CryptoGen Nepal

Saycure.

Introduction to Hardware Hacking

Where to Begin

- Nirmal Dahal **(#Nittam)**

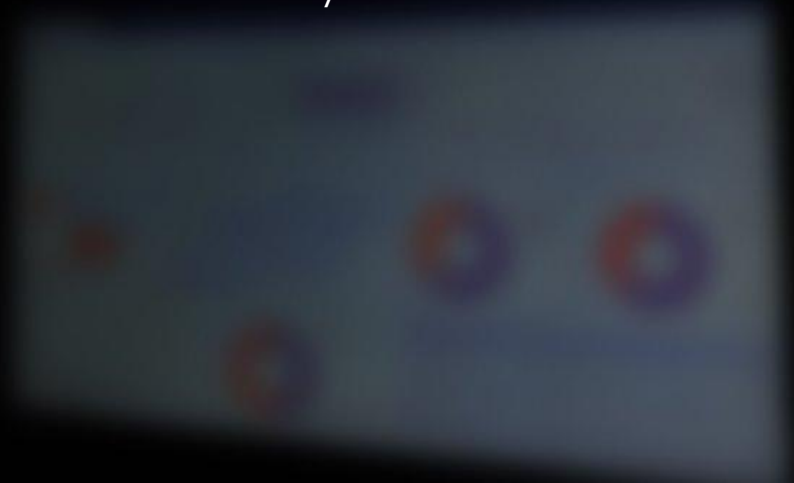
/TheNittam.me

nirmaldahal.com.np

/TheNittam

What is **Hardware Hacking**?

Hardware hacking is the art of modifying or manipulating the physical components of electronic devices to achieve specific goals, which can range from gaining unauthorized access to a system to enhancing device functionality.



Scope of **Hardware Hacking**

We can consider **SCADA system hacking, embedded system hacking, and IoT hacking** as forms of hardware hacking because **they all involve the manipulation or modification of the physical electronic hardware components within their respective systems to some extent**. While each of them has its unique emphasis, requiring specific knowledge and techniques, **they all share a common foundation: the interaction with the tangible hardware elements that power electronic systems.**

Why is Hardware Hacking **Crucial**?

Businesses tend to underestimate the security considerations associated with IoT devices, often incorporating them into the scope of Network Penetration Testing. In such cases, penetration testers, who may lack expertise in IoT or hardware hacking, typically concentrate on pinpointing vulnerabilities within open ports and services. **Consequently, this approach leaves the IoT device susceptible to potential exploitation through hardware-based attacks.**

Prerequisite **Knowledge**

Because hardware hacking revolves around manipulating the electrical input equilibrium within electronic components, it's essential to have a foundational grasp of electric and electronic components. This knowledge includes an understanding of:

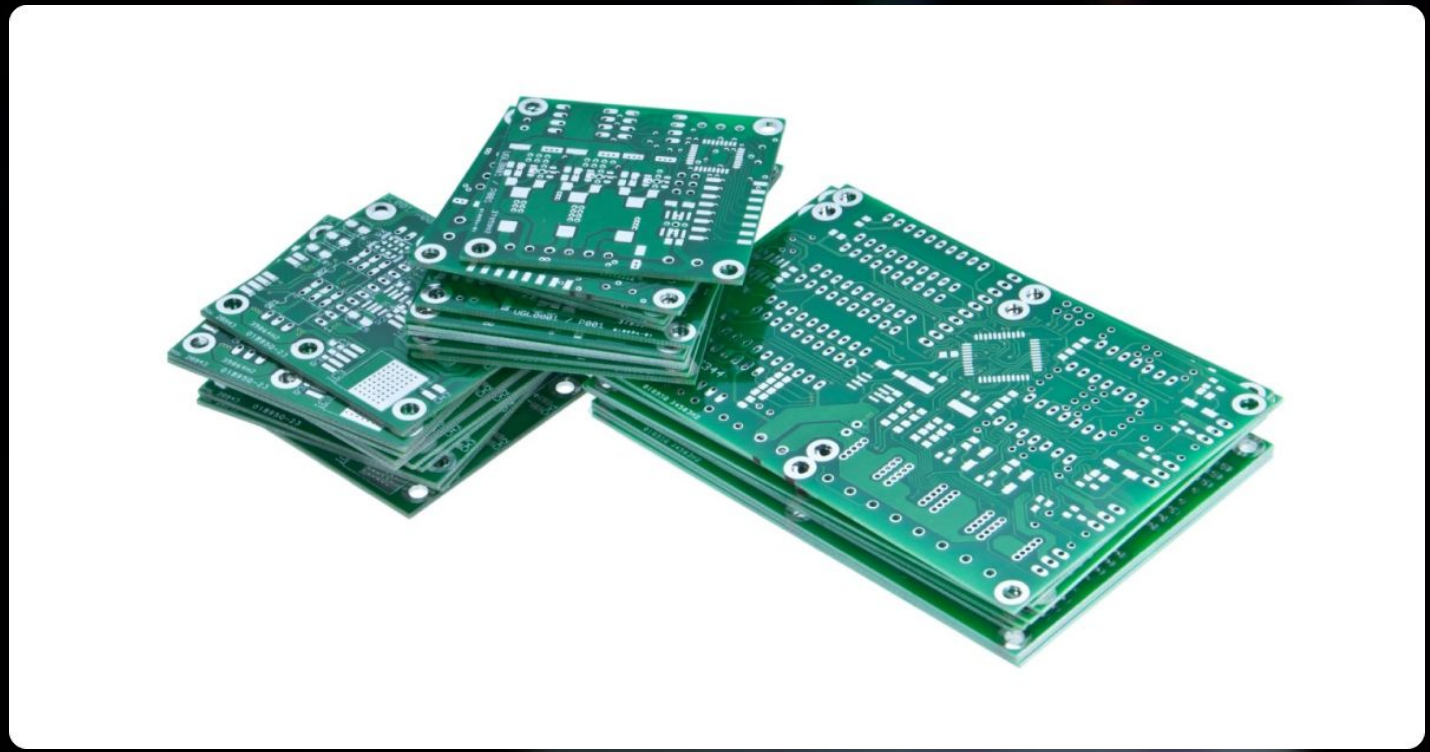
- **Voltage**
- Current (Ampere)
- **Baud Rate**
- Microcontrollers
- Sensors
- Firmware
- Sensors

Components **Involved**

Hardware hacking involves manipulation of various elements:

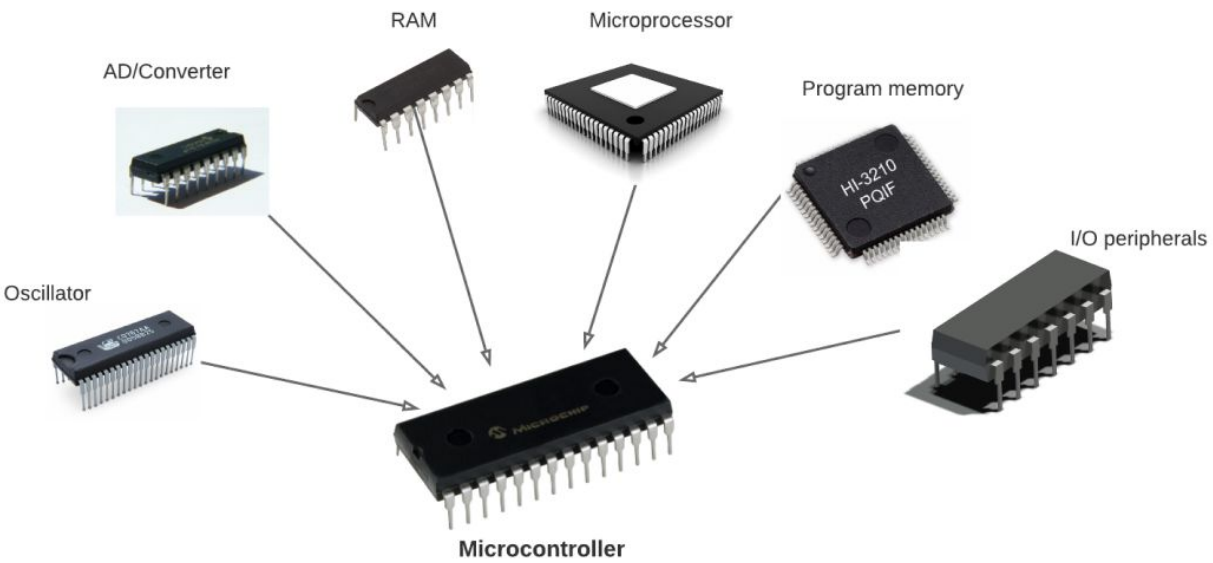
- **Circuit Board**
- **Microcontrollers**
- **Sensors**
- **Electronic Components**
  - Transistors
  - Diodes
  - Resistors
  - Capacitors
- **Firmware**
- ***uBoot***

Circuit **Board**



Circuit board is like **the roads where electrical signals travel**, connecting all the important parts.

Micro**controllers**

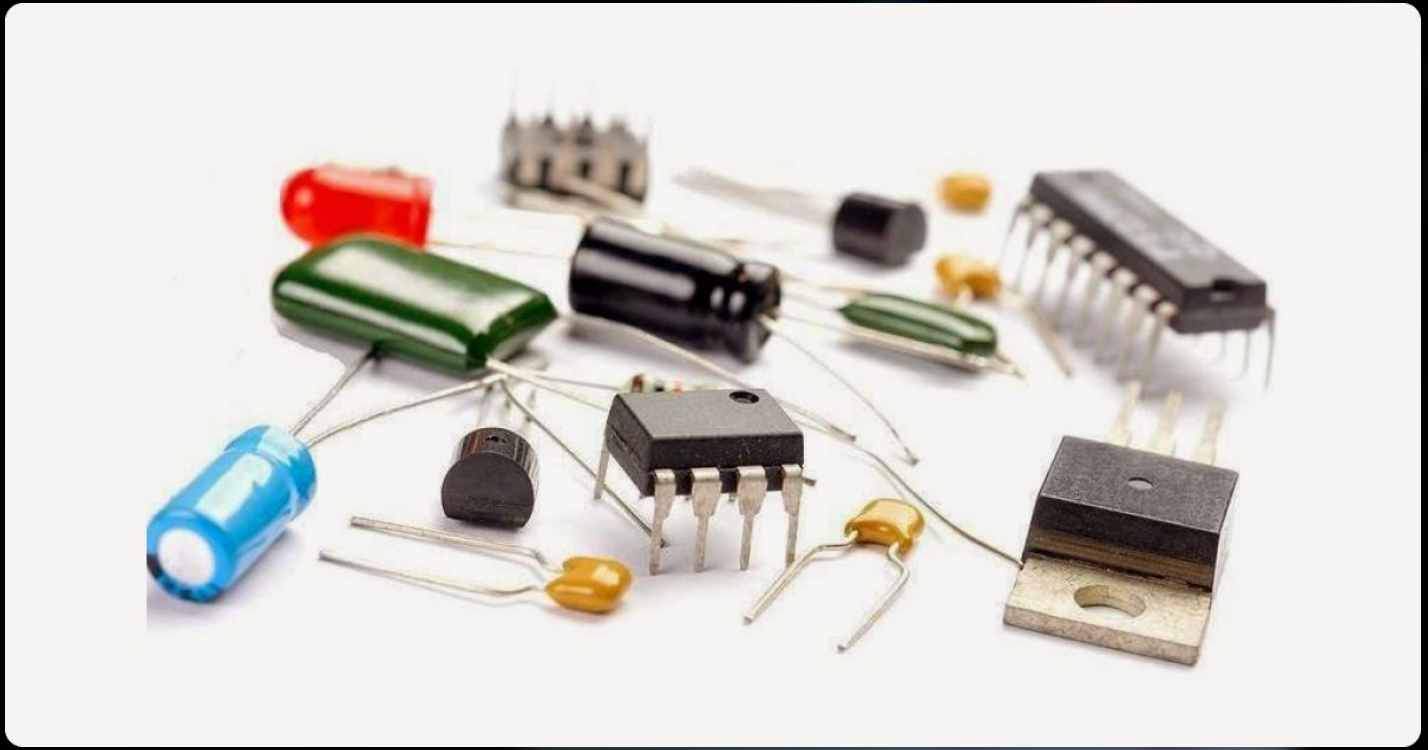


Microcontrollers are in charge of doing tasks and keeping everything running smoothly. **People often target them when they want to change how a device works.**

Sensors

Sensors can **feel and understand what's happening in the real world**, which is super important for both good and not-so-good tinkering.

