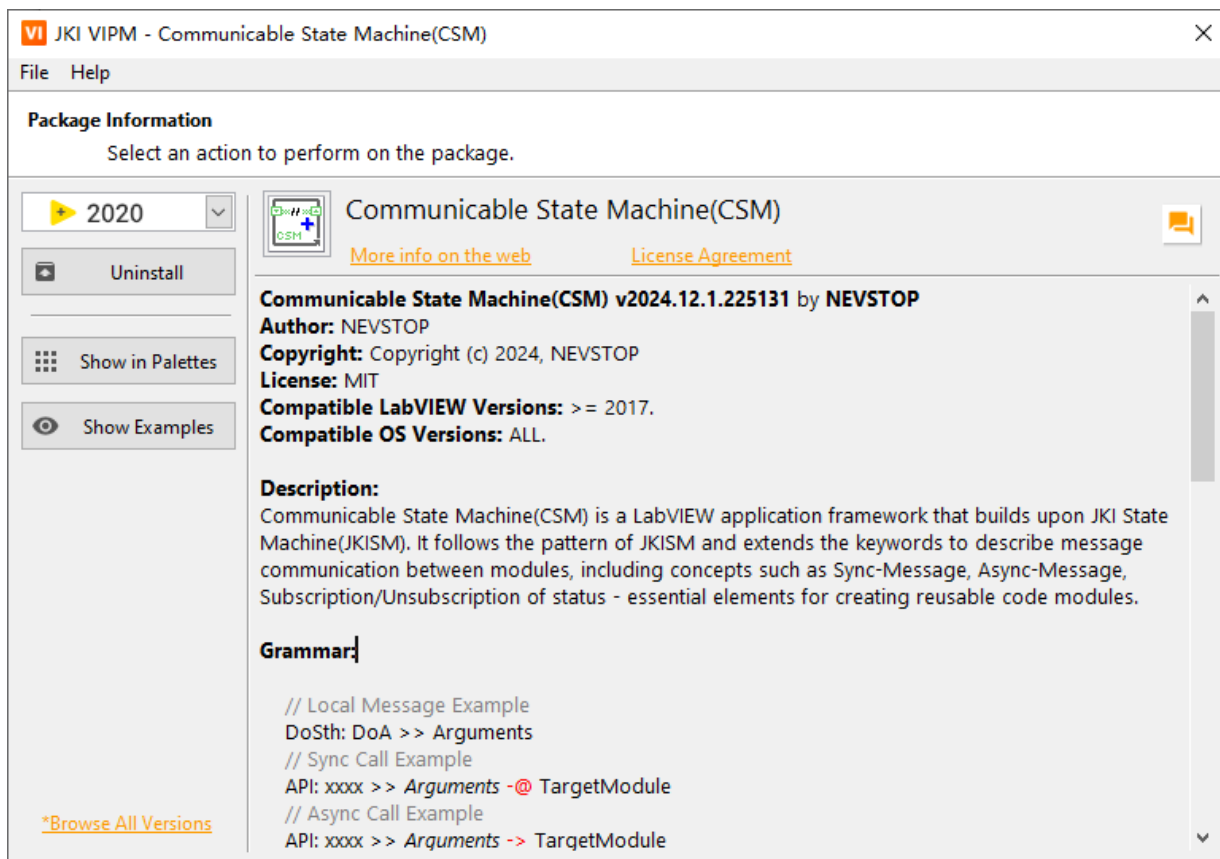
4. 持续的更新迭代



常见的 LabVIEW 编程框架比较

	DQMH	SMO	Actor Framework	CSM
优点	<ul style="list-style-type: none">• 继承于最广泛应用的框架 QMH• 无 OOP，对于大多数 LabVIEW 用户比较友好• 社区活跃度高，资料/工具众多	<ul style="list-style-type: none">• 非常符合面向对象的概念• 模块独立性好，操作调用直观• SMO 最适用的场景还是模块复用，不与调用方一起开发	<ul style="list-style-type: none">• 操作者、消息均解耦，解耦程度高，便于多人协作开发• 熟悉 OOP 的开发人员使用方便	<ul style="list-style-type: none">• 继承于 JKISM，代码集中度高• 无 OOP，对于大多数 LabVIEW 用户比较友好• 通过文档定义接口分工，无需代码• 容易实现操作序列化 (Serialization)• 内置多种高级模式
缺点	<ul style="list-style-type: none">• 代码接口冗余• 两套模板实现 Singleton/Stand-alone 模块	<ul style="list-style-type: none">• 框架复杂度高，对用户要求高• 目前社区用户少，更新频率不高• 资料比较少	<ul style="list-style-type: none">• 框架复杂度高，对用户要求高• 项目需要架构师角色	<ul style="list-style-type: none">• 所有消息必须使用 STRING 格式传递，有转换的开销• 参数不通过 LabVIEW 数据类型定义，不严格要求• 目前工具/资料还不完善

