Pclika

Руководство разработчика

От платы до ИИ — полная настройка и рабочий процесс

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

VERSION · 0.1.0 · Июнь 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

Это руководство проведёт вас от покупки оборудования до выдачи первой команды ИИ-ассистента устройству. Опыт работы с железом не требуется.

00 ОБЗОР

Pclika превращает плату ESP32-S3 в аппаратную платформу под управлением ИИ. Ассистент вызывает функции железа через стандартные MCP-инструменты (JSON-RPC 2.0) по USB в реальном времени.

ШАГ	НАЗВАНИЕ	ОПИСАНИЕ
01	Купить набор	Заказать в магазине или через MCP-сервис
02	Подключить оборудование	USB-C к ПК, I2C-датчики к JST SH
03	Прошить firmware	Одна команда: idf.py flash — ~90 секунд
04	Установить Bridge	pip install pclika-bridge — готово
05	Проверить соединение	pclika-bridge --list-ports, затем device_info
06	Настроить ИИ-клиент	Claude Code / Codex / Cursor / VS Code / OpenCode
07	Загрузить инструкции	Вставить системный промпт в клиент один раз
08	Описать задачу	Написать требование — ИИ приступает к коду
09	Итерировать и деплоить	ИИ пишет прошивку и MCP-вызовы — вы проверяете
10	Масштабировать	Добавлять модули, использовать SSE-режим

01 КУПИТЬ НАБОР

Самый простой способ — стартовый набор Pclika: плата и датчики, проверенные на всех примерах.

ОФИЦИАЛЬНЫЕ КАНАЛЫ ПОКУПКИ

Магазин для пользователей	**https://mall.pclika.com/**
MCP самообслуживание	**https://hc.pclika.com/**

Стартовый набор ¥299 (Рекомендуется)

ПОЗИЦИЯ	КОЛ.	ПРИМЕЧАНИЯ
Pclika ESP32-S3 Baseboard v0.1	1 шт.	Основной контроллер · USB-C · Wi-Fi · BLE
Датчик DHT22 темп./влажн.	1 шт.	I2C · пример env-monitor
Датчик освещённости BH1750	1 шт.	I2C · измерение люкс
Кабель USB-C (1 м)	1 шт.	Кабель с данными, не зарядный
Кабель JST SH 4P × 3	3 шт.	Подключение I2C-датчиков

DIY / ИЗГОТОВЛЕНИЕ ПЛАТЫ

Файлы KiCad и руководство JLCPCB доступны в репозитории GitHub.

ДОПОЛНИТЕЛЬНЫЕ МОДУЛИ

МОДУЛЬ	ПРИМЕНЕНИЕ
Сервопривод SG90 × 4	PWM-выходы — пример servo-control
OLED SSD1306 128x64	I2C-дисплей — инструмент display_text
Камера OV2640	Камера — пример vision-snapshot
Адаптер RS-485	Modbus RTU — пример industrial-gateway
FPGA iCE40UP5K M.2	Расширение FPGA — HDL Bridge

02 ПОДКЛЮЧИТЬ ОБОРУДОВАНИЕ

ФИЗИЧЕСКОЕ ПОДКЛЮЧЕНИЕ

УСТРОЙСТВО	ПОДКЛЮЧЕНИЕ	ПРИМЕЧАНИЕ
Кабель USB-C	Разъём J1 → USB ПК	Питание + последовательный порт
DHT22/BME280	SDA=GPIO21 SCL=GPIO22 JST SH 4P	Пример env-monitor
OLED SSD1306	Та же шина I2C, адрес 0x3C	Инструмент display_text
Сервопривод	Разъёмы PWM0–PWM3	Инструмент servo_move

СОСТОЯНИЕ LED ПЛАТЫ

СОСТОЯНИЕ LED	ЗНАЧЕНИЕ
Синий постоянно	Питание включено, прошивка работает
Медленное мерцание	Bridge в режиме ожидания MCP-команд
Быстрое мерцание	Активная MCP-сессия, принимаются вызовы
Красный горит	Ошибка — проверьте serial_log_read

>> Используйте кабель с данными — зарядные кабели не имеют линий данных и не отображаются в списке портов.

03 ПРОШИТЬ FIRMWARE

ПРЕДВАРИТЕЛЬНЫЕ ТРЕБОВАНИЯ

- Python 3.10+ (python3 --version)

- ESP-IDF v5.x (docs.espressif.com/projects/esp-idf)

- idf.py в PATH (idf.py --version)

- Плата Pclika подключена через кабель USB-C с данными

КЛОНИРОВАТЬ И ПРОШИТЬ

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

ГОТОВАЯ ПРОШИВКА (без ESP-IDF)

Скачайте .bin из GitHub Releases и прошейте через esptool:

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

>> Windows: если idf.py flash сообщает о занятом порту, закройте Arduino IDE, PuTTY и т.д.