Electronic **Components**



Electronic Components are responsible for **how a device behaves or what it can do.**

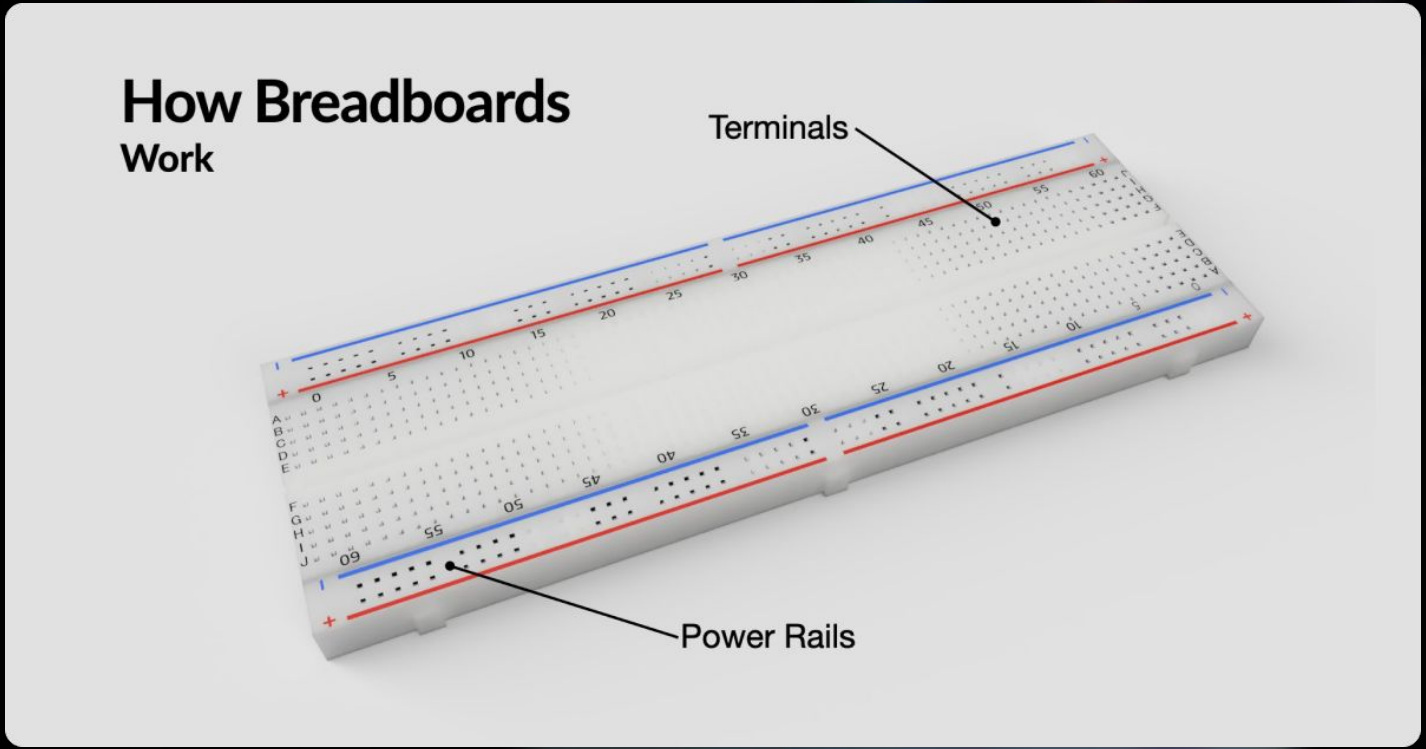
Firmware

```
// Copyright (c) Microsoft. All rights reserved.
// Licensed under the MIT license.
#include "AZ3166WiFi.h"
#include "AzureIotHub.h"
#include "DevKitMQTTClient.h"
#include "DevkitDPSClient.h"

#include "config.h"
#include "utility.h"

// Input DPS instance info
char* Global_Device_Endpoint = "[Global_Device_Endpoint]";
char* ID_Scope = "[ID_Scope]";

// Input your preferrred registrationId and only alphanumeric, lowercase, and hyphen are supporte
// If you leave it blank, one registrationId would be auto-generated based on MAC address and fir
char* registrationId = "";

// Indicate whether WiFi is ready
static bool hasWifi = false;
int messageCount = 1;
static bool messageSending = true;
```

Firmware is like **the secret recipe** that makes a device work. Sometimes, **people try to change it to uncover hidden features or to mess with a device's security.**

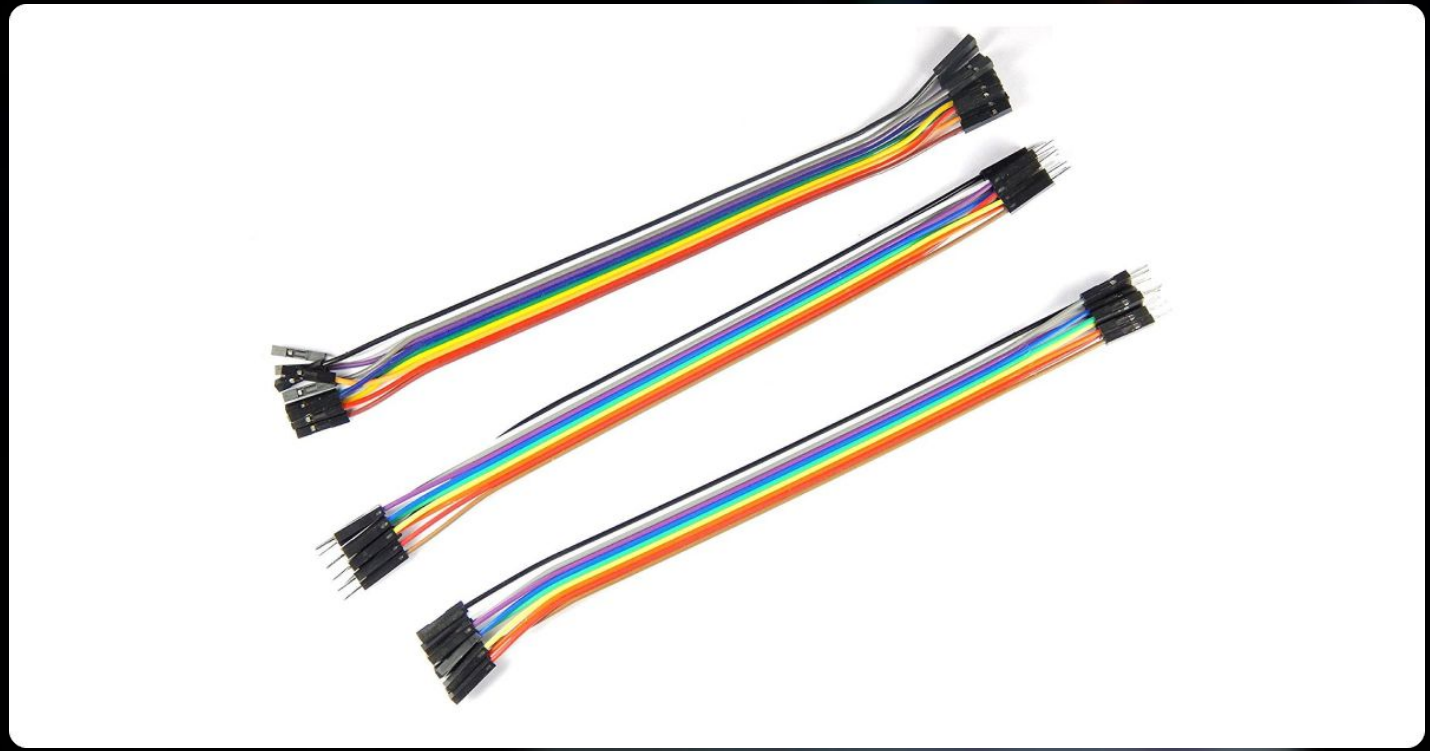

How to **Get Started**?
Get Familiar with the **Tools First**!

Bread**Board**



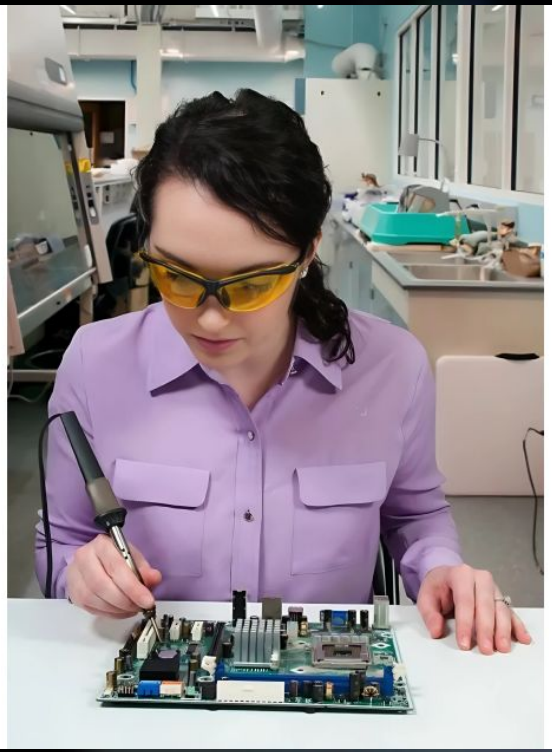
Breadboard serves as a user-friendly and straightforward **platform for constructing and evaluating electronic circuits without the need for soldering**

Jumper **Wire**



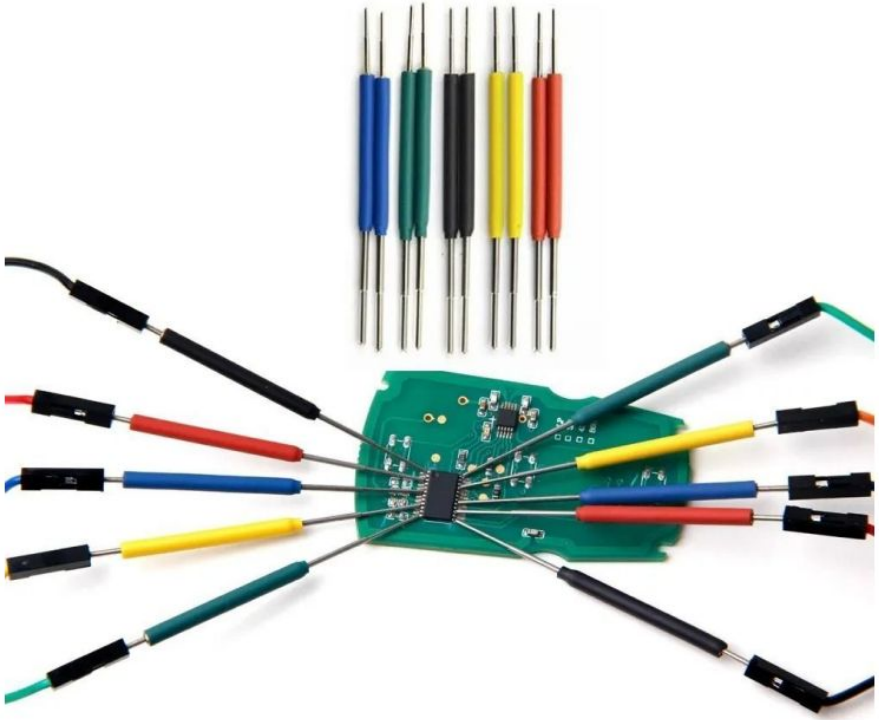
Jumper wire is a wire with connectors at both ends that **allows you to connect two points on a breadboard**.

