

PXView MCP 服务与 Web 客户端使用指南

PXView v1.6 · PXLogic 官方文档 · fpgausb.com

概述

PXView 内置了 MCP (Model Context Protocol) 服务器，允许 AI 编程工具通过标准化协议控制硬件设备进行信号采集和协议解码。同时提供了基于浏览器的 Web 客户端，支持通过自然语言与设备交互。

- **MCP 服务器地址:** `http://127.0.0.1:10110`
- **Web 客户端地址:** `http://127.0.0.1:10110/` (需先构建前端)
- **协议版本:** MCP 2025-03-26
- **传输方式:** Streamable HTTP

1. 启用 MCP 服务

PXView 启动后 MCP 服务自动运行，无需额外配置。在右侧边栏点击 **MCP** 按钮可查看服务状态和控制面板。

控制面板功能： - **打开 MCP 网页端聊天界面** — 在浏览器中打开 Web 客户端 - **连接 AI 工具** — 显示各工具的连接命令 - **重启 MCP 服务** — 重启 MCP 传输层

2. 连接 AI 工具

Claude Code

```
claude mcp add --transport http pxview http://127.0.0.1:10110
```

Codex

```
codex mcp add --url http://127.0.0.1:10110 pxview
```

OpenCode

```
opencode --mcp http://127.0.0.1:10110
```

其他 MCP 客户端

将 MCP 客户端指向 `http://127.0.0.1:10110`，使用 Streamable HTTP 传输方式。

3. 可用工具列表

MCP 服务器提供以下 15 个工具：

设备管理

工具	说明
<code>get_devices</code>	列出已连接的设备，返回设备 ID、名称、模式等信息
<code>get_channels</code>	获取当前设备的通道列表和索引

采集控制

工具	说明
start_capture	启动信号采集，配置设备、通道、采样率、触发等
stop_capture	停止正在进行的采集
wait_capture	等待采集完成（阻塞调用），可设置超时时间
get_capture_status	查询当前采集状态（idle/capturing/completed）

协议解码

工具	说明
list_analyzers	列出所有可用的协议解码器（如 SPI、I2C、UART 等）
get_analyzer_options	查询指定解码器的通道和选项配置要求
add_analyzer	添加协议解码器，建议在 start_capture 之前调用
remove_analyzer	移除已添加的解码器
get_analyzer_results	获取解码结果（标注数据）

数据导出

工具	说明
export_raw_data_csv	导出原始采集数据为 CSV 文件
export_raw_data_binary	导出原始采集数据为二进制文件
export_data_table_csv	导出解码结果为 CSV 数据表

文件操作

工具	说明
load_capture	从 .pxc 会话文件加载历史采集数据
save_capture	保存当前采集到 .pxc 会话文件
close_capture	关闭当前采集并释放资源

4. 典型工作流

4.1 采集并解码 PWM 信号

```
1. get_devices           → ID
2. get_channels          →
3. add_analyzer(analyzerName="pwm_c", settings={channel: 14})
4. start_capture(deviceId=..., logicDeviceConfiguration={digitalChannels: [14], ...})
5. wait_capture(timeoutSeconds=30)
6. get_analyzer_results(analyzerId=...) →
```

4.2 叠加解码（如 I2C → EEPROM）

```
1. add_analyzer(analyzerName="i2c_c", settings={sda: 0, scl: 1})
   → analyzerId: "12345"
```

```

2. add_analyzer(analyzerName="eeprom24xx_c", stackOnAnalyzerId="12345")
3. start_capture(...)
4. wait_capture(...)
5. get_analyzer_results(analyzerId="12345")      → I2C
6. get_analyzer_results(analyzerId="67890")      → EEPROM

```

4.3 加载历史数据分析

```

1. load_capture(filepath="path/to/session.pxc")
2. add_analyzer(analyzerName="spi_c", settings={...})
3. get_analyzer_results(analyzerId=...)

```

5. start_capture 参数详解

logicDeviceConfiguration

参数	类型	说明
digitalChannels	integer[]	启用的数字通道索引列表
analogChannels	integer[]	启用的模拟通道索引列表
digitalSampleRate	integer	数字采样率 (Hz)
analogSampleRate	integer	模拟采样率 (Hz)
digitalThresholdVolts	number	数字阈值电压
thresholdPreset	string	阈值预设 (1.8V / 3.3V / 5V / Adjustable)
channelMode	string	通道模式 (Buffer / Stream)
diskCacheEnabled	boolean	启用磁盘缓存 (长时间采集)
glitchFilters	array	毛刺滤波配置

captureConfiguration

参数	类型	说明
timedCaptureMode.durationSeconds	number	定时采集时长 (秒)
manualCaptureMode.sampleCount	integer	手动采集样本数
digitalCaptureMode.triggerChannelIndex	integer	触发通道索引
digitalCaptureMode.triggerType	string	触发类型: rising / falling / pulse_high / pulse_low
digitalCaptureMode.afterTriggerSeconds	number	触发后缓冲时长
captureRatio	integer	触发位置百分比 (0-100)

6. Web 客户端

构建与安装

```
# build
ninja webui #
ninja install-webui # +

#
ninja && ninja install # install webui
```

Web 客户端需要 Node.js 和 npm。构建产物位于 `web/dist/`，安装到 `bin/webui/`。

使用

确保已构建并安装 Web 客户端

在 PXView 侧边栏点击 **MCP** → **打开 MCP 网页端聊天界面**

或直接在浏览器访问 `http://127.0.0.1:10110/`

配置 LLM

Web 客户端需要配置 LLM API 才能使用自然语言功能。在设置页面填入： - **API Base URL**: 如 `https://api.openai.com/v1` - **API Key**: 你的 LLM API 密钥 - **Model**: 如 `gpt-4o`

开发模式

```
cd web
npm install
npm run dev # Vite http://localhost:3000
```

开发模式下前端直连 PXView MCP 服务器，支持热重载。

7. 直接调用 MCP API

MCP 使用 JSON-RPC 2.0 over HTTP 协议。可直接用 curl 测试：

初始化

```
curl -X POST http://127.0.0.1:10110/mcp \
-H "Content-Type: application/json" \
-d '{"jsonrpc":"2.0","id":1,"method":"initialize","params":{"protocolVersion":"2025-03-26","capabilities":{},"clientInfo":{"name":"test","version":"1.0"}}}'
```

列出工具

```
curl -X POST http://127.0.0.1:10110/mcp \
-H "Content-Type: application/json" \
-d '{"jsonrpc":"2.0","id":2,"method":"tools/list","params":{}}'
```

调用工具

```
curl -X POST http://127.0.0.1:10110/mcp \
-H "Content-Type: application/json" \
-d '{"jsonrpc":"2.0","id":3,"method":"tools/call","params":{"name":"get_devices","arguments":{}}}'
```

8. 故障排除

问题	解决方案
MCP 工具连接失败	确认 PXView 正在运行，检查端口 10110 未被占用

问题	解决方案
Web 客户端 404	需要先构建前端： <code>ninja webui && ninja install</code>
解码无数据	确保 <code>add_analyzer</code> 在 <code>start_capture</code> 之前调用
<code>wait_capture</code> 超时	增大 <code>timeoutSeconds</code> 参数，或检查采集是否正常启动
<code>get_analyzer_results</code> 为空	采集完成后等待几秒再查询，解码可能需要时间