Intro to Hardware Hacking

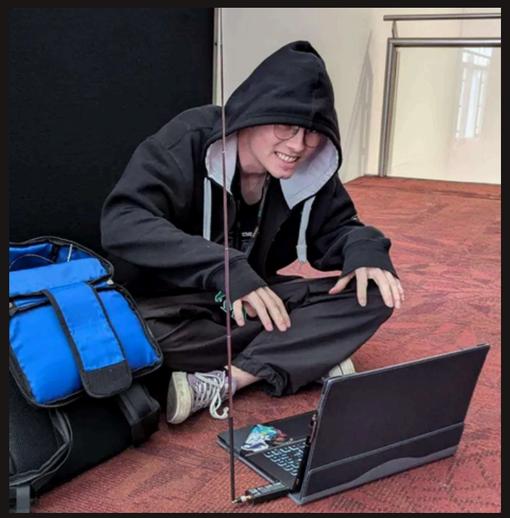
UTS:CSECCon IV

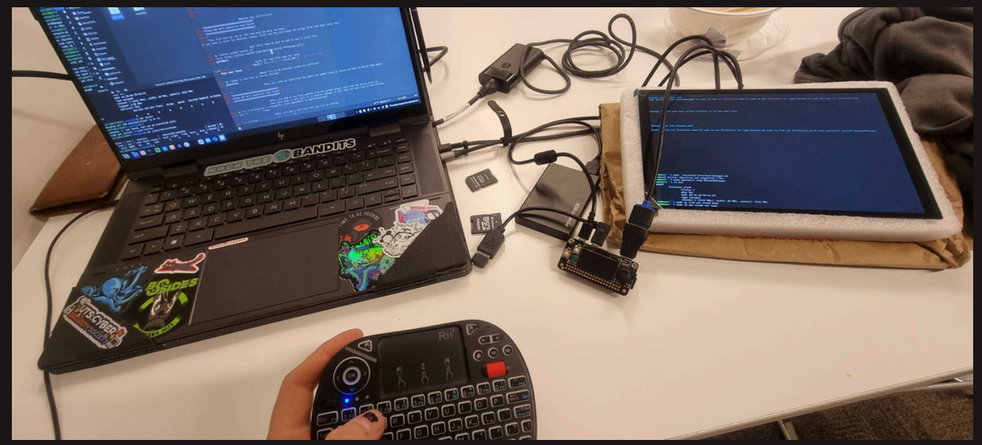
Acknowledgement of country

I would like to acknowledge the Gadigal people of the Eora Nation, upon whose ancestral lands our City campus now stands. I would also like to pay respect to the Elders both past, present, and future, acknowledging them as the traditional custodians of this land.

$whoami
> Aaron

- Final Semester - Bachelor of Cybersecurity
- Director of Ops - CSEC
- Passionate about electronics and embedded devices but still relatively new to hardware hacking itself
- Love teaching others
- Love figuring out how things work on a smaller scale
- Enjoy hiking, snowboarding, surfing and card/board/video games

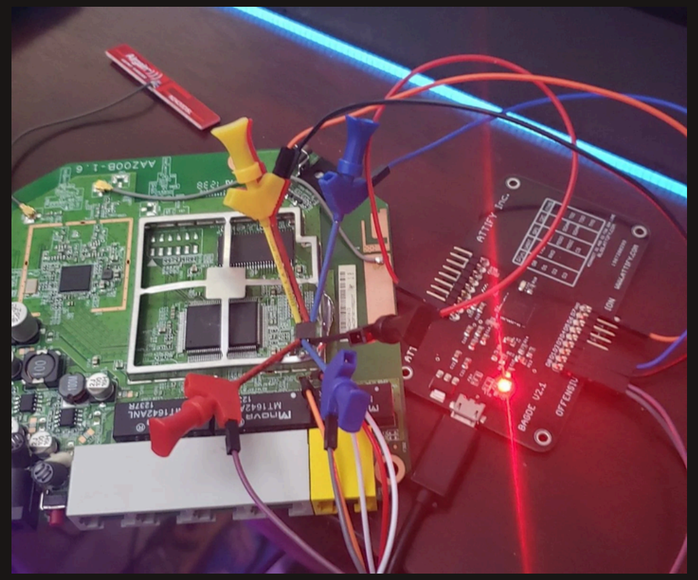




Hardware hacking

“The exploitation of a physical system”

- Hardware hacking is all about curiosity and a desire to see the inner workings of a device through unintended methods, whether it be for education, security research or both.