Soldering **Iron**



A soldering iron is a crucial tool in hardware hacking, serving as a heating device that melts soldering wire. **It has a dual purpose, allowing for both the attachment and removal of electrical components and wires.**

Micro SMD Grabber

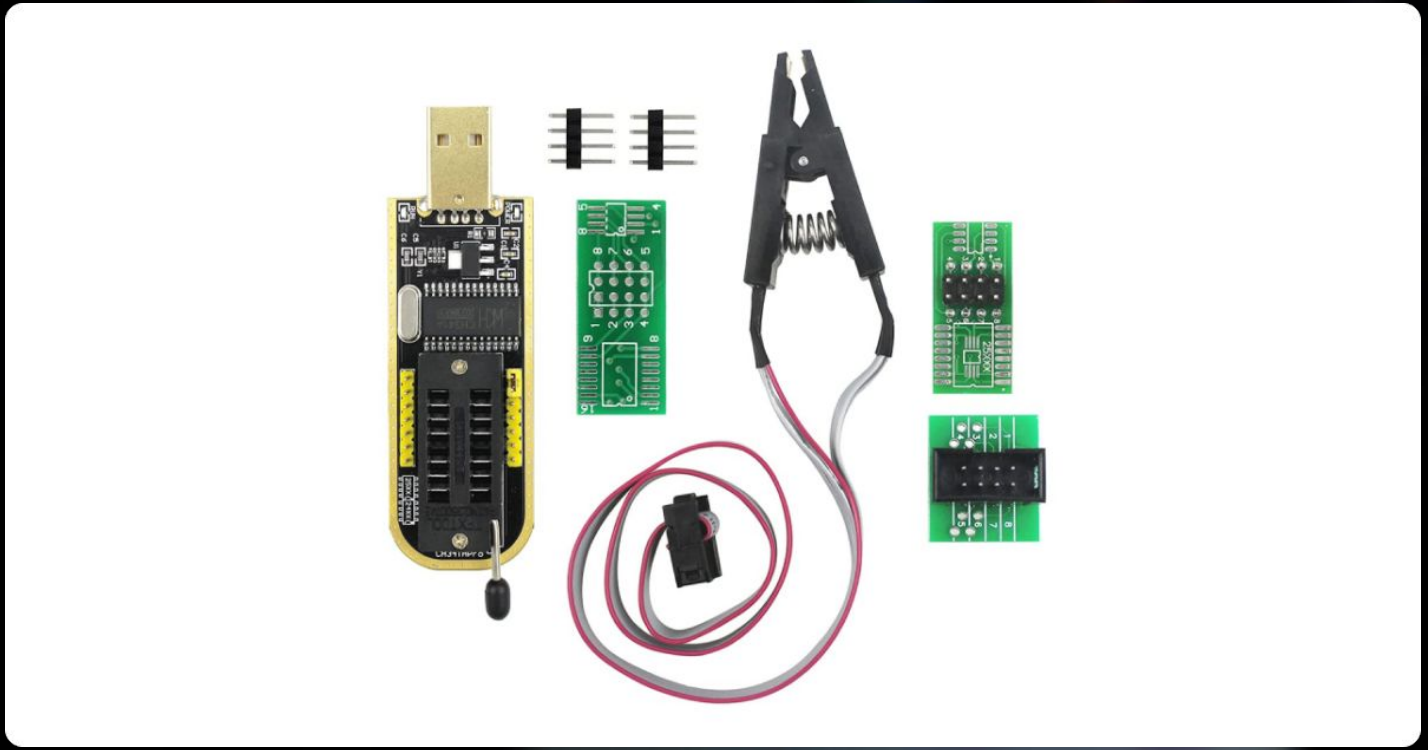


A Micro SMD Grabber’s fine tips and it’s delicate design make it **ideal for handling and probing small components during electronic testing and debugging.**

Multi-**Meter**

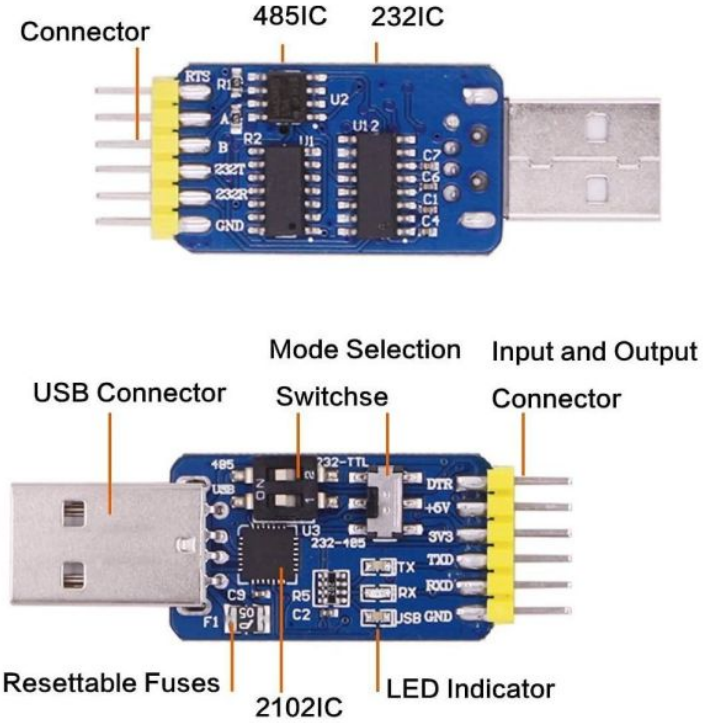
A multi-meter, just as the name suggests, is like a **Swiss army knife for measuring and testing circuits.**

SPI Flash **Programmer**



The SPI **(Serial Peripheral Interface)** Flash Programmer is a device that allows you to read and write data to the SPI flash memory chip on a PCB board

USB-**UART**



A USB-UART device plays a pivotal role in facilitating communication with the chip on a PCB board. **It acts as a conduit for transmitting and receiving data, rendering it an indispensable tool for hardware hacking enthusiasts.**

Logic **Analyzer**

Specialized electronic tool used for capturing and analyzing digital signals in electronic circuits that **helps engineers and technicians troubleshoot and debug complex digital systems** by providing a detailed view of signal behavior over time.

Hot Air **Rework Station**

Versatile tool used in electronics **soldering and desoldering tasks**. It produces a controlled stream of hot air to melt and remove or reflow solder on circuit boards and components, making it **essential for precise and efficient electronics repair and assembly.**

Anti Static Wrist Band

An anti-static wristband is an essential tool for protecting sensitive electronic components from electrostatic discharge (ESD) during assembly or repair. It safely dissipates static electricity from the wearer, ensuring a static-free environment for delicate electronics.

Flipper **Zero**

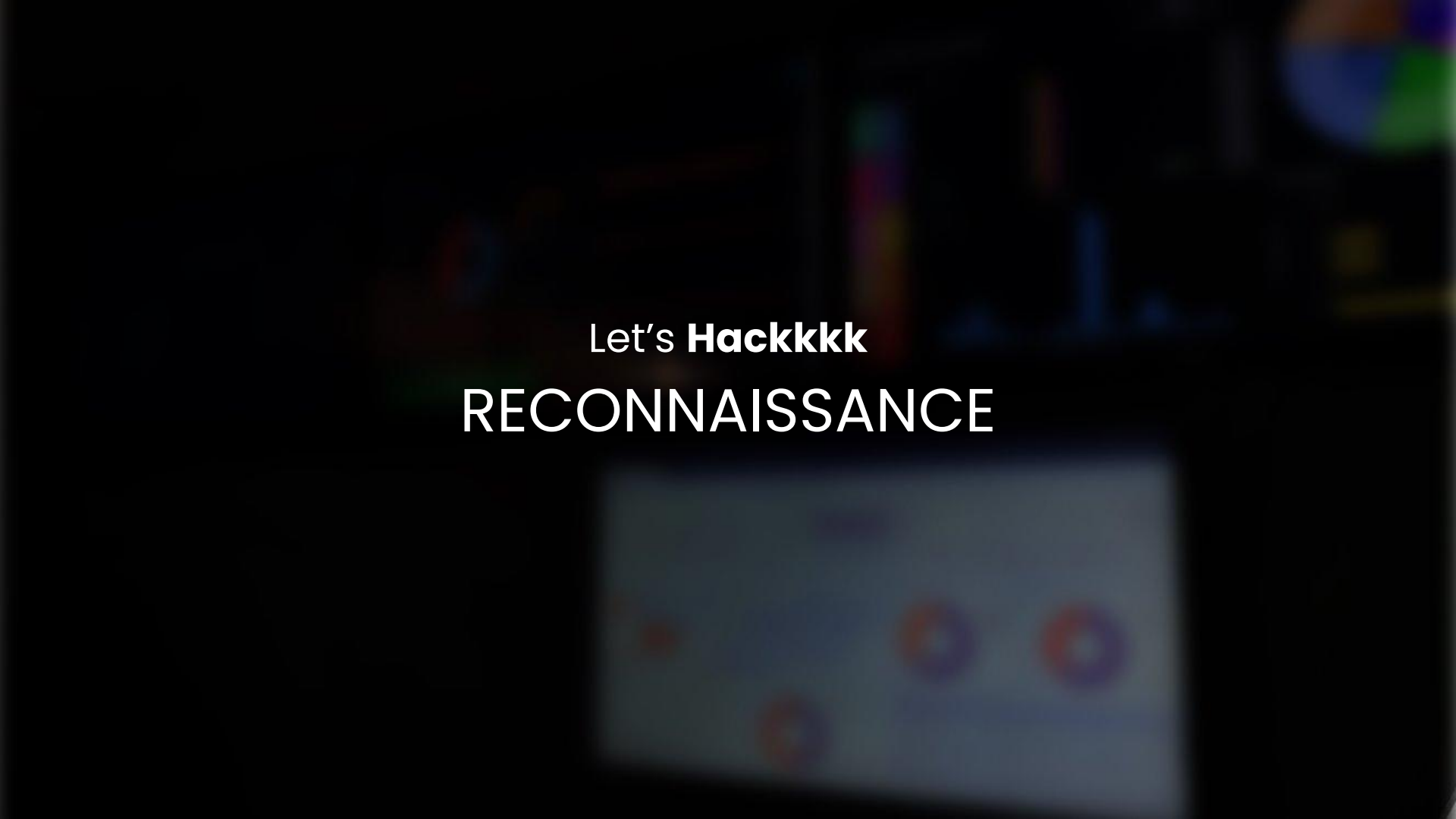
The **Flipper Zero** is a versatile device designed for security professionals specializing in **frequency and hardware security**. It serves as a multi-tool with diverse applications, boasting **features such as a built-in USB-UART and SPI Flash Programmer, among others.**

Other **Tools**

While we have already covered several essential tools for hardware hacking, there are still additional tools that may be needed in certain situations.

- **Screwdrivers**
- **Razors**
- **Tape**
- **Wire Cutter**
- **3rd Hand**

etc...


