



SPEAKER & TALK PROPOSAL

Microcontrollers 101: Your Gateway to Hardware Hacking

Exploring embedded security, hardware protocols, and CTF preparation with ESP32.

 **Speaker: M0gul**

 45 Minute Keynote

 Embedded Security

\$ whoami

M0gul

Head, Hardware Hacking Village PH · Founder, 2023

- > Penetration tester by trade — now VAPT service-delivery lead.
- > Started the Hardware Hacking Village in 2023.
- > The handle 'M0gul'? semprix gave it to me — and it stuck.

Not an engineer — just someone who likes to tinker, build things, and figure out how they work.



What We'll Cover

01

Why hardware security matters

Real breaches, and why holding the device changes everything.

02

The low budget hacking lab

Every tool you need, for the price of a pizza.

03

UART — the front door

Two wires to a device's secrets. The highest-reward skill.

04

I2C, SPI & firmware

Reading memory chips and pulling secrets out of firmware.

05

Wireless & the ESP32 radio

Wi-Fi and Bluetooth as both target and tool.

06

The Village CTF

8 flags waiting — and how everything maps to them.

Why Hardware Security Matters



Software Penetration Testing

Restricted by user permissions, network firewalls, and operating system access controls.



Physical Hardware Access

Bypasses OS restrictions through direct memory reads, pin probes, and serial interface exploitation.

Real-World Wake-Up Calls

Mirai Botnet, 2016

Hijacked ~600,000 IoT gadgets — cameras, DVRs — mostly with default passwords baked into firmware. Took Twitter, Netflix and Reddit offline.

Cloned Car Keys

Key fobs and engine immobilizers cracked by sniffing the chips inside — the car thinks the thief's copy is the real key.

Leaky Smart Homes

Wi-Fi passwords and cloud tokens pulled straight out of smart-plug and doorbell flash memory — the exact skill we practice today.

Beyond Gadgets

The same moves have opened up medical devices and voting machines. Anything with a chip is a computer you can talk to.

Microcontroller vs. Single-Board Computer

Two different animals. A microcontroller runs YOUR code straight on the metal; an SBC is a tiny computer running a full operating system.

	Microcontroller (ESP32)	Single-Board Computer (Pi)
Brain	One program on bare metal / tiny RTOS	Full OS (Linux) running many processes
Boot time	Instant - micro- to milliseconds	20-40 seconds to boot
Power	Micro-amps; can sip from a coin cell	100s of mA to amps; needs a real supply
Compute	MHz cores, kilobytes of RAM	GHz cores, gigabytes of RAM
Storage	Built-in flash (KB-MB)	microSD / eMMC (GB)
Cost	\$1 - \$5	\$35 - \$100+
Hacker's view	Code IS the device - open serial, dumpable flash	A small computer - attack like a Linux box

Selecting Your Hacking Platform

ESP32 / ESP8266

\$2-5

Dual-core with Wi-Fi + Bluetooth built in.

BEST AT

Wireless - and being target AND tool

TYPICAL USE

IoT, Wi-Fi/BLE audits, this CTF

RP2040 / RP2350

\$4

Raspberry Pi's own chip with Programmable I/O.

BEST AT

Bit-banging odd protocols & glitching

TYPICAL USE

Custom signals, logic, PIO tricks

STM32

\$2-10

Industrial ARM Cortex-M workhorse.

BEST AT

Raw performance & rich peripherals

TYPICAL USE

Appliances, motors, CAN / automotive

AVR / ATmega

\$2-5

The classic 8-bit Arduino Uno chip.

BEST AT

Being simple & beginner-friendly

TYPICAL USE

Learning, simple builds, big community

HOW TO CHOOSE

Need Wi-Fi or Bluetooth? -> **ESP32**

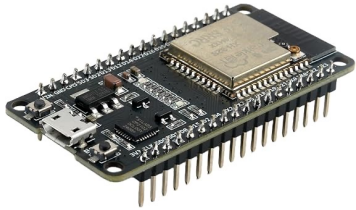
Precise signal timing or glitching? -> **RP2040 (PIO)**

Industrial / automotive, CAN bus? -> **STM32**

Rule of thumb: start with the ESP32 - cheapest path to real results, and it is what this CTF runs on.

