



Pclika 开发者指南

从购板到 AI 开发 — 完整配置流程

DOCUMENT · Pclika ESP32-S3 + pclika-bridge + MCP

VERSION · 0.1.0 · 2026年6月

CLIENTS · Claude Code · Codex · Cursor · VS Code · OpenCode

STACK · Python 3.10+ · ESP-IDF v5.x

CONTACT · pclika.com · starinvc@gmail.com

SEAL · PCK-MMXXVI-C4A32096

本指南手把手带你完成从购买硬件到下达第一条 AI 硬件指令的全流程，无需硬件经验。

00 概述

Pclika 将 ESP32-S3 开发板变成 AI 控制的硬件平台。AI 编程助手通过标准 MCP 工具调用（JSON-RPC 2.0）经 USB 实时控制硬件，无需额外编写固件。

步骤	标题	说明
01	购买套件	在官方商城或 MCP 自助平台下单
02	接线	USB-C 连接电脑，I2C 传感器接 JST SH 接口
03	刷入固件	idf.py flash 一条命令，约 90 秒
04	安装 Bridge	pip install pclika-bridge 即可
05	验证连接	pclika-bridge --list-ports 测试 device_info
06	配置 AI 客户端	Claude Code / Codex / Cursor / VS Code / OpenCode
07	加载指令集	将 System Prompt 粘贴进 AI 客户端一次即可
08	描述你的需求	用自然语言描述项目，AI 开始编码
09	迭代与部署	AI 写固件 + MCP 调用，你审核并烧录
10	规模扩展	增加模块、切换 SSE 多客户端模式

01 购买套件

最简单的入门方式是 Pclika 新手套件，包含底板和传感器包，预先通过全部示例验证。

官方购买渠道

用户商城（官方）： https	https://mall.pclika.com/
MCP 自助购买： https	https://hc.pclika.com/

新手套件 ¥ 299（推荐）

物品	数量	备注
Pclika ESP32-S3 底板 v0.1	1 块	主控 · USB-C · Wi-Fi · BLE
DHT22 温湿度传感器	1 个	I2C · env-monitor 示例
BH1750 光照传感器	1 个	I2C · 勒克斯测量
USB-C 数据线（1 米）	1 条	数据线，非充电线
JST SH 4P 线缆 × 3	3 条	I2C 传感器连接

DIY / 自制板

如果你想自己打板，GitHub 仓库提供完整 KiCad 设计文件和 JLCPCB 订购指南。

可选模块

模块	用途
SG90 舵机 × 4	舵机 PWM 接口 — servo-control 示例
SSD1306 OLED 128 × 64	I2C 显示 — display_text 工具
OV2640 摄像头	摄像头 — vision-snapshot 示例
RS-485 模块	Modbus RTU — industrial-gateway 示例
iCE40UP5K M.2 FPGA	FPGA 扩展 — HDL Bridge

02 连接硬件

物理连接

外设	连接	说明
USB-C 数据线	板 J1 → 电脑 USB	供电 + 串口通信
DHT22 / BME280	SDA=GPIO21 SCL=GPIO22 JST SH 4P	env-monitor 示例
SSD1306 OLED	同一 I2C 总线，地址 0x3C	display_text 工具
舵机	板边 PWM0 – PWM3 接口	servo_move 工具

板载 LED 状态

LED状态	含义
蓝灯常亮	上电正常，固件运行中
青灯慢闪	Bridge 待机，等待 MCP 指令
青灯快闪	MCP 会话活跃，正在接收工具调用
红灯亮	发生错误，用 serial_log_read 检查日志

>> 注意：使用 USB 数据线，纯充电线无数据引脚，不会出现在串口列表。如果 --list-ports 找不到设备，请先换线。

03 刷入固件

前置条件

- Python 3.10+ (`python3 --version`)
- ESP-IDF v5.x (<https://docs.espressif.com/projects/esp-idf>)
- `idf.py` 在 PATH 中 (`idf.py --version`)
- Pclika 板通过 USB-C 数据线连接