Let’s **Hackkkk**
RECONNAISSANCE

Basic Things That We Should Look At

Purchasing hardware for exploitation without assessing its susceptibility to vulnerabilities is not advisable. Therefore, you can explore the following sources for reconnaissance purposes:

- Firmware **Reverse Engineering**
- Official **Product Documentation**
- **FCC ID** Database

Federal Communications Commission **(FCC) ID**

FCC ID stands for **Federal Communications Commission Identification** which is a special alphanumeric code given to electronic devices and products that emit **radio frequency (RF)** signals and Its purpose is to identify and govern these devices within the **United States.**

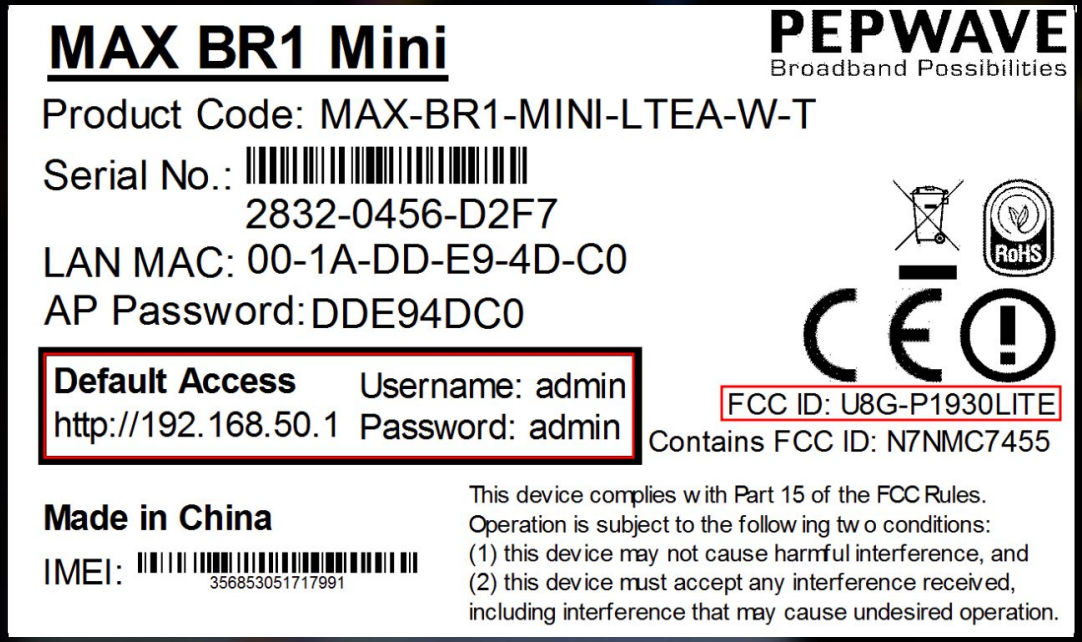
Federal Communications Commission (FCC) ID

US Government Official
https://www.fcc.gov/oet/ea/fccid

Easy to Search
https://fccid.io

Usages
https://gov.fccid.io/**{{FCCID}}**

Example
https://gov.fccid.io/**U8GP1930LITE**



Pismo Labs Technology P

fccid.io/U8G-P1930LITE

Online Tools | Research | Programming | Tutorials | Devlopment | AI | Movie and Series | vx-undergrou... | Awesome | Machine Lear...

FCC ID.io Blog Search

FCC ID U8G-P1930LITE

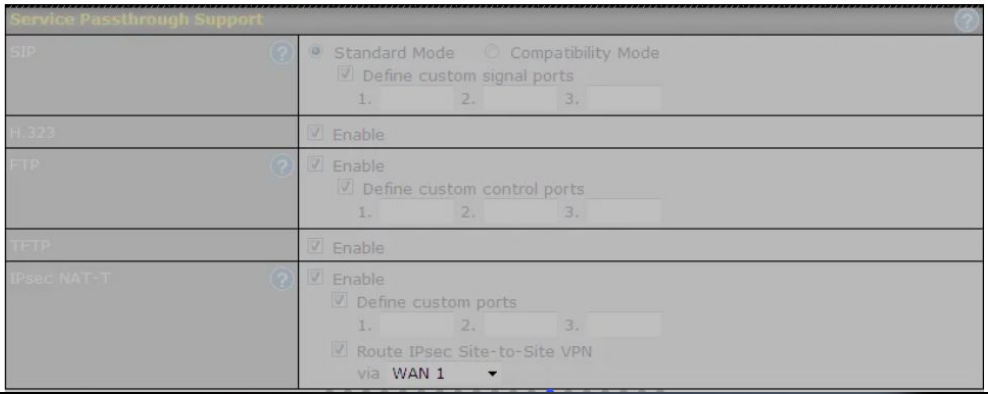
FCC ID U8G-P1930LITE Users-Manual, U8G P1930LITE, U8G-P1930LITE, U8G-P193OLITE, U8G-PI930LITE, U8G-P1930L1TE

Pismo Labs Technology Limited Peplink/ Pepwave/ Pismo Labs wireless product **P1930LITE**

FCC ID› / Pismo Labs Technology Limited› / P1930LITE

An FCC ID is the product ID assigned by the FCC to identify wireless products in the market. The FCC chooses 3 or 5 character "Grantee" codes to identify the business that created the product. For example, the grantee code for **FCC ID: U8G-P1930LITE** is **U8G**. The remaining characters of the FCC ID, **-P1930LITE**, are often associated with the product model, but they can be random. These letters are chosen by the applicant. In addition to the application, the FCC also publishes ***internal images, external images, user manuals, and test results*** for wireless devices. They can be under the "exhibits" tab below.

Purchase on Amazon: Peplink/ Pepwave/ Pismo Labs wireless product



App #	Purpose	Date	Unique ID
1	Original Equipment	2016-06-15	OmhAzRGiQRxnGs/ZQS9wWQ==