The Low Budget Hardware Hacking Lab

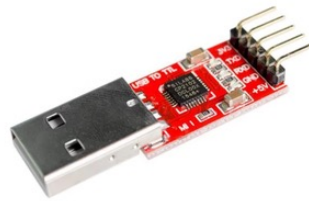
You do not need an oscilloscope. Four cheap parts solve almost every hardware CTF - no soldering required.



ESP32 Dev Board

~ Php 240

Your target and your tool - a tiny computer with Wi-Fi/BLE you can program, probe and break.



USB-to-UART Adapter

~ Php 150

The cable that lets your laptop talk to a chip's serial port. CP2102 or FTDI. Your #1 tool.



Logic Analyzer (8ch)

~ Php 390

Records the wiggles on a wire so PulseView can decode UART, I2C and SPI traffic.



SOIC8 Clip + Jumpers

~ Php 330

Grip a flash chip's 8 legs - no soldering. Clip on, dump firmware, clip off.

UART - The Serial Console

WHAT IT IS

Universal Asynchronous Receiver/Transmitter - a 2-wire way for two chips to swap bytes as a stream. TX out, RX in, shared GND. 'Asynchronous' means no clock wire, so both sides must agree on the speed (the baud rate) in advance.

WHAT IT'S FOR

Debug & boot consoles, log output, GPS modules, chip-to-PC links - and on many devices, a full command shell.

WHAT DATA YOU'LL SEE

Human-readable text: bootloader and kernel logs, debug prints, login prompts - and sometimes a root shell with no password at all.

```

serial_console

$ picocom -b 115200 /dev/ttyUSB0
U-Boot 2022.04 (Hacking Village Build)
DRAM: 128 MiB
Hit any key to stop autoboot: 0
U-Boot> setenv bootargs init=/bin/sh
U-Boot> boot

# id
uid=0(root) gid=0(root)
```

// CTF CLUE · Serial Flags (x2) - CTF

- > A chatty board on the obvious pins, at a common speed, nearly reads you its secret - mind its parting hint.
- > A second voice is tuned to an unusual speed and may not be on the pins you first tried. Garbled = wrong pace.

I2C & SPI - The Chip-to-Chip Buses

I2C = Inter-Integrated Circuit

WHAT IT IS

2-wire synchronous bus: SDA (data) + SCL (clock). Many chips share it; each answers to its own address.

USED FOR

Sensors, EEPROMs, real-time clocks, small OLED displays.

DATA YOU'LL SEE

Register reads/writes and memory contents - config, calibration, sometimes credentials.

SPI = Serial Peripheral Interface

WHAT IT IS

4-wire synchronous bus: MOSI, MISO, SCK + a select line per chip. One controller, very fast, full-duplex.

USED FOR

Flash memory, SD cards, displays, fast ADCs.

DATA YOU'LL SEE

Raw memory - dumping an SPI flash gives you the ENTIRE firmware: bootloader, OS, filesystem, secrets.

// CTF CLUE · Silent Conversation - CTF

- > A secret rides a two-wire bus your serial cable can't hear.
- > Bring the bus to life, find who's listening on it, and ask the right address to hand it over.

UART vs. I2C vs. SPI vs. JTAG

Four ways to reach a chip. Ask: how many devices, how fast - and how deep does it get you?

UART

Universal Asynchronous Receiver/Transmitter

2 wires · async · point-to-point
Speed: up to ~115 kbps

STRENGTHS

- + TX, RX, GND - that's it
- + No clock; on everything

TRADE-OFFS

- Only 2 devices
- Must guess the baud

BEST FOR

Consoles, GPS, boot logs, shells.

I2C

Inter-Integrated Circuit

2 wires · shared clock · addressed
Speed: up to 3.4 Mbps

STRENGTHS

- + Many chips on 2 wires
- + Pin-efficient; has ACKs

TRADE-OFFS

- Slower; needs pull-ups
- Half-duplex

BEST FOR

Sensors, EEPROMs, clocks, OLEDs.