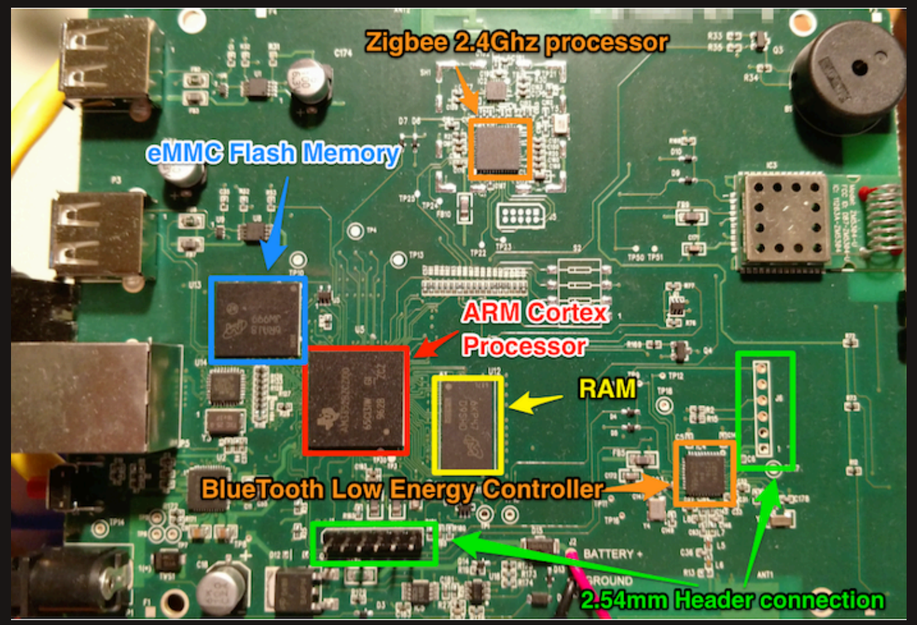
Common pathways

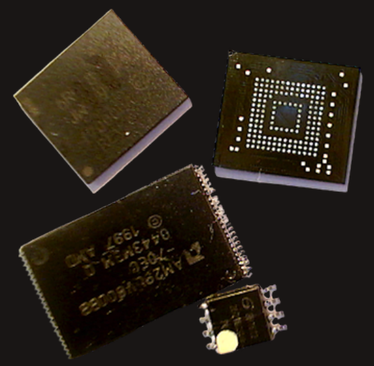
- **Accessing internal OS through unrestricted interfaces**
  - UART
  - JTAG
  - Ethernet/serial connections
  - other interfaces
- **Tricking or forcing a shell by altering the device**
  - increasing voltage
  - sending unexpected input at the right time
- **Memory/Firmware Dumping**
- **Firmware analysis and reverse engineering**

Disclaimer

Only hack what you have permission to hack.
The techniques shown in this workshop are for educational and security awareness purposes only.

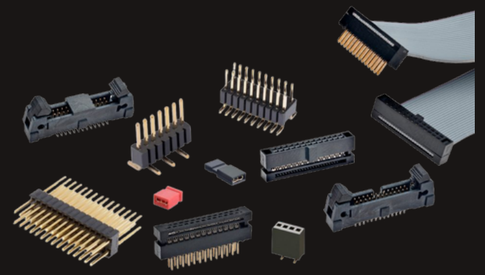
Key targeted components





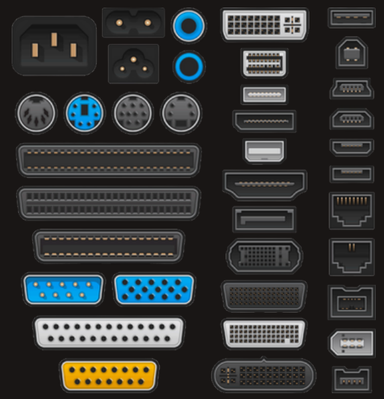
Integrated circuits (IC)

- Contained complicated circuits
- Used for complicated functions like processing, memory, logic and more