Operating Frequencies

Frequency Range	Power Output	Rule Parts	Line Entry
2.412-2.462 GHz	76.6 mW	15C	1

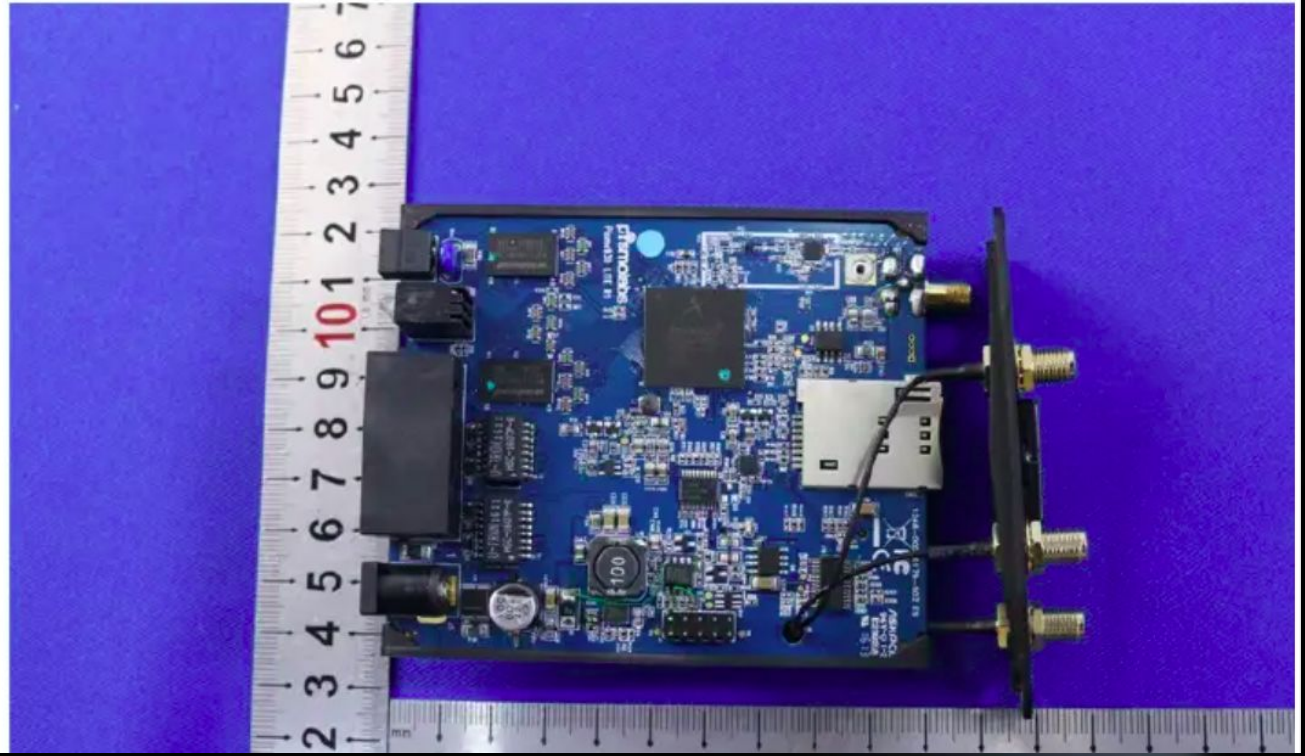
Exhibits

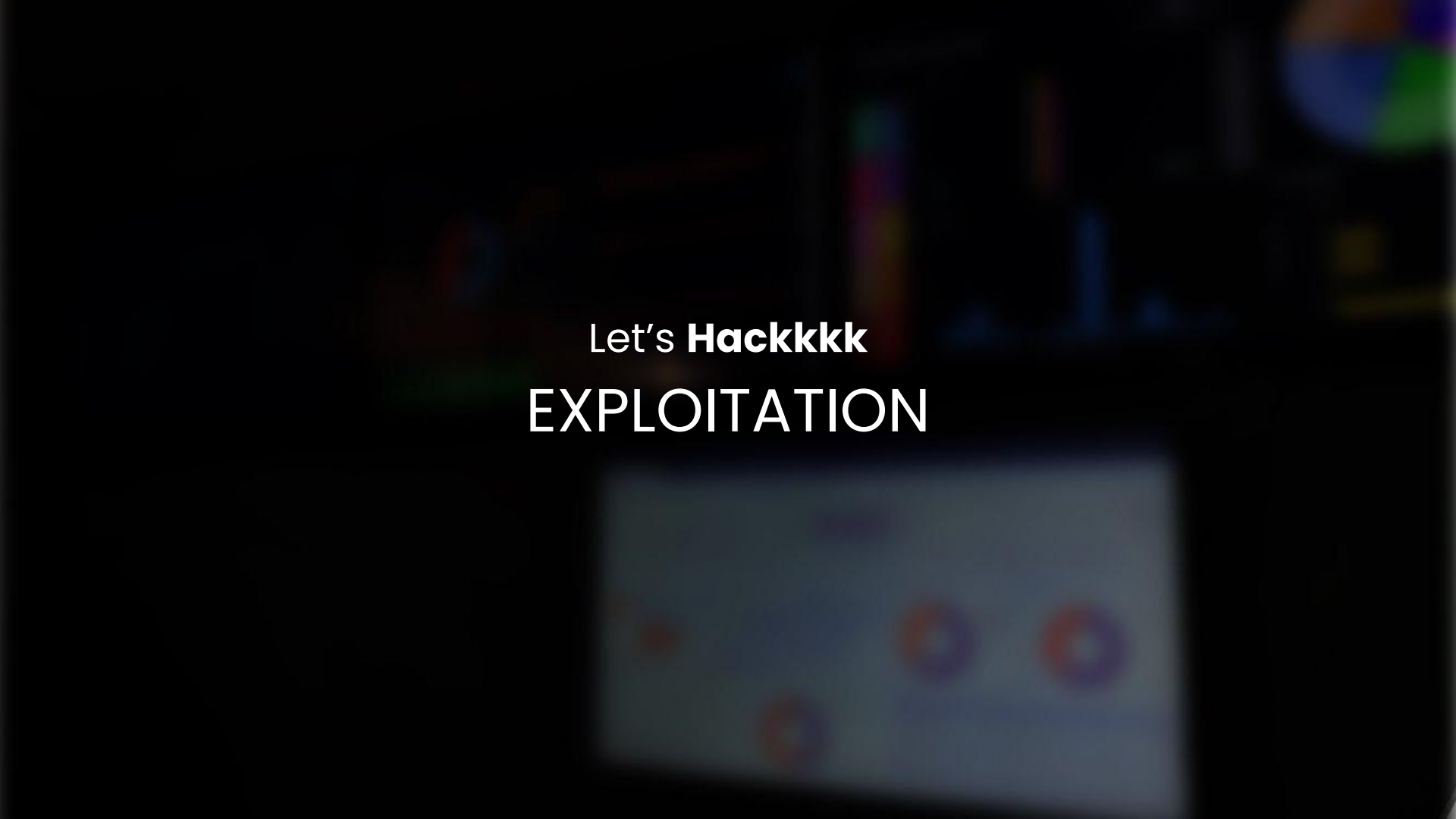
All

Document	Type	Submitted Available
User manual_U8G-P1930LITE_rev2	Users Manual Adobe Acrobat PDF (5303 kB)	2016-06-15 2016-06-15
setup photos	Test Setup Photos Adobe Acrobat PDF (317 kB)	2016-06-15 2016-06-15
HKES160500087702	Test Report Adobe Acrobat PDF (2986 kB)	2016-06-15 2016-06-15
HKES160500087703-revised	RF Exposure Info Adobe Acrobat PDF (193 kB)	2016-06-15 2016-06-15
LTE MPE	RF Exposure Info Adobe Acrobat PDF (457 kB)	2016-06-15 2016-06-15
Label-U8G-P1930LITE Contains FCC ID N7NMC7455	ID Label/Location Info Adobe Acrobat PDF (164 kB)	2016-06-15 2016-06-15
Label_U8G-P1930LITE contans FCC ID N7NMC7355	ID Label/Location Info Adobe Acrobat PDF (159 kB)	2016-06-15 2016-06-15
Label location_U8G-P1930LITE	ID Label/Location Info Adobe Acrobat PDF (18 kB)	2016-06-15 2016-06-15
Internal Photos -revised	Internal Photos Adobe Acrobat PDF (4024 kB)	2016-06-15 2016-06-15
External Photos	External Photos Adobe Acrobat PDF (1261 kB)	2016-06-15 2016-06-15
Statement of model declaration_BR1 mini_03-06-2016-revised	Cover Letter(s) Adobe Acrobat PDF (167 kB)	2016-06-15 2016-06-15
Confidentiality letter_U8G-P1930LITE	Cover Letter(s) Adobe Acrobat PDF (1073 kB)	2016-06-15 2016-06-15
Channel statement_U8G-P1930LITE	Cover Letter(s) Adobe Acrobat PDF (474 kB)	2016-06-15 2016-06-15

1 Photographs - EUT Constructional Details

Test model No.: MAX BR1 mini




Let’s **Hackkkk**
EXPLOITATION

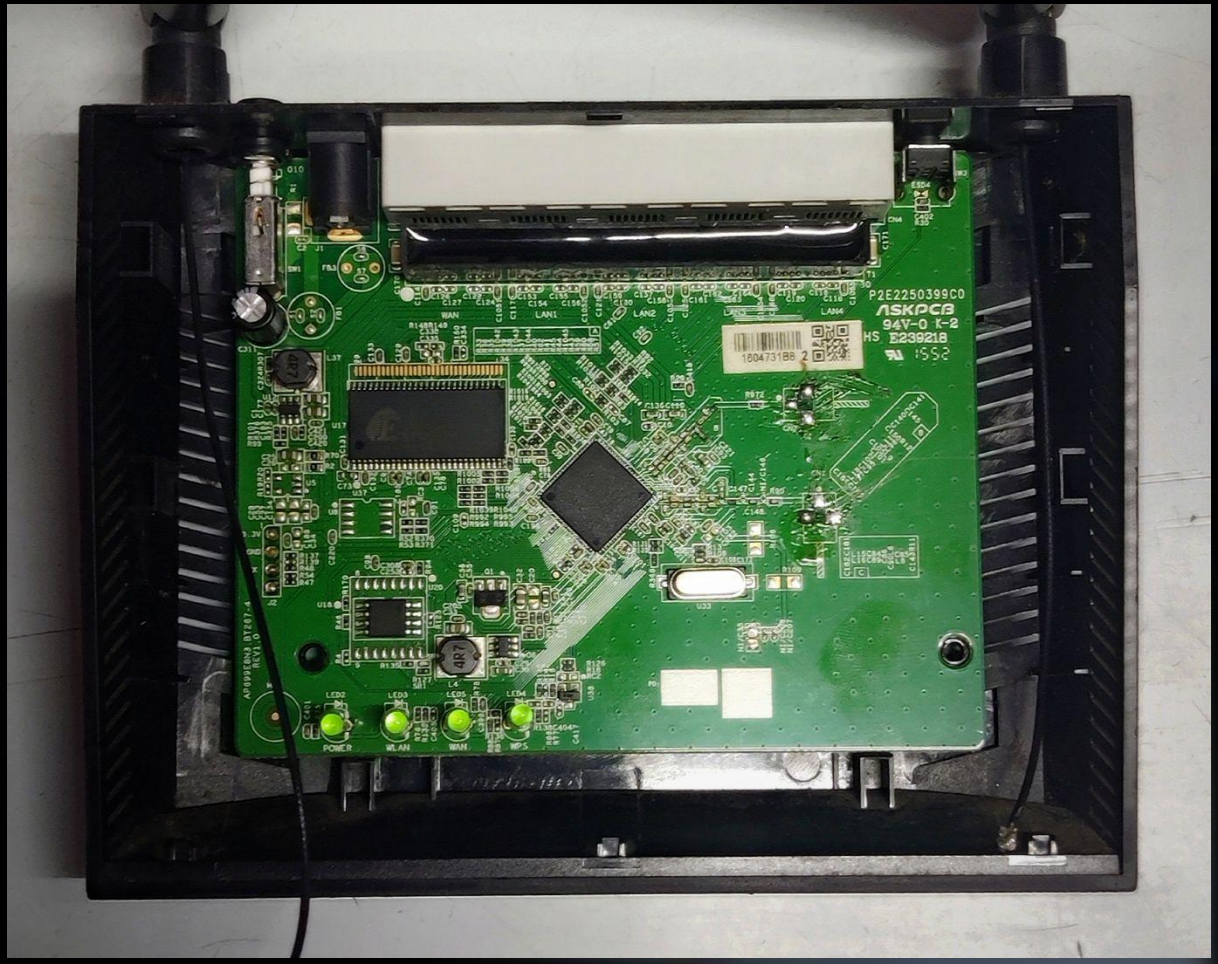
Hunting the **debugging interface**

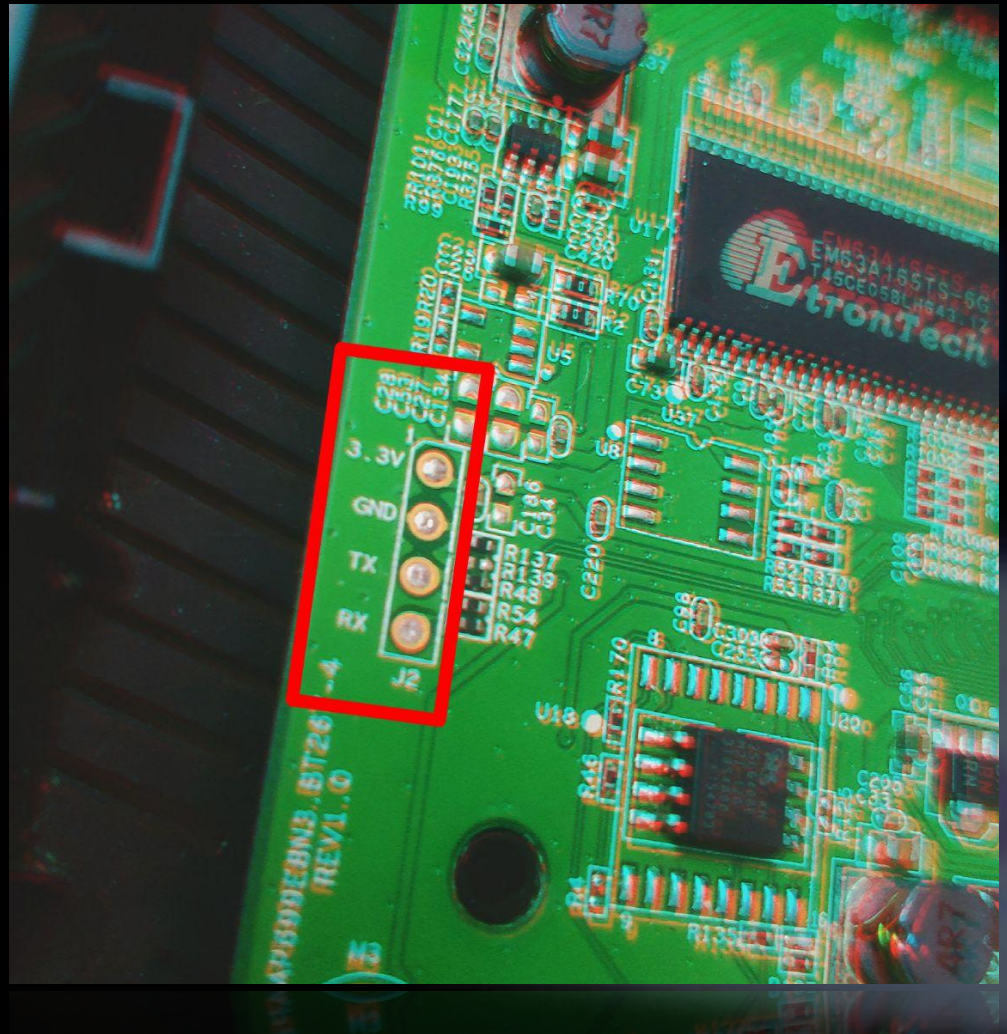
UART, JTAG and similar interfaces are found on circuit boards to debugging, programming, testing, and maintenance. Developers use them to monitor systems, program microcontrollers, ensure quality, and simplify upkeep. These interfaces enhance device flexibility and functionality but require security to prevent unauthorized access.

- **UART - Universal Asynchronous Receiver & Transmitter (Full Duplex)**
- jTAG - Joint Test Action Group
- SPI - Serial Peripheral Interface
- I2C - Inter-Integrated Circuit