SPI

Serial Peripheral Interface

4 wires · shared clock · CS/chip
Speed: up to 80 Mbps

STRENGTHS

- + Fastest; full-duplex
- + Simple, high throughput

TRADE-OFFS

- More wires; 1 CS/chip
- No addressing

BEST FOR

Flash, SD cards, displays.

JTAG

Joint Test Action Group

4-5 wires · debug / boundary-scan
Speed: up to 30 Mbps

STRENGTHS

- + Deepest chip access
- + Halt CPU, R/W memory

TRADE-OFFS

- Often hidden/disabled
- Pins need discovery

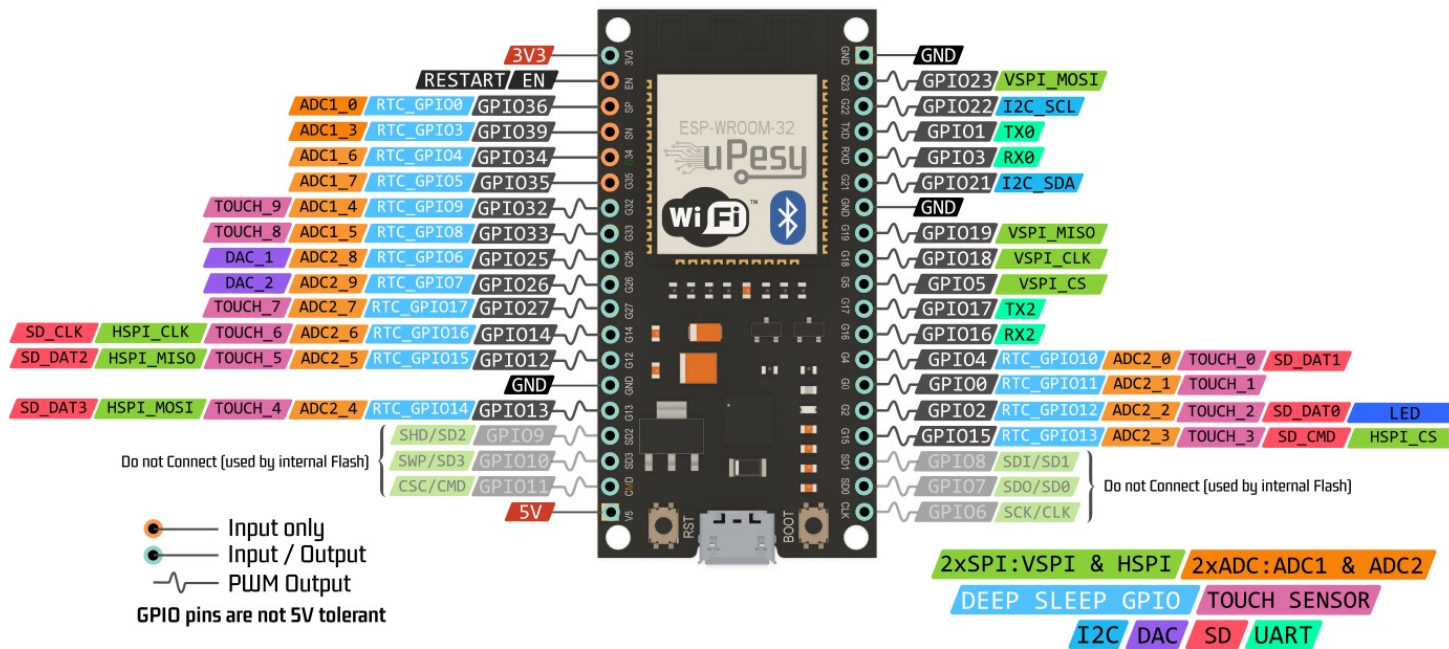
BEST FOR

Debugging, memory dumps, defeating locks.

HACKER'S LENS UART -> shells I2C -> config/EEPROM SPI -> firmware dumps JTAG -> deep chip control

ESP32 - UART, I2C, SPI & JTAG Pins

ESP32 Wroom DevKit Full Pinout



UART

UART0: GPIO1 (TX) · GPIO3 (RX)
 UART2: GPIO17 (TX) · GPIO16 (RX)

I2C

SDA: GPIO21 SCL: GPIO22
 (default pins - remappable)

SPI

MOSI 23 · MISO 19 · SCK 18 · CS 5
 (VSPI default)

JTAG

TDI 12 · TCK 13 · TMS 14 · TDO 15
 (MTDI / MTCK / MTMS / MTDO)

Heads-up: most ESP32 pins are remappable in firmware - these are the common DevKit defaults.
 Always check your board's silkscreen.