PCB Connectors

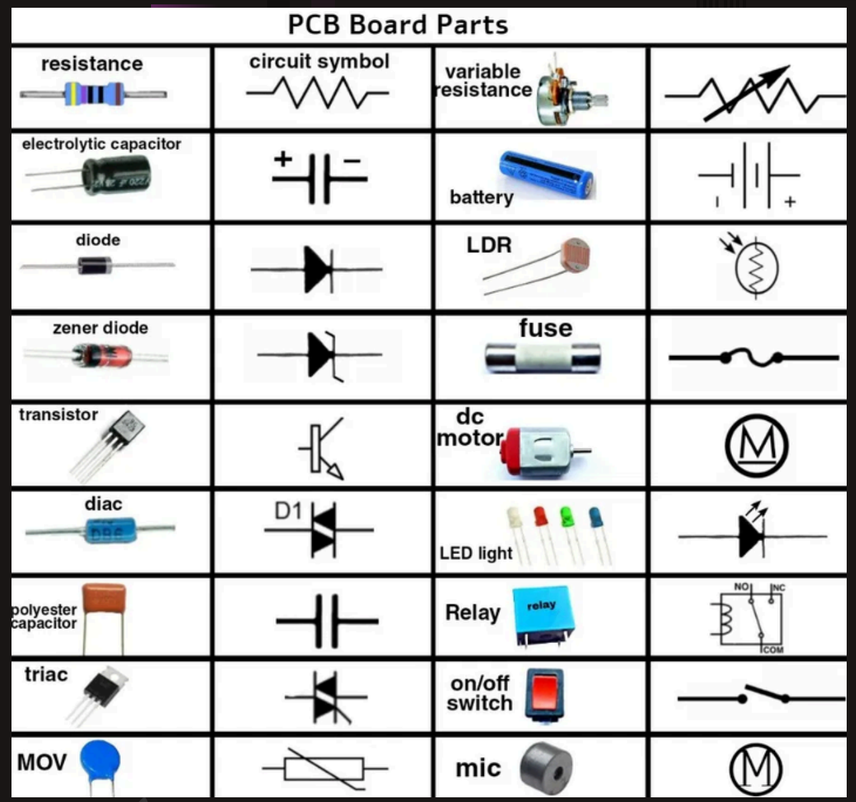
- Communication interfaces
- Used for debugging, programming and data transfer



Ports and connections

- Communication interfaces
- Used for external device communication

“passive” components



“Passive” refers to the fact that these components do not require external power to function, they simply effect any electricity that passes through them. Some of these will still dissipate power, like resistors, while some others are considered “lossless” like capacitors.

Resistors - control flow and voltage of electricity in a circuit

Capacitors - Very flexible in use. Filters signal and stabilise voltage.

Diode - Control current direction

Transistor - Electrically controlled switch. used for logic

Inductors - current stabling, stores energy in a magnetic field

Transformers and regulators - change voltage

Rectifier - Not technically one component but a very common circuit. Changes AC to DC

Fuse - Protects electrical equiptment

Common tools

Our toolkit for today:

Multimeter

- Used for finding connected surfaces like ground
- Check if components have been damaged
- Check voltage of pins
- Check for live data on connectors

Screwdrivers

- Used for disassembly/reassembly
- Essential but inexpensive

TTY Serial adapters

- Used for communication through onboard interfaces

Soldering Iron

- Used for removal or installation of PCB components
- Used to solder interface headers or jumper cables
- Dangerous if misused

EEPROM Programmer

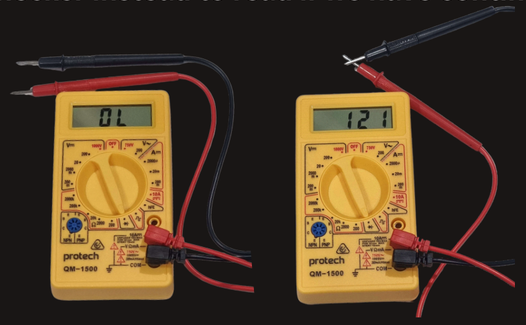
- Used for reading and programming EEPROM Memory ICs
- Can be used to dump memory
- More advanced