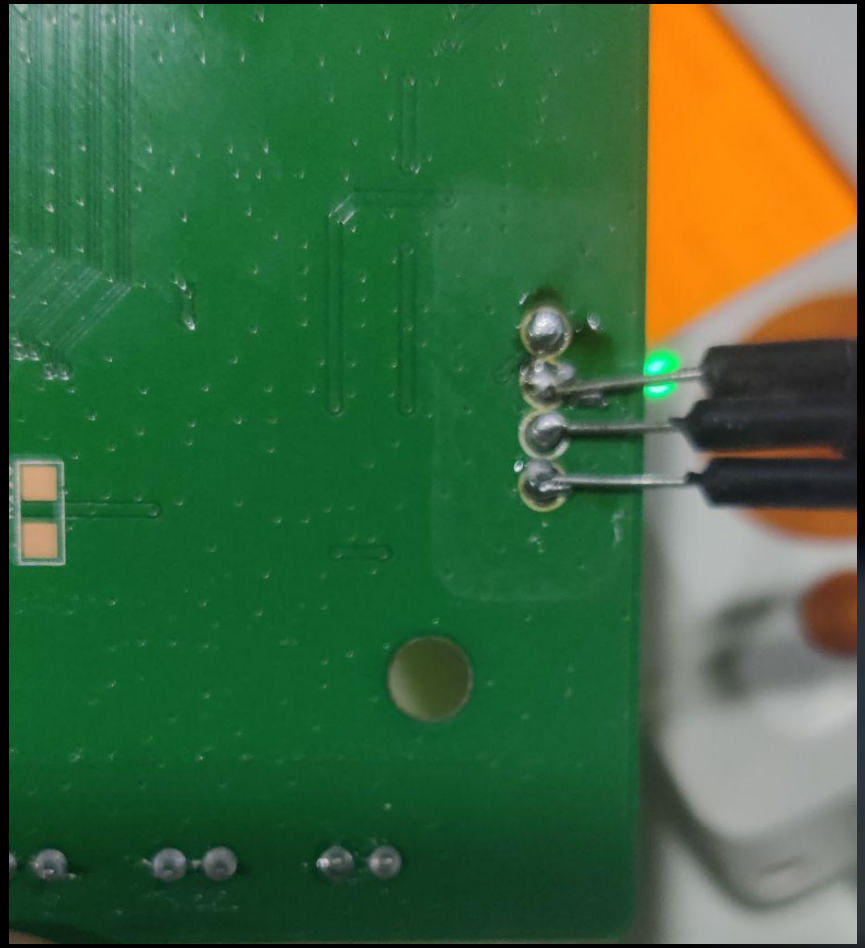
etc

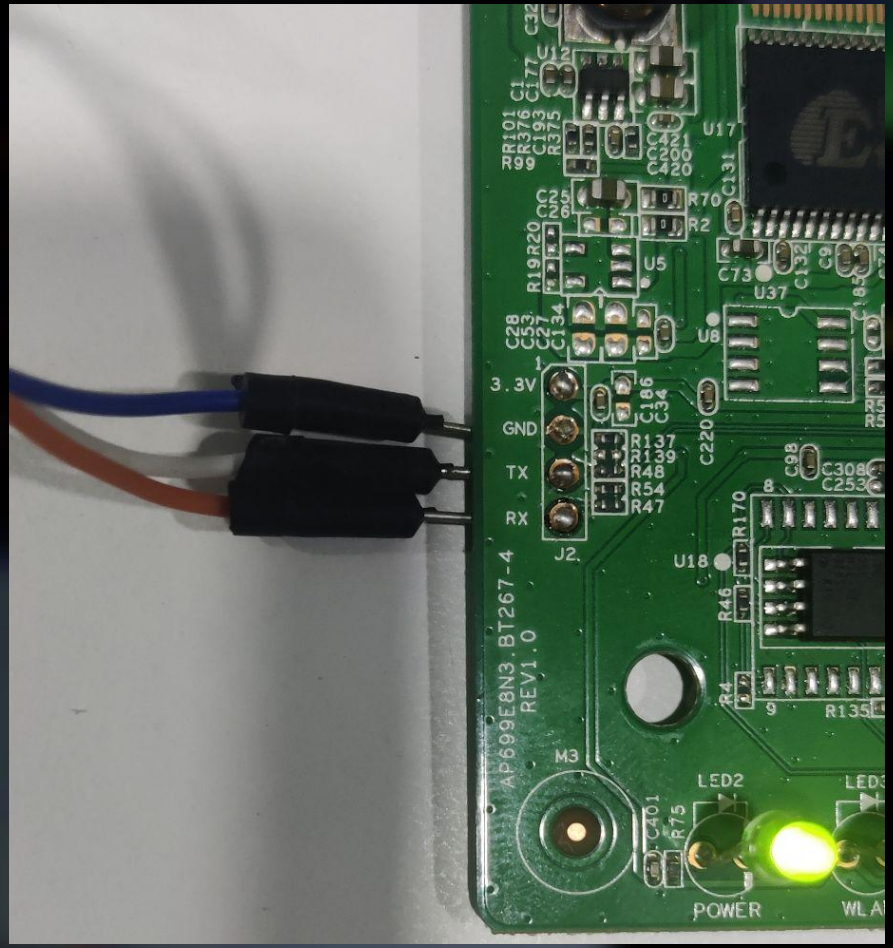

Let’s **Hackkkk**
Into The Shell


P2E225039900
ASKPCB
94V-0 K-2
E239218
POWER
WLAN
WAN
WPS

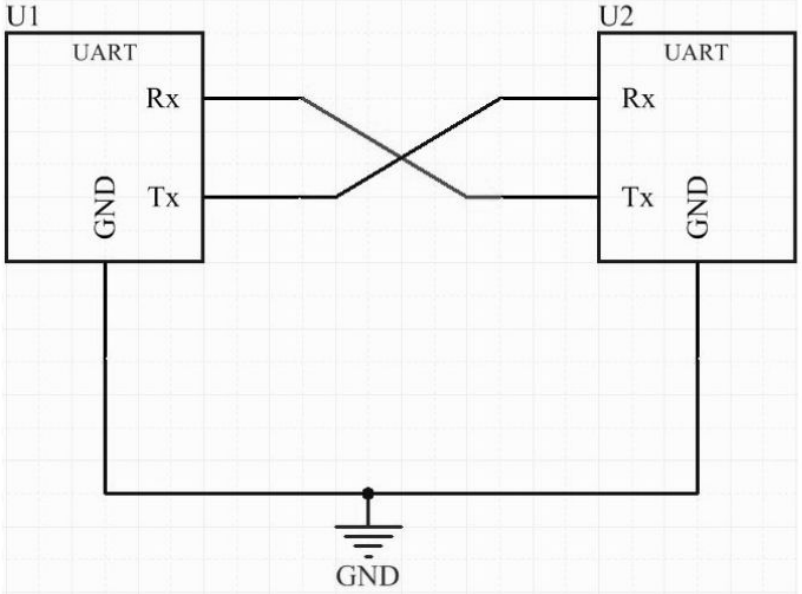

3.3V
GND
TX
RX
J2
EM63A165TS-6G
EtronTech


P2E225039900
ASKPCB
94V-0 L-2
E239218
POWER
WLAN
WAN

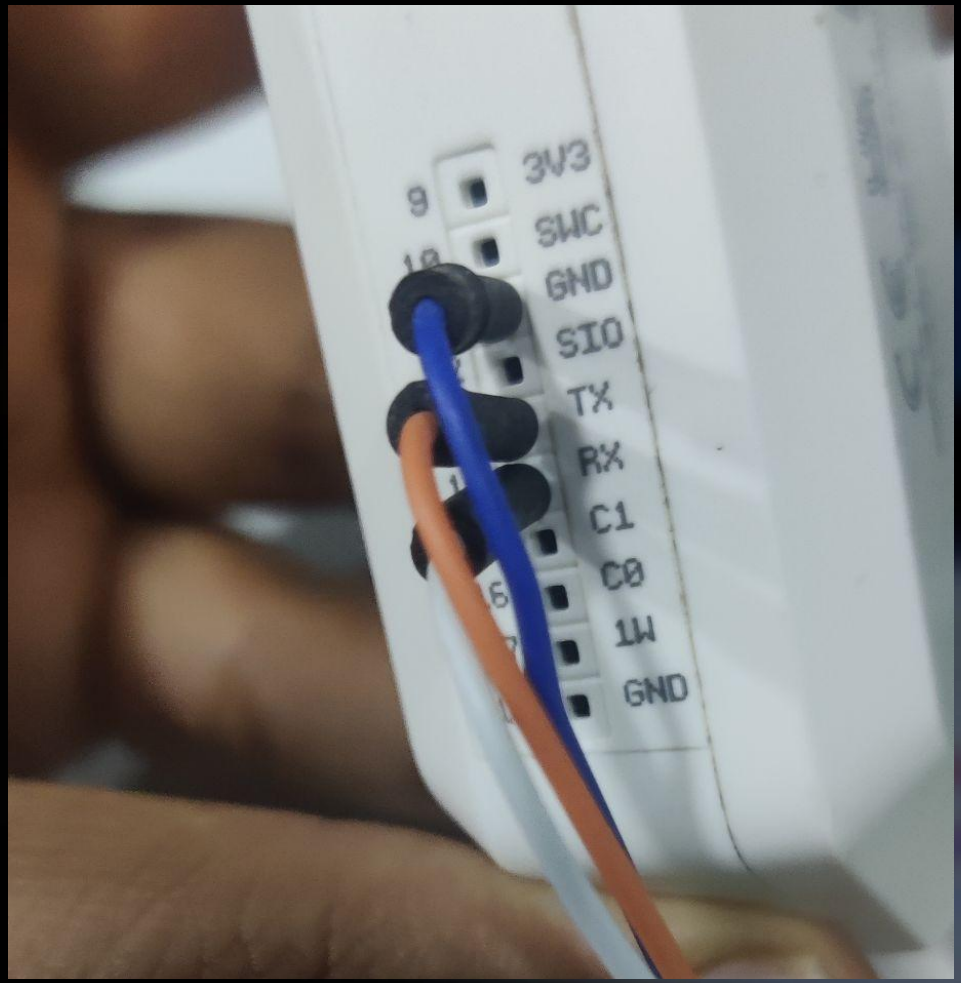


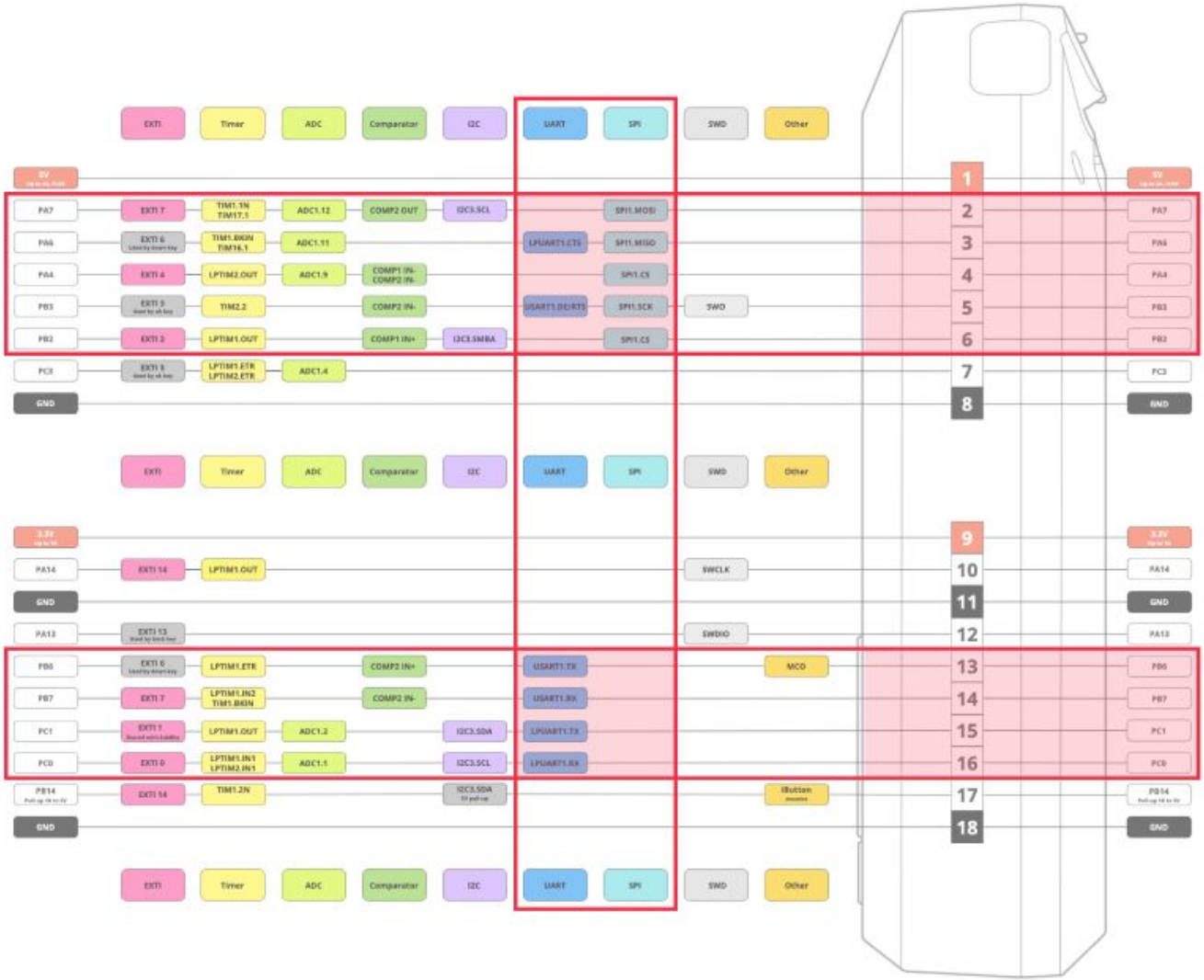

3.3V
GND
TX
RX
J2
U18
M3
LED2
POWER
AP699E8N3.BT267-4
REV1.0

Connection




Shift
PgDn


9
3V3
SWC
GND
SIO
TX
RX
C1
C0
1W
GND

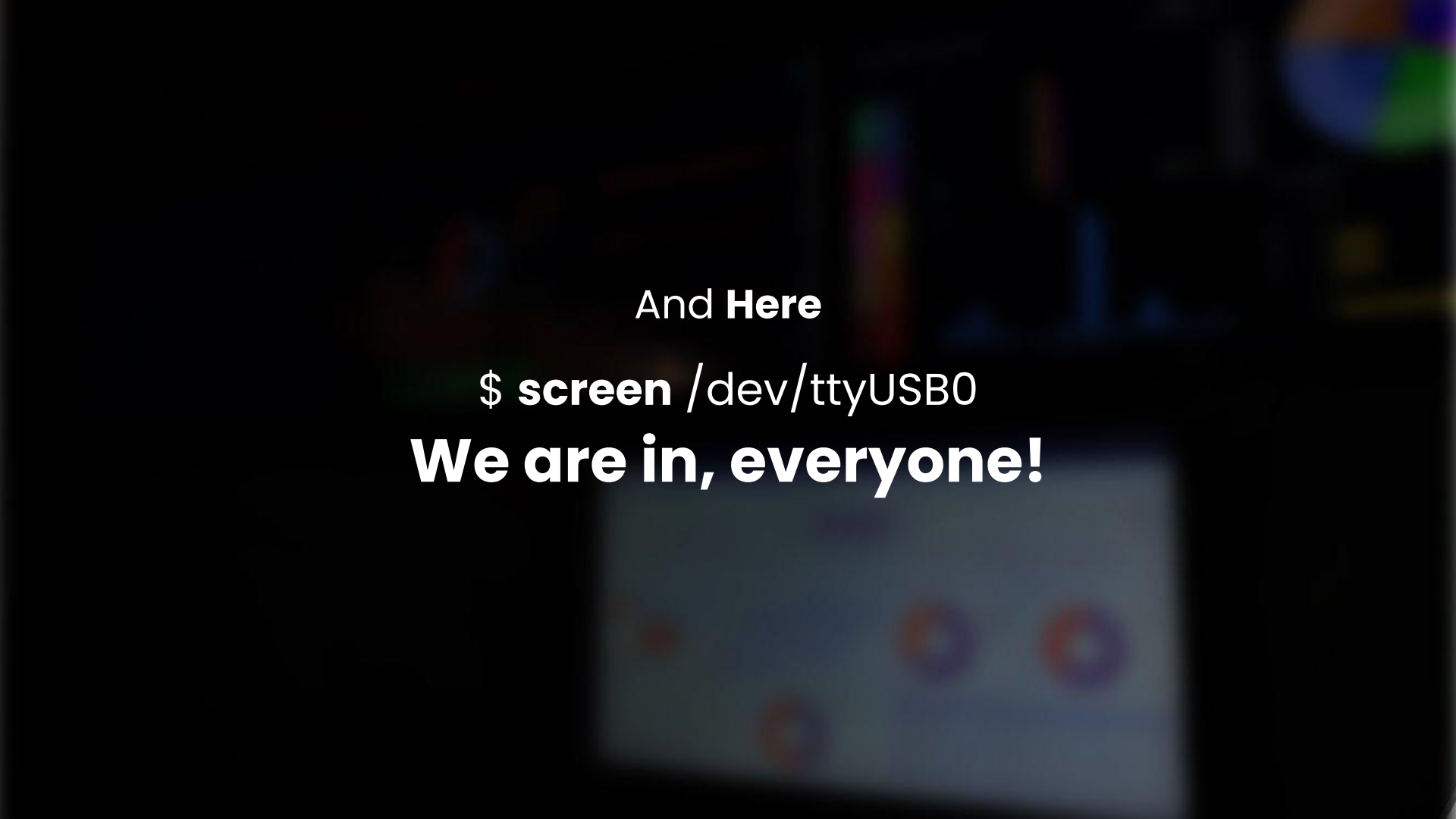

EXTI
Timer
ADC
Comparator
I2C
UART
SPI
SWD
Other
5V
PA7
EXTI 7
TIM1.1N
TIM17.1
ADC1.12
COMP2 OUT
I2C3.SCL
SPI1.MOSI
PA6
EXTI 6
TIM1.BKIN
TIM16.1
ADC1.11
LPUART1.CTS
SPI1.MISO
PA4
EXTI 4
LPTIM2.OUT
ADC1.9
COMP1 IN-
COMP2 IN-
