

Pclika

Guide du Developpeur

De la carte a l'IA — Configuration et flux de travail complets

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

VERSION · 0.1.0 · Juin 2026

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

Ce guide vous accompagne pas a pas de l'achat du materiel jusqu'a la premiere commande materielle pilotee par IA. Aucune experience en electronique requise.



00 APERÇU

Pclika transforme une carte ESP32-S3 en plateforme materielle pilotee par IA. L'assistant appelle des fonctions materiel via des outils MCP standard (JSON-RPC 2.0) par USB en temps reel.

ÉTAPE	TITRE	DESCRIPTION
01	Acheter le kit	Commander en boutique ou via MCP
02	Connecter le materiel	USB-C au PC, capteurs I2C sur JST SH
03	Flasher le firmware	Une commande : idf.py flash — ~90 secondes
04	Installer le Bridge	pip install pclika-bridge — termine
05	Verifier la connexion	pclika-bridge --list-ports puis device_info
06	Configurer le client IA	Claude Code / Codex / Cursor / VS Code / OpenCode
07	Charger les instructions	Coller le prompt systeme une seule fois
08	Decrire votre projet	Ecrire une exigence en langage naturel
09	Iterer et deployer	L'IA ecrit le firmware, vous validez
10	Passer a l'echelle	Ajouter des modules, mode SSE multi-clients

01 ACHETER LE KIT

La façon la plus simple de démarrer est le Kit Starter Pclika — carte et capteurs testés sur tous les exemples.

CANAUX D'ACHAT OFFICIELS

Boutique utilisateurs	https://mall.pclika.com/
Achat MCP self-service	https://hc.pclika.com/

Kit Starter ¥299 (Recommandé)

ARTICLE	QTÉ	NOTES
Pclika ESP32-S3 Baseboard v0.1	1 pc	MCU principal · USB-C · Wi-Fi · BLE
Capteur DHT22 temp./humidité	1 pc	I2C · exemple env-monitor
Capteur de luminosité BH1750	1 pc	I2C · mesure en lux
Cable USB-C (1 m)	1 pc	Cable de données — pas de charge uniquement
Cable JST SH 4P × 3	3 pcs	Connexions capteurs I2C

DIY / FABRIQUER SA PROPRE CARTE

Les fichiers KiCad et le guide de commande JLCPCB sont dans le dépôt GitHub.

MODULES OPTIONNELS

MODULE	UTILISATION
Servo SG90 × 4	Sorties PWM — exemple servo-control
OLED SSD1306 128x64	Ecran I2C — outil display_text
Camera OV2640	Camera — exemple vision-snapshot
Adaptateur RS-485	Modbus RTU — exemple industrial-gateway
FPGA iCE40UP5K M.2	Extension FPGA — HDL bridge

02 CONNECTER LE MATÉRIEL

CONNEXIONS PHYSIQUES

PÉRIPHÉRIQUE	CONNEXION	NOTE
Cable USB-C	Carte J1 → USB PC	Alimentation + port serie
DHT22/BME280	SDA=GPIO21 SCL=GPIO22 JST SH 4P	Exemple env-monitor
OLED SSD1306	Meme bus I2C, adresse 0x3C	Outil display_text
Servo	Connecteurs PWM0–PWM3	Outil servo_move

ÉTAT DES LED

ÉTAT LED	SIGNIFICATION
Bleu fixe	Sous tension — firmware en cours d'execution
Pulse cyan lent	Bridge en attente de commandes MCP
Clignotement rapide	Session MCP active — appels en cours
LED rouge allumee	Erreur — verifier serial_log_read

>> Utilisez un cable USB avec donnees — les cables de charge uniquement n'ont pas de lignes de donnees et n'apparaissent pas dans la liste des ports.

03 FLASHER LE FIRMWARE

PRÉREQUIS

- Python 3.10+ (python3 --version)
- ESP-IDF v5.x (docs.espressif.com/projects/esp-idf)
- idf.py dans le PATH (idf.py --version)
- Carte Pclika connectée via câble USB-C avec données

CLONER ET FLASHER

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

FIRMWARE PRÉCOMPILÉ (sans ESP-IDF)

Téléchargez le .bin depuis GitHub Releases et flashez avec esptool :

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

>> Windows : si idf.py flash signale le port occupé, fermez Arduino IDE, PuTTY, etc.