Todays Techniques

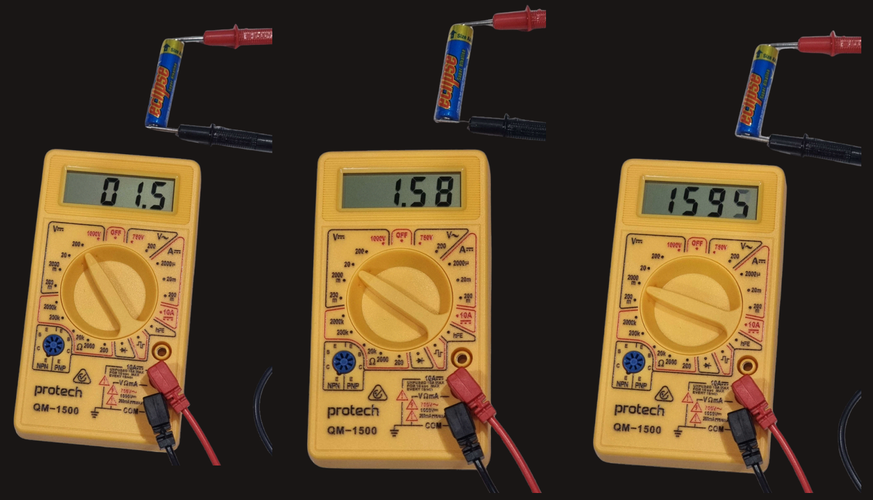
The (humble) Multimeter

(lore accurate multimeter strength)

- Continuity will tell us if two surfaces are connected, usually with an audible noise which these meters lack unfortunately
- We can use the diode checker instead to read if we have continuity as shown below



- These meters are known as manual ranging, meaning we need to select what range to read ourselves
- Not a major deal for our case, but when you have to measure several resistors without knowing the range, you'll feel the difference



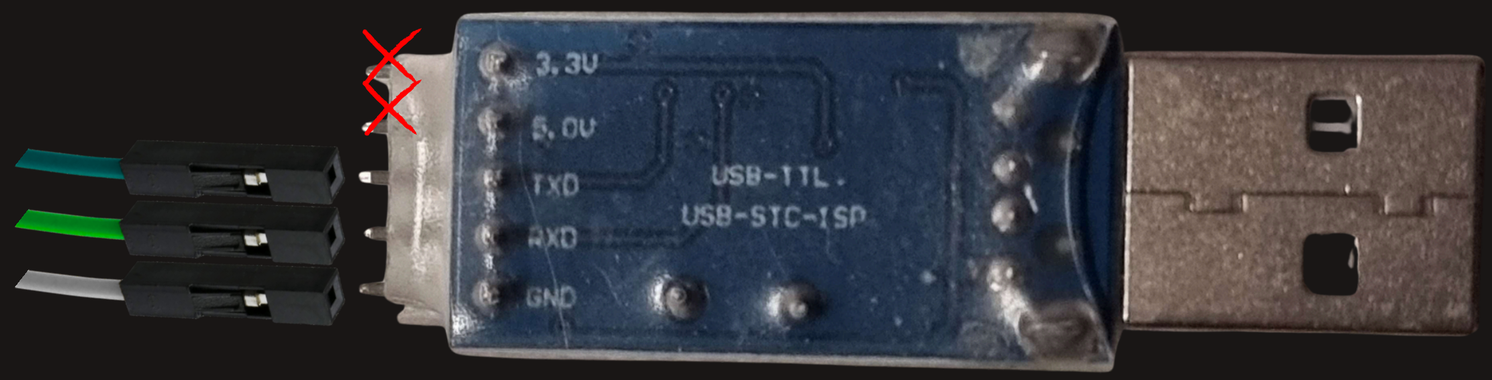
Manually ranging the 25th stray resistor that still isn't the value you need. (its been 2 hours)

Todays Techniques

Dubious USB TTY Serial adapter

- While cheap and outdated in terms of driver support, these will still allow us to connect to open interfaces that we might find on the board.
- Lack of driver support on windows means that we will need to use this adapter in Linux, where the drivers are part of the kernel.
- When interfacing with UART, we only want to connect the transmit (TX), receive (RX) and ground (GND) pins. Powering the CPU through UART can be damaging if done wrong.