SPI1.CS
PB3
EXTI 3
TIM2.2
COMP2 IN-
USART1.DE/RTS
SPI1.SCK
SWD
PB2
EXTI 2
LPTIM1.OUT
COMP1 IN+
I2C3.SMBA
SPI1.CS
PC3
EXTI 3
LPTIM1.ETR
LPTIM2.ETR
ADC1.4
GND
1
2
3
4
5
6
7
8
3.3V
PA14
EXTI 14
LPTIM1.OUT
SWCLK
9
10
11
12
PA13
EXTI 13
SWDIO
PB6
EXTI 6
LPTIM1.ETR
COMP2 IN+
USART1.TX
MCO
PB7
EXTI 7
LPTIM1.IN2
TIM1.BKIN
COMP2 IN-
USART1.RX
PC1
EXTI 1
LPTIM1.OUT
ADC1.2
I2C3.SDA
LPUART1.TX
PC0
EXTI 0
LPTIM1.IN1
LPTIM2.IN1
ADC1.1
I2C3.SCL
LPUART1.RX
13
14
15
16
PB14
EXTI 14
TIM1.2N
I2C3.SDA
IButton
17
18


USB Serial COM1PORT:0
TX:Pin 13
0 B
RX:Pin 14
196 B
Config
Baud:115200
FLIPPER

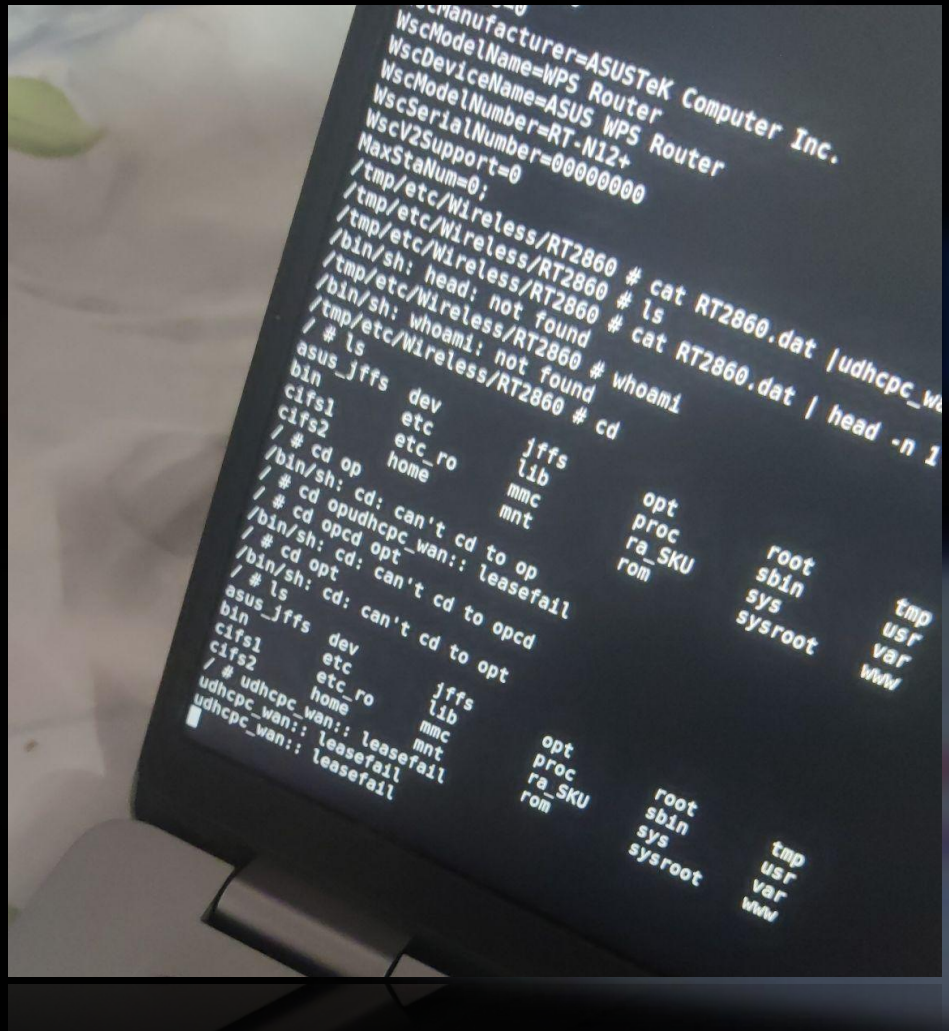

USB-UART Bridge
GP10 Manual Control
5V on GP10
OFF

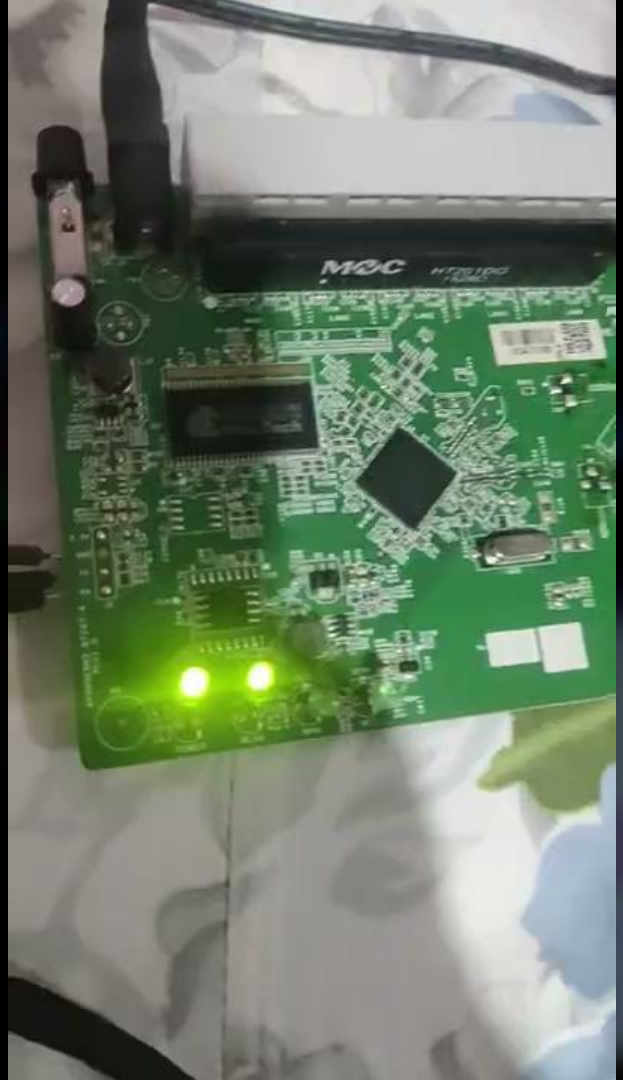

USB Serial
COM PORT:0
TX: Pin 13
0 B.
RX: Pin 14
268 B.
Config
Baud: 115200
FLIPPER