克隆并刷写

```
git clone https://github.com/Pclika/mcp-platform
cd mcp-platform/firmware
idf.py set-target esp32s3
idf.py flash monitor
```

预编译固件（无需 ESP-IDF）

如果不想安装 ESP-IDF，可在 GitHub Releases 下载预编译 .bin 文件，用 esptool 烧录：

```
pip install esptool
esptool.py --port /dev/ttyUSB0 --baud 460800 \
  write_flash -z 0x0 pclika-firmware-v0.1.bin
```

>> Windows：如果 `idf.py flash` 提示端口占用，请先关闭 Arduino IDE、PuTTY 等串口工具。

04 安装 Bridge

```
pip install pcliika-bridge

# verify
pcliika-bridge --version
pcliika-setup --list
```

安装内容

命令/包	用途
pcliika-bridge	CLI 命令 — 启动 MCP 服务（STDIO 或 SSE 模式）
pcliika-setup	CLI 命令 — 自动为所有 AI 客户端写入配置
pcliika_bridge	Python 库 — 可在自定义脚本中导入

连接模式

模式	说明	最适合
STDIO（默认）	每个 AI 客户端独立启动一个 Bridge 进程	简单，开箱即用
SSE / 自适应	单个 Bridge 进程，所有客户端共享连接	多客户端并发推荐

05 验证连接

查看可用串口

```
pcliika-bridge --list-ports
# /dev/ttyUSB0   Pcliika ESP32-S3   (USB VID:PID=303A:1001)
```

快速功能测试

以调试模式运行 Bridge，手动调用 device_info：

```
pcliika-bridge --debug
> {"jsonrpc":"2.0","id":1,"method":"tools/call",
  "params":{"name":"device_info","arguments":{}}}
< {"mcu":"ESP32-S3","firmware":"0.1.0","ip":"192.168.1.x","mac":"AA:BB:CC:..."}
>> 如报 DeviceNotConnected：检查是否是数据线、板子 LED 是否亮、端口名是否与 --list-ports 一致。
```

06 配置 AI 客户端

运行 pcliكا-setup 一键配置所有已安装客户端：

```
pcliكا-setup
# Detected: Claude Code, Cursor, VS Code
# Writing ~/.cursor/mcp.json      ... OK
# Writing .vscode/mcp.json      ... OK
# Running: claude mcp add pcliكا...  ... OK

# SSE mode (multi-client):
pcliكا-bridge --serve &
pcliكا-setup --sse
```

客户端	配置文件/命令	位置
Claude Code	claude mcp add pcliكاPlatform pcliكا-bridge	run once in terminal
Cursor	cursor.mcp.json	~/.cursor/mcp.json
VS Code	vscode.mcp.json	.vscode/mcp.json
Codex	codex.config.toml	~/config/codex/config.toml
OpenCode	opencode.jsonc	~/config/opencode/config.json

07 加载工具说明

将以下 System Prompt 粘贴到 AI 客户端的会话开头（或设置为持久系统提示）。这告诉 AI 有哪些工具、每个工具的作用，以及如何安全使用。

系统提示 — 直接复制粘贴

```
You are an AI assistant with direct hardware control via MCP tools.

Connected: Pclika ESP32-S3 board on local USB serial port.

ALWAYS call device_info() first to confirm connection.

Available MCP tools (server: pclikaPlatform):
  sensor_read      - read I2C/UART sensor (temp, humidity, light, distance, IMU)
  display_text     - write text to SSD1306 OLED or ST7789 TFT
  servo_move       - move servo channel (0-3) to angle (0-180 degrees)
  gpio_write       - set GPIO pin HIGH (1) or LOW (0)
  gpio_read        - read GPIO pin state
  led_control      - set R/G/B (0-255) on 4x WS2812B LEDs
  wifi_scan        - list nearby Wi-Fi networks (SSID, dBm, channel)
  button_read      - read BOOT/RESET/USER button state
  device_info      - MCU model, firmware version, IP, MAC
  serial_log_read  - UART diagnostic log (default 20 lines, max 200)

Safety: GPIO is 3.3V. Servos need 5V via PWM header.
On DeviceNotConnected error -- stop and report, do not retry in a loop.

>> Claude Code : 加入 CLAUDE.md 或 --system-prompt。Cursor/VS Code : 设置 → AI → 系统提示。Codex/OpenCode : --system
参数或配置文件 system_prompt 字段。
```