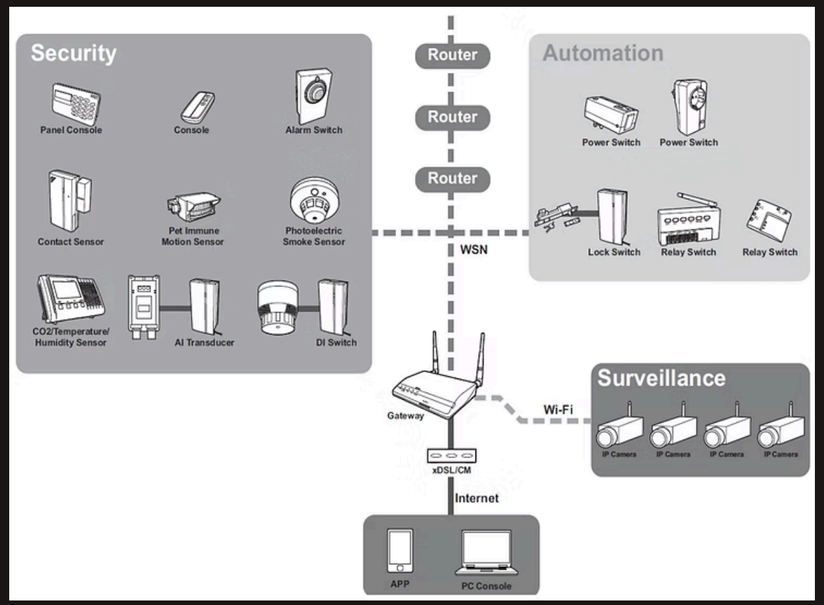
```
+-----------------------------------------------------------------------+
|                    Minicom Command Summary                            |
|                                                                       |
|         Commands can be called by CTRL-A <key>                        |
|                                                                       |
|               Main Functions                 Other Functions          |
|                                                                       |
| Dialing directory..D  run script (Go)....G | Clear Screen.......C     |
| Send files.........S  Receive files......R | cOnfigure Minicom..O     |
| comm Parameters....P  Add linefeed.......A | Suspend minicom....J     |
| Capture on/off.....L  Hangup.............H | eXit and reset.....X     |
| send break.........F  initialize Modem...M | Quit with no reset.Q     |
| Terminal settings..T  run Kermit.........K | Cursor key mode....I     |
| lineWrap on/off....W  local Echo on/off..E | Help screen........Z     |
| Paste file.........Y                       | scroll Back........B     |
|                                                                       |
|     Select function or press Enter for none.                          |
|                                                                       |
|           Written by Miquel van Smoorenburg 1991-1995                 |
|           Some additions by Jukka Lahtinen 1997-2000                  |
+-----------------------------------------------------------------------+
```




```
@          -VirtualBox: ~
File  Edit  View  Search  Terminal  Help
      @          -VirtualBox:~$ sudo picocom -b 115200 /dev/ttyUSB0
[sudo] password for          :
picocom v1.7

port is        : /dev/ttyUSB0
flowcontrol    : none
baudrate is    : 115200
parity is      : none
databits are   : 8
escape is      : C-a
local echo is  : no
noinit is      : no
noreset is     : no
nolock is      : no
send_cmd is    : sz -vv
receive_cmd is : rz -vv
imap is        :
omap is        :
emap is        : crcrlf,delbs,

Terminal ready

>>>
```

- We can then use a number of programs to read and write data through these devices, the two I will touch on today will be picocom and minicom, but many other programs exist.
- If we had an adapter that was supported on windows, PuTTY is the go to program.

Our target



Wireless Gateway Home Automation Controller

CAT.NO: LA5570 ★★½ 2.5 (2) Write a review

Overview

- Have you ever dreamt of controlling your homes lighting and appliances remotely?
- Have you ever wanted to turn your homes lights and appliances off and activate your security system with the touch of one button?
- Have you ever been on holidays and wanted to check on your home via video and be updated via Email of any events?

Security Alarm Function

- Real Time / Emergency Notifications Via email
- Home Automation Control Timers, scene control, remote operation
- Video Surveillance IP cameras support for live video monitoring
- Cost Effective True wireless connectivity
- Easy Installation DIY simple installation

Security

Fully secure your home with the LA-5570 Wireless Gateway. Featuring a wide range of wireless accessories such as door and window switches, PIR motion detectors and smoke detectors. You can easily build up a fully secure home environment to protect your home, family and property. Use either the wireless digital pulse input or the analogue input modules to connect a range of hard wired security sensors.