2024 CSM 更新历史



1. [2024.Jan Release: Global Event, Utility and bug fix](#)
2. [2024.Feb Release: Toolbar Entry for CSM Tools](#)
3. [2024.Mar. Release: Debug Tools and Add-ons Update](#)
4. [2024.Apr Release: 添加右键弹出插件](#)
5. [2024.May Release: 增强状态订阅](#)
6. [2024.June Release: 增强INI变量功能](#)
7. [2024.July Release: API String增强 Enum支持](#)
8. [2024.August Release:添加 Log Filter 功能](#)
9. [2024.September Release: CSM Helper API 及小问题修复](#)
10. [2024. October Release: 更新中文 VI Description](#)
11. [2024.November Release: 周期性 LOG 过滤功能](#)
12. [2024.December Release: Update Soon...](#)

IV. CSM 模块接口调试工具

CSM Debug Console (CSM)

History Length 50

Execution:State Bad

Defer Panel Updates

Message

Arguments

Response

[14:05:23.190] [Module Destroyed] Acquisition

[14:05:23.190] [State Change] Acquisition | <Remaining States> >> <Empty>

[14:05:23.190] [State Change] Acquisition | Exit

[14:05:23.190] [State Change] Acquisition | Data: Cleanup

[14:05:23.190] [State Change] Acquisition | Macro: Exit(Complete) <- Tool-Testpnl

[14:05:23.190] [State Change] Acquisition | Macro: Exit <- Tool-Testpnl

[14:04:34.802] [State Change] Acquisition | VI Reference(Complete) <- CSM-DebugConsole

[14:04:34.802] [State Change] Acquisition | VI Reference <- CSM-DebugConsole

[14:04:20.091] [State Change] Acquisition | Data: Update Indicators

[14:04:20.091] [State Change] Acquisition | Initialize Core Data

[14:04:20.091] [State Change] Acquisition | Data: Initialize

[14:04:20.090] [State Change] Acquisition | Macro: Initialize

[14:04:20.090] [Module Created] Acquisition

Acquisition.lvlib:Acquisition.vi Front Panel on CSM Continuous Measure

Name(" to use uid)

Acquisition

Error Occurred

0

State History

[14:05:23.161] Exit

[14:05:23.160] Data: Cleanup

[14:05:23.160] Macro: Exit <- Tool-Testpnl

[14:04:34.771] VI Reference <- CSM-DebugConsole

[14:04:20.090] Data: Update Indicators

[14:04:20.090] Initialize Core Data

[14:04:20.090] Data: Initialize

[14:04:20.090] Macro: Initialize

程序框图

DebugTool Dialog (CSM)

Add VI Reference Case to CSMs..

CSM - PTR Mointor

Remove all CSM Bookmarks and Comments..

Switch VI Decription Language..

CSM-INI Variable Viewer - DebugTool

状态列表

"Idle"

"CSM Documentation"

"----- Core -----"

Default

"VI Reference"

"Error Handler"

"Critical Error"

"Target Busy Error"

"Target Timeout Error"

"Target Error"

"Initialize Core Data"

"Async Response", "Response"

"Async Message Posted"

"Exit"

"----- Data -----"

"Data: Initialize"

"Data: Update Indicators"

"Data: Set Configuration"

"Data: Get Configuration", "Data: Get Internal Data"

"Data: Cleanup"

"----- Macro -----"

"Macro: Initialize"

"Macro: Exit"

"----- API -----" API 接口

"API: Start"