08 描述项目

System Prompt 加载后，直接描述你想做什么。AI 会规划实现方案、调用硬件工具验证传感器读数、生成固件代码，并引导你完成烧录。

示例需求

标题	需求	技能
Hello World	让板载 LED 绿色闪烁 3 次，然后变蓝，最后关闭。	测试整个技术栈，10 秒内完成。
环境监测	每 5 秒读取 BME280 传感器，将温度和湿度显示在 OLED 上。温度超过 28 ° C 时 LED 变红。	传感器 + 显示 + 逻辑控制。
舵机扫描	将 0 号舵机从 0 ° 到 180 ° 每步 10 ° 运动，每个位置停留 500ms，并回读当前角度记录日志。	PWM 输出 + 反馈。
Wi-Fi 扫描	扫描附近 Wi-Fi，输出按信号强度排序的 SSID、dBm、信道表格，标出开放网络。	RF 数据 → AI 分析。
自定义项目	我要做一个植物浇水监测器：每分钟读取 ADC0 通道的土壤湿度，湿度低于 30% 时记录警告并红灯闪 5 次，所有读数带时间戳写入串口日志。	ADC + LED + 日志。

09 完整开发循环

提交需求后，AI 自动执行以下工作流：

#	阶段	内容
1	连接检查	AI 调用 device_info() 确认板子在线，再执行任何操作。
2	传感器验证	涉及传感器的项目，AI 先读取一次确认响应正常，数值合理。
3	规划	AI 用白话描述实现方案，确认后才写代码。
4	固件代码	AI 生成带注释的 ESP-IDF C 代码，每个函数都有说明，GPIO 精确对应你的板子。
5	烧录指引	AI 告诉你运行哪条 idf.py 命令，以及串口监控中应看到什么。
6	实时测试	AI 调用 MCP 工具验证运行中的固件：读传感器、检查 LED 状态、验证舵机角度。
7	迭代	出错时，AI 读取 serial_log_read() 诊断并提出修复方案，通常一轮解决。

完整会话示例（hello-mcp）

```
You: Make the LED pulse cyan 5 times then show 'HELLO' on the OLED.  
  
[AI] device_info()          => ESP32-S3 online, firmware 0.1.0  
[AI] led_control(r=0,g=200,b=160) x5 => cyan pulse  
[AI] led_control(r=0,g=0,b=0)    => off  
[AI] display_text('HELLO', line=0)  => OLED updated  
AI: Done. LED pulsed cyan 5x. 'HELLO' is shown on OLED line 0.
```

10 规模扩展

多客户端 SSE 模式

同时打开 Cursor 和 VS Code 时，运行一个共享 Bridge，两者共用同一个硬件连接：

```
# Terminal 1 - keep running
pclika-bridge --serve
# => Listening on http://localhost:3141/sse

# Terminal 2 - write SSE config to all clients
pclika-setup --sse

curl http://localhost:3141/health
# {"status":"ok","port":"/dev/ttyUSB0","connected_clients":2}
```

增加更多模块

模块	功能	MCP调用
BME280	温度 · 湿度 · 气压	sensor_read("bme280")
MPU6050	加速度计 · 陀螺仪 · IMU	sensor_read("imu")
VL53L0X	飞行时间距离传感器	sensor_read("distance")
SHT31	高精度温湿度	sensor_read("sht31")
PCA9685	16 路 PWM (更多舵机)	servo_move(ch, angle)
OV2640	摄像头 — JPEG 捕获	camera_capture()