Our target

Unrestricted UART Access

CVE-2023-34724 Detail

MODIFIED

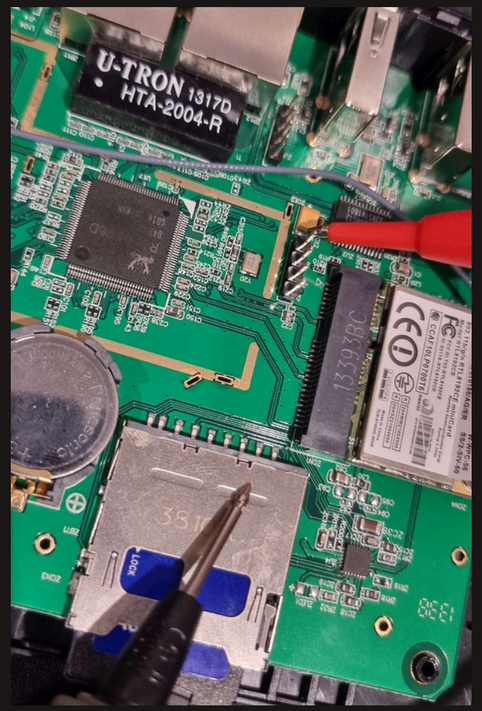
This CVE record has been updated after NVD enrichment efforts were completed. Enrichment data supplied by the NVD may require amendment due to these changes.

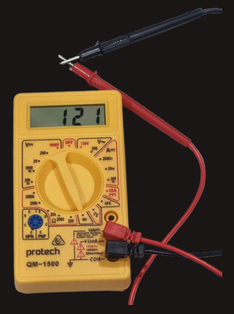
Description

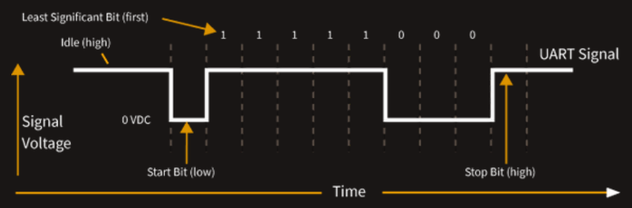
An issue was discovered in TECHView LA5570 Wireless Gateway 1.0.19_T53, allows physical attackers to gain escalated privileges via the UART interface.

Credit to Exploit Security here in Sydney for finding these discoveries

Finding UART





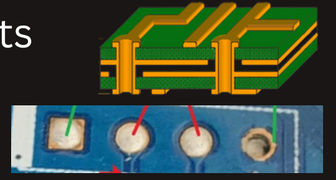


1 - Locate Interface pins

- These may sometimes be desoldered,

2 - Use continuity to test for ground pin

- Good ground planes include the metal brackets of connectors
- Also can be found by checking for a square based pin or “through hole” instead of circles



3 - Power device and use the multimeter to identify the voltage pin

- Also can be identified by an arrow on the board, though not all boards will identify this and should be tested.

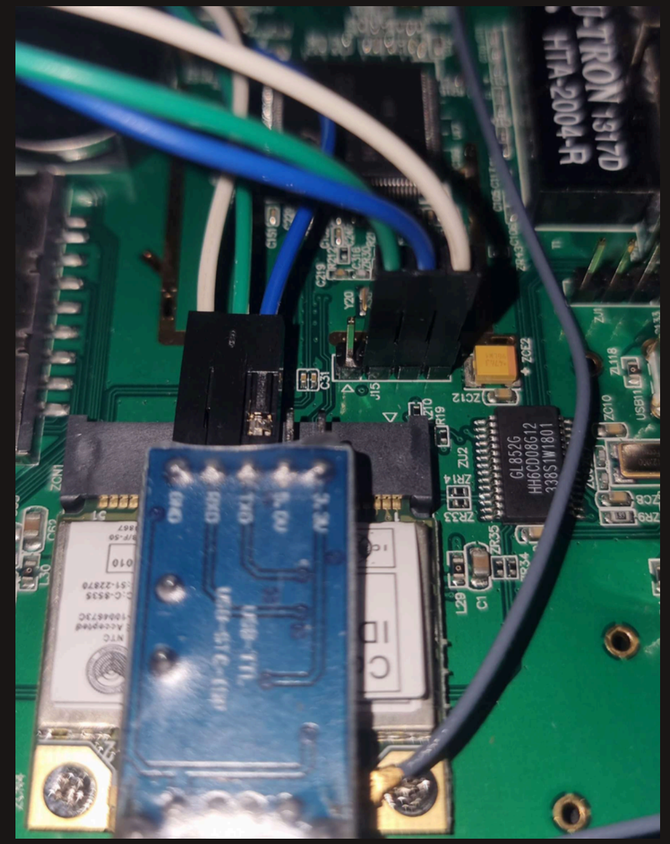
4 - Read remaining pins with the multimeter for any voltage fluctuation on boot. These will be the transmit pins