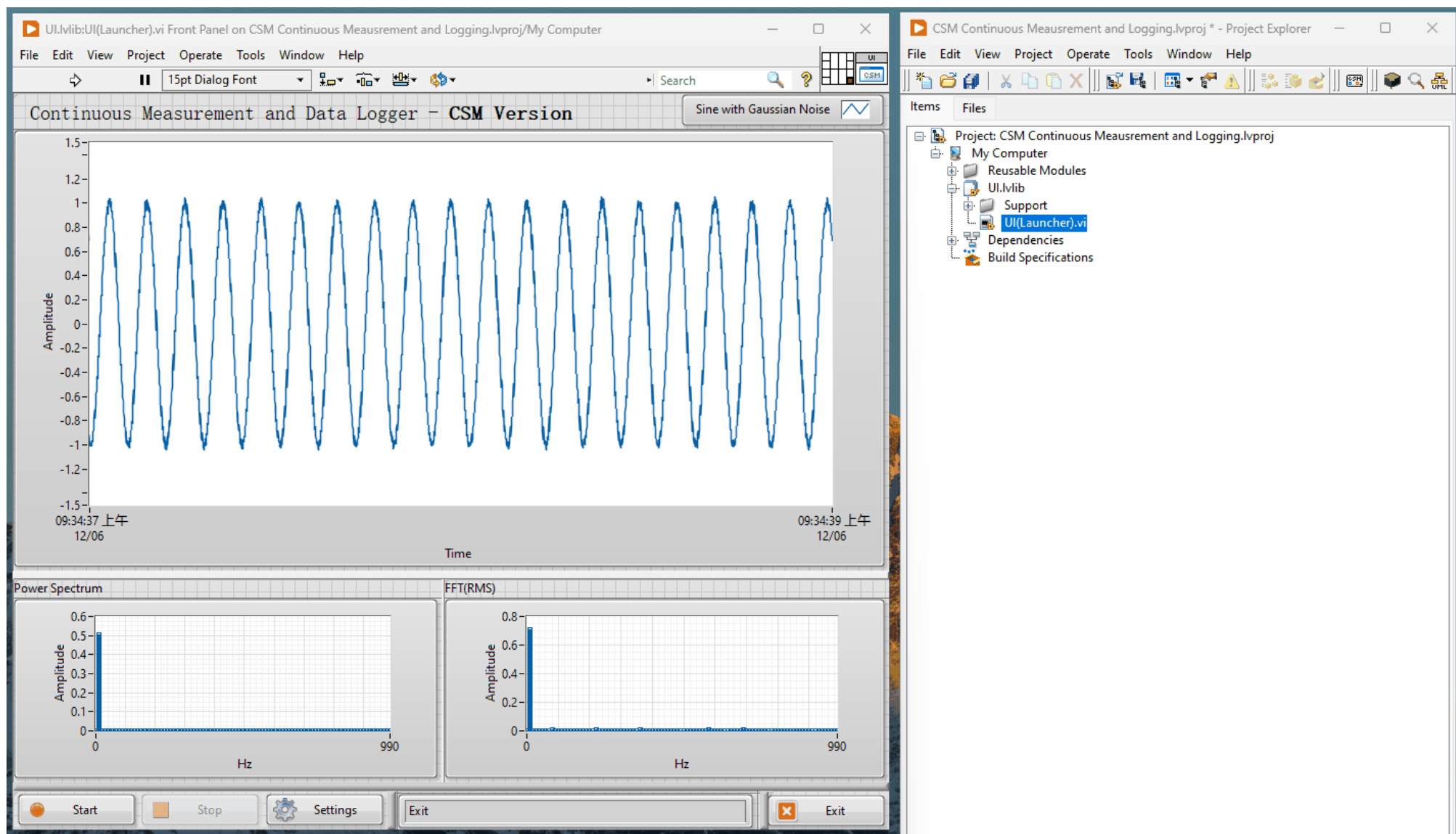
✓ "API: Stop"

"API: Acquire"

"API: Update Settings"

"API: Update Settings v2.0"

IV. CSM 模块接口调试工具



欢迎下载试用，并提供反馈

通过 GitHub/Gitee Issue/PR/Discussions 参与开发，提供反馈。

1



GitHub

[NEVSTOP-LAB/Communicable-State-Machine](https://github.com/NEVSTOP-LAB/Communicable-State-Machine)

2



gitee

[Communicable-State-Machine: 可通信状态机 \(CSM\)](https://gitee.com/Communicable-State-Machine:可通信状态机(CSM))

3



VIPM

[Communicable State Machine Framework - Package List](https://github.com/Communicable-State-Machine/Package-List)

4



知乎

[知乎专栏：可通讯状态机\(CSM\) 框架](https://zhuanlan.zhihu.com/Communicable-State-Machine)