04 INSTALLER LE BRIDGE

```
pip install pcliika-bridge

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

CE QUI EST INSTALLÉ

COMMANDE / PAQUET	OBJECTIF
pcliika-bridge	Commande CLI — demarrer le serveur MCP (STDIO ou SSE)
pcliika-setup	Commande CLI — auto-configuration de tous les clients IA
pcliika_bridge	Bibliotheque Python — importable dans vos scripts

MODES DE CONNEXION

MODE	DESCRIPTION	IDÉAL POUR
STDIO (default)	Chaque client IA lance son propre processus bridge	Simple, fonctionne immediatement
SSE / Adaptatif	Un seul processus bridge pour tous les clients	Ideal pour multi-clients



05 VÉRIFIER LA CONNEXION

LISTER LES PORTS DISPONIBLES

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

TEST FONCTIONNEL RAPIDE

Executer le bridge en mode debug et appeler device_info manuellement :

```
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:..."}
>> Si DeviceNotConnected : verifiez le cable, la LED bleue, et le nom du port avec --list-ports.
```

06 CONFIGURER LE CLIENT IA

Exécutez `pclika-setup` pour configurer automatiquement tous les clients detectes :

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

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

CLIENT	FICHIER CONFIG / COMMANDE	EMPLACEMENT
Claude Code	claude mcp add pclikaPlatform pclika-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 CHARGER LES INSTRUCTIONS

Collez le prompt systeme suivant dans votre client IA au debut d'une session (ou comme prompt systeme permanent). Cela indique a l'IA les outils disponibles et comment les utiliser.

PROMPT SYSTÈME — copier et coller

```
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 : ajoutez dans CLAUDE.md ou --system-prompt. Cursor/VS Code : Parametres → IA → Prompt systeme.
Codex/OpenCode : option --system ou champ system_prompt.
```

08 DÉCRIRE VOTRE PROJET

Une fois le prompt charge, decrivez simplement ce que vous voulez construire. L'IA planifiera l'implementation, appellera les outils materiel et generera le code firmware.

EXEMPLES D'EXIGENCES

TITRE	EXIGENCE	COMPÉTENCES
Hello World	Faire clignoter la LED en vert 3 fois, puis en bleu, puis l'eteindre.	Test du stack complet en 10 secondes.
Moniteur env.	Lire BME280 toutes les 5 s, afficher temp. et humidite sur OLED. LED rouge si >25°C	C++ affichage + logique.
Balayage servo	Faire pivoter le servo 0 de 0 a 180 degres par pas de 10, pause 500ms, journaliser	Sortie PWM + retour.
Scan Wi-Fi	Scanner les reseaux Wi-Fi, afficher tableau SSID/dBm/canal trie, surligner les reseau	analyse IA.
Projet custom	Moniteur d'arrosage : lire ADC0 chaque minute, si <30% flash rouge 5x et journaliser	ADC + LED + journalisation.

09 LA BOUCLE DE DÉVELOPPEMENT

Une fois l'exigence soumise, l'IA suit automatiquement ce flux :

#	PHASE	CE QUI SE PASSE
1	Vérification connexion	L'IA appelle device_info() avant toute action.
2	Vérification capteurs	Pour les projets capteurs, l'IA lit une fois pour confirmer.
3	Planification	L'IA explique le plan en langage simple et demande confirmation.
4	Code firmware	L'IA écrit du C ESP-IDF commenté avec les GPIO exacts de votre carte.
5	Instructions flash	L'IA vous indique la commande idf.py et ce qu'observer.
6	Test en direct	L'IA appelle les outils MCP pour vérifier le firmware en cours.
7	Iteration	En cas d'erreur, l'IA lit serial_log_read() et propose une correction.

EXEMPLE DE SESSION COMPLÈTE (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 PASSER À L'ÉCHELLE

MODE SSE MULTI-CLIENTS

Quand Cursor et VS Code sont ouverts simultanément, lancez un bridge partage :