ls
/bin/sh: syntax error: unex
pected ")"
/ # udhcpc_wan:: leasefail

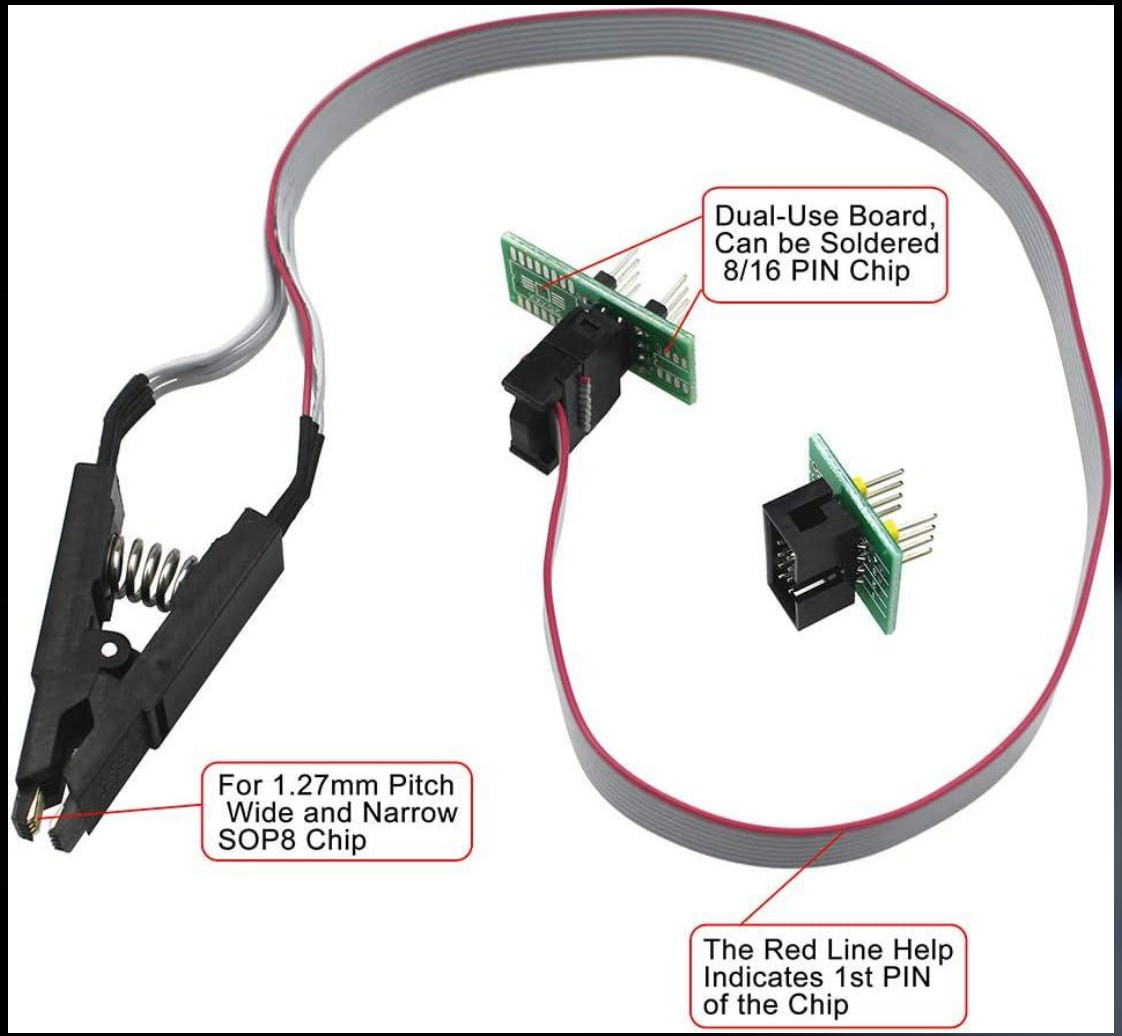

And **Here**
$ **screen** /dev/ttyUSB0
We are in, everyone!

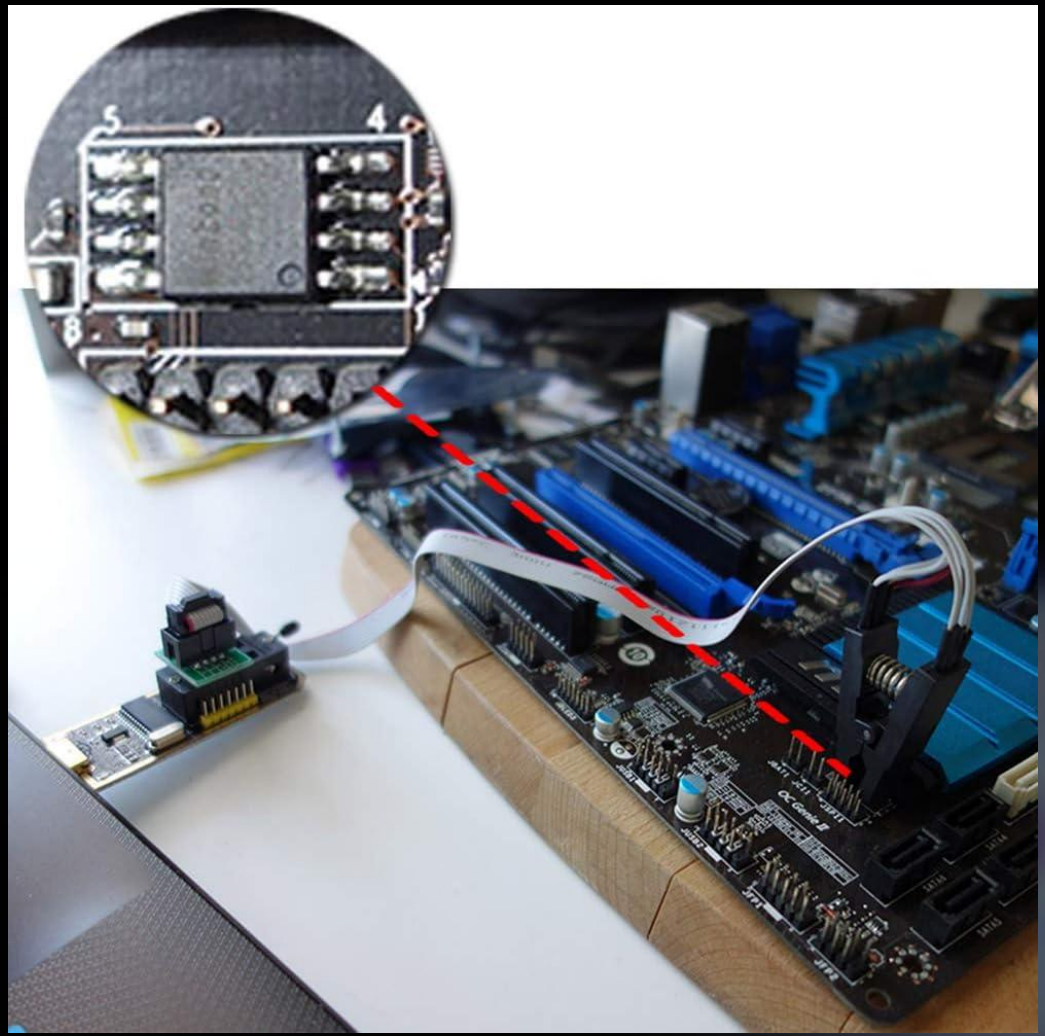
```
# Enable direct rule
/ #


U-Boot 1.1.3 (Apr  3 2014 - 17:08:04)

ASUS PRODUCT bootloader version: 1.0.0.0
Board: Ralink APSoC DRAM:  32 MB
ASUS ASUS PRODUCT gpio init : reset pin
enable ephy clock...done. rf reg 29 = 5
SSC enabled. swing=5000, upperbound=0
Ralink SPI flash driver, SPI clock: 29MHz
spi device id: c2 20 17 c2 20 (2017c220)
find flash: MX25L6405D
raspi_read: from:30000 len:1000
Maximum malloc length: 1024 KBytes
mem_malloc_start/brk/end: 0x81eaf000/81eb1000/81fb0000
*** Warning - bad CRC, using default environment

==========================================
Ralink UBoot Version: 4.1.1.0
------------------------------------------
ASIC 7620_MP (Port5<->None)
DRAM component: 256 Mbits SDR
DRAM bus: 16 bit
Total memory: 32 MBytes
Flash component: SPI Flash
Date:Apr  3 2014  Time:17:08:04
==========================================
icache: sets:512, ways:4, linesz:32 ,total:65536
dcache: sets:256, ways:4, linesz:32 ,total:32768

Please choose the operation:
   0: Load System code then write to Flash via Serial.
   1: Load System code to SDRAM via TFTP.
   2: Load System code then write to Flash via TFTP.
   3: Boot System code via Flash (default).
   4: Entr boot command line interface.
   5: Load Boot Loader code to SDRAM via Serial.
   7: Load Boot Loader code then write to Flash via Serial.
   8: Load Boot Loader code to SDRAM via TFTP.
   9: Load Boot Loader code then write to Flash via TFTP.

3: System Boot System code via Flash.
raspi_read: from:4018a len:4

ASUS PRODUCT bootloader version: 1.0.0.0
raspi_read: from:40004 len:6
MAC Address: D0:17:C2:36:7E:F0
raspi_read: from:40004 len:6

## Checking 1st firmware at bc050000 ...
```


WscManufacturer=ASUSTeK Computer Inc.
WscModelName=WPS Router
WscDeviceName=ASUS WPS Router
WscModelNumber=RT-N12+
WscSerialNumber=00000000
WscV2Support=0
MaxStaNum=0;
/tmp/etc/Wireless/RT2860 # cat RT2860.dat |udhcpc_wa
/tmp/etc/Wireless/RT2860 # ls
/tmp/etc/Wireless/RT2860 # cat RT2860.dat | head -n 1
/bin/sh: head: not found
/tmp/etc/Wireless/RT2860 # whoami
/bin/sh: whoami: not found
/tmp/etc/Wireless/RT2860 # cd
/ # ls
asus_jffs
bin
cifs1
cifs2
dev
etc
etc_ro
home
jffs
lib
mmc
mnt
opt
proc
ra_SKU
rom
root
sbin
sys
sysroot
tmp
usr
var
www
/ # cd op
/bin/sh: cd: can't cd to op
/ # cd opudhcpc_wan::
/bin/sh: cd: can't cd to opcd
/ # cd opcd opt
/bin/sh: cd: can't cd to opcd
/ # cd opt
/bin/sh: cd: can't cd to opt
/ # ls
asus_jffs
bin
cifs1
cifs2
dev
etc
etc_ro
home
jffs
lib
mmc
mnt
opt
proc
ra_SKU
rom
root
sbin
sys
sysroot
tmp
usr
var
www
/ # udhcpc_wan::
udhcpc_wan:: leasefail
udhcpc_wan:: leasefail


MOC


What If
No SHELL?


Dual-Use Board,
Can be Soldered
8/16 PIN Chip
For 1.27mm Pitch
Wide and Narrow
SOP8 Chip
The Red Line Help
Indicates 1st PIN
of the Chip


5
4
8

There is a lot to cover, and it's not possible to do so in one session. However, I can provide you with

RESOURCES

Resources

1. **Samy Kamkar's Crash Course in How to Be a Hardware Hacker**
   https://www.youtube.com/watch?v=tlwXmNnXeSY

2. **How We Hacked a TP-Link Router and Took Home $55,000 in Pwn2Own**
   https://www.youtube.com/watch?v=zjafMP7EgEA

3. **Joe Grand**
   https://www.youtube.com/@JoeGrand/playlists

4. **Nirmal Dahal - #Nittam**
   https://nirmaldahal.com.np/series/hardware-hacking/

CryptoGen Nepal

Saycure.

Thank You!

For tolerating me until this time :D

/TheNittam.me

nirmaldahal.com.np

/TheNittam