04 УСТАНОВИТЬ BRIDGE

```
pip install pclika-bridge

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

ЧТО УСТАНАВЛИВАЕТСЯ

КОМАНДА / ПАКЕТ	НАЗНАЧЕНИЕ
pclika-bridge	Команда CLI — запуск MCP-сервера (STDIO или SSE)
pclika-setup	Команда CLI — авто-конфигурация всех ИИ-клиентов
pclika_bridge	Python-библиотека — импорт в собственные скрипты

РЕЖИМЫ ПОДКЛЮЧЕНИЯ

РЕЖИМ	ОПИСАНИЕ	ОПТИМАЛЬНО
STDIO (по умолчанию)	Каждый клиент запускает свой процесс bridge	Просто, работает сразу
SSE / Адаптивный	Один процесс bridge для всех клиентов	Для одновременной работы

05 ПРОВЕРИТЬ СОЕДИНЕНИЕ

СПИСОК ДОСТУПНЫХ ПОРТОВ

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

БЫСТРЫЙ ФУНКЦИОНАЛЬНЫЙ ТЕСТ

Запустите bridge в режиме отладки и вызовите device_info вручную:

```
pclika-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:..."}
```

>> Если DeviceNotConnected: проверьте кабель, убедитесь что синий LED горит, порт совпадает с --list-ports.

06 НАСТРОИТЬ ИИ-КЛИЕНТ

Запустите pclika-setup для авто-конфигурации всех обнаруженных клиентов:

```
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
```

КЛИЕНТ	ФАЙЛ / КОМАНДА	РАСПОЛОЖЕНИЕ
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 ЗАГРУЗИТЬ ИНСТРУКЦИИ

Вставьте следующий системный промпт в ИИ-клиент в начале сессии (или как постоянный системный промпт). Это сообщает ИИ какие инструменты доступны и как ими пользоваться.

СИСТЕМНЫЙ ПРОМПТ — скопируйте и вставьте

```
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: добавьте в CLAUDE.md или --system-prompt. Cursor/VS Code: Настройки → ИИ → Системный промпт. Codex/OpenCode: флаг --system или поле system_prompt в конфиге.

08 ОПИСАТЬ ЗАДАЧУ

После загрузки промпта просто опишите что хотите сделать. ИИ распланирует реализацию, вызовет инструменты для проверки датчиков, сгенерирует прошивку и проведёт вас через прошивку.

ПРИМЕРЫ ТРЕБОВАНИЙ

НАЗВАНИЕ	ТРЕБОВАНИЕ	НАВЫКИ
Hello World	Мигнуть встроенным LED зелёным 3 раза, затем синим, затем выключить.	Тест всего стека за 10 секунд.
Мониторинг	Читать BME280 каждые 5 секунд, показывать темп. и влажность на OLED. При >28°C LED красный.	Датчик + дисплей + логика.
Серво-скан	Двигать серво 0 от 0° до 180° с шагом 10°, пауза 500мс, записывать угол в лог.	PWM + обратная связь.
Wi-Fi сканер	Сканировать Wi-Fi, вывести таблицу SSID/дБм/канал по убыванию сигнала.	РЧ-данные → анализ ИИ.
Свой проект	Мониторинг полива растений: читать ADC0 каждую минуту, при <30% мигать красным 5 раз и логировать с временной меткой.	ADC + LED + лог.

09 ПОЛНЫЙ ЦИКЛ РАЗРАБОТКИ

После отправки требования ИИ автоматически выполняет следующий рабочий процесс:

#	ФАЗА	ЧТО ПРОИСХОДИТ
1	Проверка связи	ИИ вызывает device_info() прежде чем делать что-либо.
2	Проверка датчиков	Для проектов с датчиками ИИ читает их один раз для подтверждения.
3	Планирование	ИИ объясняет план реализации и запрашивает подтверждение.
4	Код прошивки	ИИ пишет ESP-IDF C с комментариями, точные GPIO вашей платы.
5	Инструкция прошивки	ИИ сообщает команду idf.py и что ожидать в мониторе.
6	Живой тест	ИИ вызывает MCP-инструменты для проверки работающей прошивки.
7	Итерация	При ошибке ИИ читает serial_log_read() и предлагает исправление.