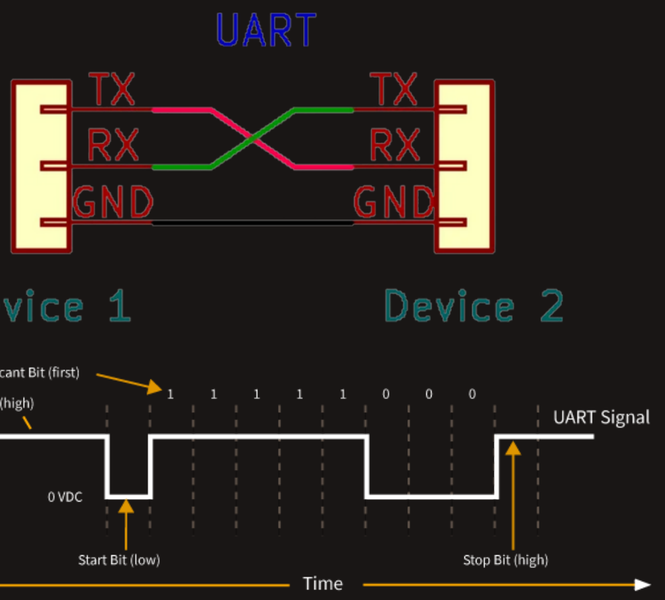
- In some devices, there will be no data transmission on boot but still may have an active UART interface.

Connecting to UART

Power down device

- DO NOT POWER DEVICE until you are sure that you have connected the pins correctly. We will also set up picocom/minicom before we power the device in order to see any boot messages





Cable placement

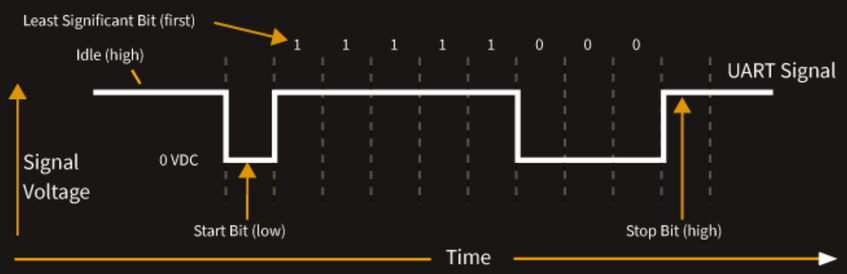
- The transmit and receive pins are each reversed, as each device wants to receive what the other device sends
- Connect common ground but DO NOT CONNECT VOLTAGE
- Double check pins are correct, especially the ground pin, as incorrect placement could result in a short and potentially damage to the device or your adapter.

Reading UART

```
baud: ?          baud: 38400
databit ?        databit: 8
stop bit ?       stop bit: 1
parity ?         parity: none
flow control ?   flow control: none
```

To communicate via UART, we need to first set a couple of variables. These include:

- **the baud rate** - the rate of which the data is sent
- **the number of data bits** - how many bits of data before expecting a stop bit
- **the number of stop bits** - how many bits constitute a stop bit
- **the type of parity** - error checking bits added to data
- **and flow control** - can be used to help ensure correct comms between slower devices



Since we have no way of knowing what the board uses, its time for some trial and error!

- try with different parameters and see what works, for the sake of time, there is no parity or flow control and the stop bits are 1.
- Try to find the data bits and the baud rate!

Reading with Picocom

```
sudo picocom -r -l /dev/ttyUSB0

Ctl[a] + Ctl[h]
Ctl[a] + Ctl[u]/Ctl[d]
Ctl[a] + Ctl[i]/Ctl[j]
Ctl[a] + Ctl[f]
Ctl[a] + Ctl[y]
Ctl[a] + Ctl[v]
```

Reading with Minicom

```
sudo minicom -D /dev/usbttyUSB0

Ctl[a] + [z]
Ctl[a] + [z] + [p]
Ctl[a] + [z] + [o]
```

Driver debugging

```
lsusb
dmesg
ls /dev/ttyUSB*
sudo modprobe pl2303
sudo modprobe usbserial
```

Body

Vulnerability No. 1 - Unrestricted root shell on UART

This alone is bad, but what can we find from here?