FPGA 扩展 (进阶)

将 iCE40UP5K FPGA 模块插入 M.2 插槽，获得确定性时序外设。安装 HDL Bridge 后新增 6 个 MCP 工具：

```
pip install pclika-hdl-bridge
pclika-setup # adds pclikaHDL server automatically
```

附录 A — MCP 工具快速参考

工具	参数	返回/操作
sensor_read	sensor_id: "temp_humidity" "bme280" "light" "distance" "imu" "sht31"	返回传感器读数 JSON
display_text	text: str · line: int (0 – 7 OLED 行)	写入 SSD1306 / ST7789 显示
servo_move	channel: int (0 – 3) · angle: float (0 – 180)	移动舵机到指定角度
gpio_write	pin: int · value: int (0 或 1)	设置 GPIO 高低电平
gpio_read	pin: int	读取 GPIO 电平 (0 或 1)
led_control	r/g/b: int (0 – 255)	设置 4 个 WS2812B LED 颜色
wifi_scan	(无参数)	返回附近 Wi-Fi 列表
button_read	"BOOT" "RESET" "USER"	读取按键状态 (0 松开, 1 按下)
device_info	(无参数)	返回 MCU 型号、固件、IP、MAC
serial_log_read	lines: int (默认 20 , 最大 200)	读取 UART 日志, 用于诊断

附录 B — 故障排查

DeviceNotConnected 错误	提示: 90% 原因：充电线当数据线用
1. 板子蓝灯是否亮？ 2. 换数据线（纯充电线无数据引脚）。 3. 运行 --list-ports 查看端口。 4. Linux：sudo usermod -aG dialout \$USER	
端口出现但 Bridge 卡住	提示: idf.py monitor 查看固件启动日志
1. 固件版本错误，重新烧录。 2. 波特率错误，使用 --baud 115200。 3. 其他程序占用端口，关闭 Arduino IDE 等。	
AI 客户端找不到 pcliPlatform 服务器	提示: Claude Code：/mcp 查看服务器状态
1. 重新运行 pcli-platform-setup。 2. 修改配置后重启 AI 客户端。 3. 确认 pcli-bridge 在 PATH 中：which pcli-bridge	
sensor_read 返回 null	提示: I2C 扫描器 sketch 可查设备地址
1. 检查接线 SDA=GPIO21 SCL=GPIO22。 2. 运行 serial_log_read(50) 让 AI 诊断 I2C 错误。 3. 确认传感器地址（BME280=0x76/0x77）。	
servo_move 舵机不动	提示: SG90 额定 4.8 – 6V，直接用 USB 5V 供电
1. 检查供电，舵机需要 5V，不是 3.3V。 2. PWM 通道：0=GPIO38, 1=39, 2=40, 3=41。 3. 角度必须在 0.0 – 180.0。	
SSE 模式客户端无法连接	提示: pcli-platform-setup --sse 重写所有客户端配置为 SSE 模式
1. 确认 pcli-bridge --serve 已运行。 2. 浏览器访问 http://localhost:3141/health。 3. 防火墙可能屏蔽 3141 端口，添加例外。	

支持与反馈

遇到问题？有功能请求或改进建议？我们的支持门户始终开放。

<https://pclika.com/Support>

渠道	联系方式
网页门户	https://pclika.com/Support
电子邮件	starinvc@gmail.com
GitHub Issues	github.com/Pclika/mcp-platform/issues
用户商城	https://mall.pclika.com/
MCP自助	https://hc.pclika.com/

您的反馈直接影响下一个版本。每一份错误报告、功能请求和评价都至关重要。

github.com/Pclika/mcp-platform

pypi.org/project/pclika-bridge

商城：<https://mall.pclika.com/> · MCP自助：<https://hc.pclika.com/>

Apache 2.0 (软件) · CERN-OHL-S v2 (硬件) · © 2026 Pclika

Pclika