ПОЛНЫЙ ПРИМЕР СЕССИИ (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 МАСШТАБИРОВАТЬ

SSE-РЕЖИМ ДЛЯ НЕСКОЛЬКИХ КЛИЕНТОВ

При одновременном использовании Cursor и VS Code запустите один общий bridge:

```
# 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}
```

ДОБАВИТЬ МОДУЛИ

МОДУЛЬ	ФУНКЦИЯ	ВЫЗОВ MCP
BME280	Темп. · влажность · давление	`sensor_read("bme280")`
MPU6050	Акселерометр · гироскоп · IMU	`sensor_read("imu")`
VL53L0X	Дальномер (ToF)	`sensor_read("distance")`
SHT31	Высокоточный темп./влажн.	`sensor_read("sht31")`
PCA9685	16-кан. PWM (больше серво)	`servo_move(ch, angle)`
OV2640	Камера — захват JPEG	`camera_capture()`

РАСШИРЕНИЕ FPGA (продвинутый уровень)

Установите модуль iCE40UP5K в слот M.2. Установите HDL Bridge для 6 дополнительных MCP-инструментов:

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

ПРИЛОЖЕНИЕ A — СПРАВОЧНИК MCP-ИНСТРУМЕНТОВ

ИНСТРУМЕНТ	ПАРАМЕТРЫ	ВОЗВРАЩАЕТ / ДЕЙСТВИЕ
`sensor_read`	sensor_id: "temp_humidity"|"bme280"|"light"|"distance"|"imu"|"sht31"	Возвращает показания датчика JSON
`display_text`	text: str · line: int (0–7 строк OLED)	Запись на SSD1306/ST7789
`servo_move`	channel: int (0–3) · angle: float (0–180)	Поворот серво в градусах
`gpio_write`	pin: int · value: int (0 или 1)	Установить GPIO высокий/низкий
`gpio_read`	pin: int	Прочитать состояние GPIO
`led_control`	r/g/b: int (0–255)	Установить цвет 4x WS2812B LED
`wifi_scan`	(без параметров)	Список ближайших Wi-Fi сетей
`button_read`	"BOOT"|"RESET"|"USER"	Состояние кнопки (0=отпущена, 1=нажата)
`device_info`	(без параметров)	MCU, прошивка, IP, MAC
`serial_log_read`	lines: int (по умолч. 20, макс. 200)	Читать UART-лог для диагностики

ПРИЛОЖЕНИЕ Б — УСТРАНЕНИЕ НЕПОЛАДОК

DeviceNotConnected	Подсказка: В 90% случаев — зарядный кабель без данных

1. Горит ли синий LED?
2. Замените кабель на кабель с данными.
3. Запустите --list-ports.
4. Linux: sudo usermod -aG dialout $USER

Порт виден, bridge зависает	Подсказка: idf.py monitor для просмотра загрузки

1. Неверная прошивка — перепрошейте.
2. Неверная скорость — используйте --baud 115200.
3. Порт занят другой программой — закройте Arduino IDE.

Клиент не находит сервер	Подсказка: Claude Code: /mcp для проверки статуса

1. Перезапустите pclika-setup.
2. Перезапустите ИИ-клиент после изменения конфига.
3. Проверьте PATH: which pclika-bridge

sensor_read возвращает null	Подсказка: I2C-сканер поможет найти адрес устройства

1. Проверьте подключение: SDA=GPIO21, SCL=GPIO22.
2. Запустите serial_log_read(50).
3. Проверьте адрес датчика (BME280=0x76/0x77).

servo_move не работает	Подсказка: SG90 питание от USB 5В напрямую

1. Серво требует 5В, не 3.3В.
2. Каналы: 0=GPIO38, 1=39, 2=40, 3=41.
3. Угол должен быть 0.0–180.0.

SSE: клиенты не подключаются	Подсказка: pclika-setup --sse перезапишет все конфиги

1. Убедитесь, что pclika-bridge --serve запущен.
2. Откройте http://localhost:3141/health.
3. Брандмауэр — добавьте исключение для 3141.

ПОДДЕРЖКА И ОБРАТНАЯ СВЯЗЬ

Возникла проблема? Есть запрос на функцию или предложение? Наш портал поддержки всегда открыт.

https://pclika.com/Support

КАНАЛ	КОНТАКТ
Веб-портал	https://pclika.com/Support
Электронная почта	starinvc@gmail.com
GitHub Issues	github.com/Pclika/mcp-platform/issues
Магазин	https://mall.pclika.com/
MCP-сервис	https://hc.pclika.com/

Ваша обратная связь напрямую формирует следующий релиз. Каждый отчёт об ошибке и запрос важны.

github.com/Pclika/mcp-platform

pypi.org/project/pclika-bridge

Магазин: https://mall.pclika.com/ · MCP-сервис: https://hc.pclika.com/

Pclika