Bonus points if we can find any critical information

- We can try to look around the file system with tools like grep to try and find sensitive data
- We can usually look for areas such as server/ var/ and usr/ see if you can find anything. (Hint: think of the other vulnerabilities this device might have)

grep command cheat sheet — bashsenpai.com

Basic Format

```
grep [options] [pattern] [file]
```

Key Options

```
-i    ⇒   Ignore case (for example, treat "abc" and "ABC" as equal)
-v    ⇒   Invert match (only show lines not matching pattern)
-r    ⇒   Recursive search
-R    ⇒   (same as -r)
-l    ⇒   Show only names of files containing the pattern
-n    ⇒   Show line numbers of lines containing the pattern
-c    ⇒   Count how many lines match the pattern
-w    ⇒   Match whole word
-l    ⇒   Show only the names of files with matching lines,
          separated by newline
```

Examples

```
grep 'pattern' filename              ⇒   Search for a pattern within
                                         a file
grep 'string1\|string2' filename     ⇒   Search for lines containing
                                         string1 or string2
grep -r 'pattern' /dir               ⇒   Recursive search for pattern
                                         in /dir
grep -w 'full word' filename         ⇒   Search for lines containing
                                         'full word'
grep -n 'pattern' filename           ⇒   Show line numbers of lines
                                         containing the pattern
grep -c 'pattern' filename           ⇒   Count the lines containing
                                         the pattern
grep --color 'pattern' filename      ⇒   Highlight the pattern in the
                                         output
grep -i 'pattern' filename           ⇒   Case-insensitive search
```

```
#
==========================================
Init zigbeectrl...T1
==========================================
Init - ZoneDevInfo!!
Init - System Config!!
System Config : Init - sSystemConf & system.conf!!
Init - Zone_Device_Status!!
Init - RD Information!!
Daemon start socket server
Init - UIS Log Information!!
UIS Log Init

 Init - Test Command
Opening serial port:/dev/ttyS1
killall: syslogd: no process killed

 Change Syslog Server syslogd -n -m 0 -R #>> system_log &
syslogd: option requires an argument -- R
uiscoServerDaemon: Started!
semAPI: reset fail!
shmAPI: reset fail!
semAPI: reset fail!
*** UISCO Server Daemon PID[1186] ***
Play record streams from this server using the URL
        rtsp://192.168.19.254/<MountPoint>/<FileFolder>/<FileName>
Play live   streams from this server using the URL
        rtsp://192.168.19.254/<DeviceMAC>/Stream?
cam_fg_arr[0].DeviceMAC ====xxxxxxxxxxx
cam_fg_arr[1].DeviceMAC ====xxxxxxxxxxx
cam_fg_arr[2].DeviceMAC ====xxxxxxxxxxx
cam_fg_arr[3].DeviceMAC ====xxxxxxxxxxx
CheckCommandThread: InitOK!
uiscoServerDaemon: init check command thread success!
-> Schedule Time Checking

#
#
#
# Init RX Therad..
Start   RX pthread.

--- Thread Uar Rx 1026 ---

--- Thread Send Remote Msg 2051 ---
Reset CC2530
pwd
/
# RTC Sync..
<Alive Check 35 Min.>
Set System Time
Sat Sep 13 00:28:12 UTC 2025
Sat Sep 13 00:28:12 UTC 2025
RD LOG Write X1->[2025/09/13 Sat 00:28:12 - Gateway NGW-240 Bootup]

PANId:0x2B79, Channel:0, Channel List:0x02000000, IEEE Addr:0x00124B0002F55452
PANId:0x2B79, IEEE Addr:0x00124B0002F55452, Channel List:0x02000000
Flash Write - SUCCESS!ls
bin
```

Further research

The Gateway has several more vulnerabilities in its web interface

Learn from other researchers

- Youtube Hardware hacking videos - Matt brown - Amazing IoT Hacking videos
- Hardware projects - Raspberry Pi
- Hardware CTFs - Bsides Badge challenges, exploit security CTF

Find yourself the next challenge!

- Investigate your own unused tech for any interesting devices
- Look for known vulnerable devices to practice your skills
- Pick up some cheap tech online or at second hand stores
- Get curious!

Thanks for attending!

Hack the planet!