Firmware Extraction & Analysis Pipeline



STEP 01

Locate Flash IC

Identify SOIC8 or WSON memory chip on board.

STEP 02

Clip & Dump

Read memory via Flashrom and programmer clip.

STEP 03

Unpack Image

Extract filesystem with Binwalk utility.

STEP 04

Extract Flags

Find API keys, credentials, and secret flags.

// CTF CLUE · Three Flags in the Firmware

- > Pull the whole image and read what's sitting there in plain sight.
- > One block looks like gibberish - tidy letters and digits, maybe padded with '='. Not encryption; a reversible encoding.
- > A labelled vault holds the last; the label is readable, the value twisted by one simple operation. Undo it.

Wi-Fi & Bluetooth Reconnaissance

14 / 20

802.11 Frame Injection

ESP32 custom firmware allows raw Wi-Fi frame craft and deauthentication monitoring.

BLE Device Enumeration

Scan GATT services, analyze advertisement beacons, and intercept IoT sensors.

Rogue Access Point Deployment

Host rogue captive portals directly on \$4 hardware chips.

// CTF CLUE · The Hidden Portal

> A board quietly joined the Wi-Fi and serves a page. Read the note it keeps for crawlers, then open the door it hides.



The Village CTF — 8 Flags Await

Two warm-ups, then it ramps. You've seen the clues on the protocol slides — here's the whole board.

1**Warm-Up Signal**

Easy serial flag on the obvious pins.

2**Crossed Wires**

A second serial voice at an unusual speed.

3**The Hidden Portal**

A web page hiding a door from crawlers.

4**Skeletons in the Flash**

A secret sitting in the firmware in plain sight.

5**Hidden in Plain Sight**

A base64 blob that decodes in one step.

6**The Scrambled Vault**

A stored value twisted by one simple operation.

7**Silent Conversation**

A flag on a 2-wire bus your cable can't hear.

8**Access Denied**

A locked touchscreen that's quietly counting.

Hack Curious, Hack Legal

1**Only what you own**

Hack devices you own or have explicit, written permission to test. The boards at this village are yours to break — go wild. A random product on a shelf is not.

2**Responsible disclosure**

Find a real flaw in a real product? Tell the vendor privately and give them time to fix it before you go public. That's how we make things safer.

3**Purpose over damage**

We open hardware to understand and defend it — not to cause harm. The skill is powerful; the intent is what keeps it ethical.

The Hardware Hacker's Roadmap



1. Start Budget-Friendly

Build your lab with \$5 ESP32 boards and low-cost logic analyzers.



2. Master Bus Logic

Understand UART, I2C, and SPI before buying expensive oscilloscopes.



3. Attack Village CTFs

Apply serial shell and firmware extraction concepts directly on real targets.

Hardware Hacking Resources

Where to learn more, the tools you'll actually use, and where the community lives.

LEARN

Random Nerd Tutorials

randomnerdtutorials.com

OWASP Firmware Testing (FSTM)

<https://github.com/scriptingxss/owasp-fstm>

The Hardware Hacking Handbook

IoT Pentest Methodology

<https://github.com/mohamed-algabry/IoT-Pentest-Methodology>

TOOLS

esptool - flash the ESP32

github.com/espressif/esptool

sigrok / PulseView

sigrok.org

binwalk - firmware extraction

github.com/ReFirmLabs/binwalk

Ghidra - reverse engineering

ghidra-sre.org

flashrom - read SPI flash

flashrom.org

REFERENCE & COMMUNITY

Espressif ESP32 Docs

docs.espressif.com

Hackaday

hackaday.com

r/hardwarehacking

reddit.com/r/hardwarehacking



Thank You & Happy Hacking!

Questions & Answer Session



Speaker: M0gul



Hardware Hacking Village PH CTF

CONNECT WITH HHV PH



Discord



Facebook



Instagram



YouTube