```
# 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}
```

AJOUTER DES MODULES

MODULE	FONCTION	APPEL MCP
BME280	Temp. · humidite · pression	<code>sensor_read("bme280")</code>
MPU6050	Accelerometre · gyroscope	<code>sensor_read("imu")</code>
VL53L0X	Capteur de distance ToF	<code>sensor_read("distance")</code>
SHT31	Temp./humidite haute prec.	<code>sensor_read("sht31")</code>
PCA9685	PWM 16 voies	<code>servo_move(ch, angle)</code>
OV2640	Camera — capture JPEG	<code>camera_capture()</code>

EXTENSION FPGA (avancé)

Branchez un module iCE40UP5K dans le slot M.2. Installez le HDL Bridge pour 6 outils MCP supplémentaires :

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

ANNEXE A — RÉFÉRENCE RAPIDE DES OUTILS MCP

OUTIL	PARAMÈTRES	RETOURNE / ACTION
<code>sensor_read</code>	sensor_id: "temp_humidity" "bme280" "light" "distance" "imu" "sht31"	Renvoie les lectures du capteur en JSON
<code>display_text</code>	text: str · line: int (0–7 lignes OLED)	Ecrire sur SSD1306/ST7789
<code>servo_move</code>	channel: int (0–3) · angle: float (0–180)	Pivoter le servo en degres
<code>gpio_write</code>	pin: int · value: int (0 ou 1)	Definir GPIO haut ou bas
<code>gpio_read</code>	pin: int	Lire l'etat GPIO (0 ou 1)
<code>led_control</code>	r/g/b: int (0–255)	Couleur des 4 LED WS2812B
<code>wifi_scan</code>	(aucun parametre)	Liste des reseaux Wi-Fi proches
<code>button_read</code>	"BOOT" "RESET" "USER"	Etat bouton (0=relache, 1=appuye)
<code>device_info</code>	(aucun parametre)	MCU, firmware, IP, MAC
<code>serial_log_read</code>	lines: int (defaut 20, max 200)	Lire le log UART pour diagnostic

ANNEXE B — DÉPANNAGE

DeviceNotConnected

Conseil: 90% des cas : cable de charge sans donnees

1. La LED bleue est-elle allumee ?
2. Changez le cable (certains ne transmettent pas de donnees).
3. Lancez `--list-ports`.
4. Linux : `sudo usermod -aG dialout $USER`

Port visible mais bridge bloque

Conseil: `idf.py monitor` pour verifier le demarrage

1. Mauvais firmware — reflasher.
2. Mauvais debit — utiliser `--baud 115200`.
3. Port occupe — fermer Arduino IDE.

Client IA : serveur introuvable

Conseil: Claude Code : `/mcp` pour verifier le statut

1. Relancer `pclika-setup`.
2. Redemarrer le client IA apres modification.
3. Verifier PATH : `which pclika-bridge`

sensor_read retourne null

Conseil: Scanner I2C pour trouver l'adresse

1. Verifier branchement SDA=GPIO21, SCL=GPIO22.
2. Lancer `serial_log_read(50)`.
3. Verifier adresse capteur (BME280=0x76/0x77).

servo_move sans mouvement

Conseil: SG90 alimente directement en USB 5V

1. Servo necessite 5V, pas 3.3V.
2. Canaux : 0=GPIO38, 1=39, 2=40, 3=41.
3. Angle entre 0.0 et 180.0.

SSE : clients non connectes

Conseil: `pclika-setup --sse` reconfigure tout

1. `pclika-bridge --serve` est-il lance ?
2. Verifier `http://localhost:3141/health`.
3. Pare-feu — ajouter exception pour 3141.

SUPPORT ET RETOUR

Un problème ? Une demande de fonctionnalité ? Notre portail de support est toujours ouvert.

<https://pclika.com/Support>

CANAL	CONTACT
Portail web	https://pclika.com/Support
E-mail	starinvc@gmail.com
GitHub Issues	github.com/Pclika/mcp-platform/issues
Boutique	https://mall.pclika.com/
MCP libre-service	https://hc.pclika.com/

Votre retour façonne directement la prochaine version. Chaque rapport de bug et demande compte.

github.com/Pclika/mcp-platform

pypi.org/project/pclika-bridge

Boutique : <https://mall.pclika.com/> · MCP self-service : <https://hc.pclika.com/>

Apache 2.0 (logiciel) · CERN-OHL-S v2 (materiel) · © 2026 Pclika

Pclika