IoT Firmware Dumpster Fire

Dumping IoT Firmware

for Non-Electrical Engineers

Lucipher | arntsonl@gmail.com | github.com/arntsonl

DC614 – Dec' 2019

Intro

Who Am I?

Why Would You Do This?

Firmware Dumping vs. Analysis

Hardware Hacking is Fun!

How to Approach the Topic

Consumer IoT Devices are Designed like Discount Computers

Computer:

CPU, Memory, Hard Drive, PSU, Peripherals

PCB Board:

System-on-a-Chip (SoC)/CPU, DRAM, Flash Chips, VCC Regulators, Random ICs

Things You Must Learn (Hint: YouTube)

Reading Chips & Finding Datasheets

PCB Boards & Following Traces

Understanding Voltage (High / Low)

Clock Cycles & Timing (What is a MegaHurts?)

Dumping Firmware Must Knows

SPI, I2C, UART, Etc. are Just Communication Interfaces

Computers -> IoT : USB, Parallel Port, Ethernet Port, Firewire

Chips *Usually* Follow Their Datasheets

Generally, the “CPU” Does Not Store Firmware

Learn By Doing, Not By Reading

Disassemble Everything!

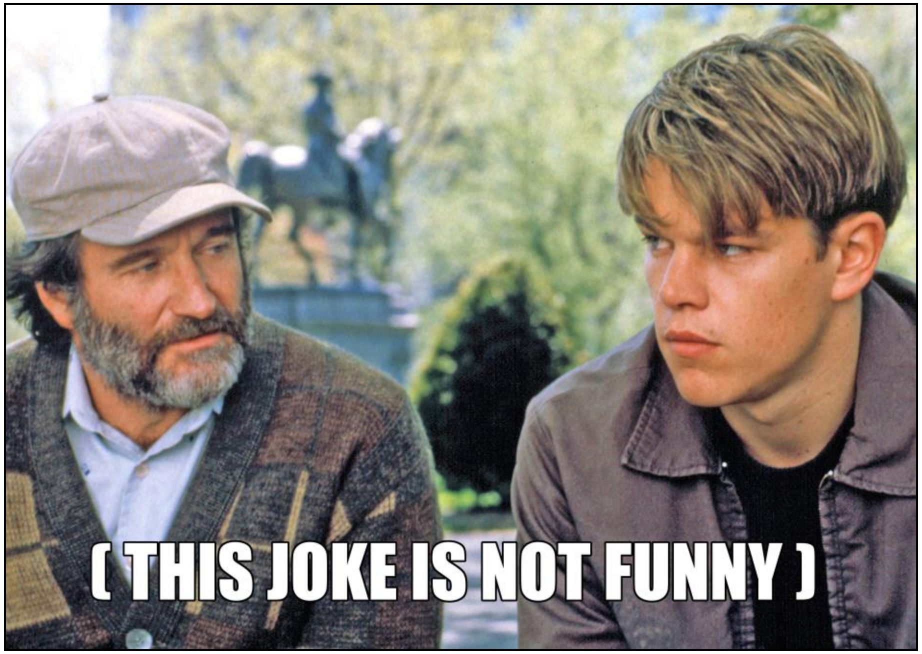
Start Easy! Find Tutorials on Hackaday/Reddit

Follow Tutorials! CS/CE/EE Schooling Does Not Teach This!

Practice, Rinse, Repeat

Goodwill Hunting

Go to Goodwill, Buy Some Crap



Anything < $10 is Fair Game

My Finds:

- Netgear R6200 Internet Route : $4.99
- Arris SBG6700-AC Cable Modem : $4.99

Must Have Toolbox

Soldering Gear (Flux, Tips, Kapton, Braid)

Logic Analyzer (DSLogic Plus, Salae Logic Pro 16, ~100mhz)

BusPirate (1-wire, UART, i2C, SPI, limited JTAG)

Arduino or FTDI USB (Easy Serial Interface)

Lab / Nice to Haves

Jeweler’s Loop / Magnifying Headset or Desk Magnifier

Good Lighting / Workmat

Helping Hands / PCB Vice

(Not Mine)

How Do I Read Chips?

Basic Firmware Dumping – Lesson 1

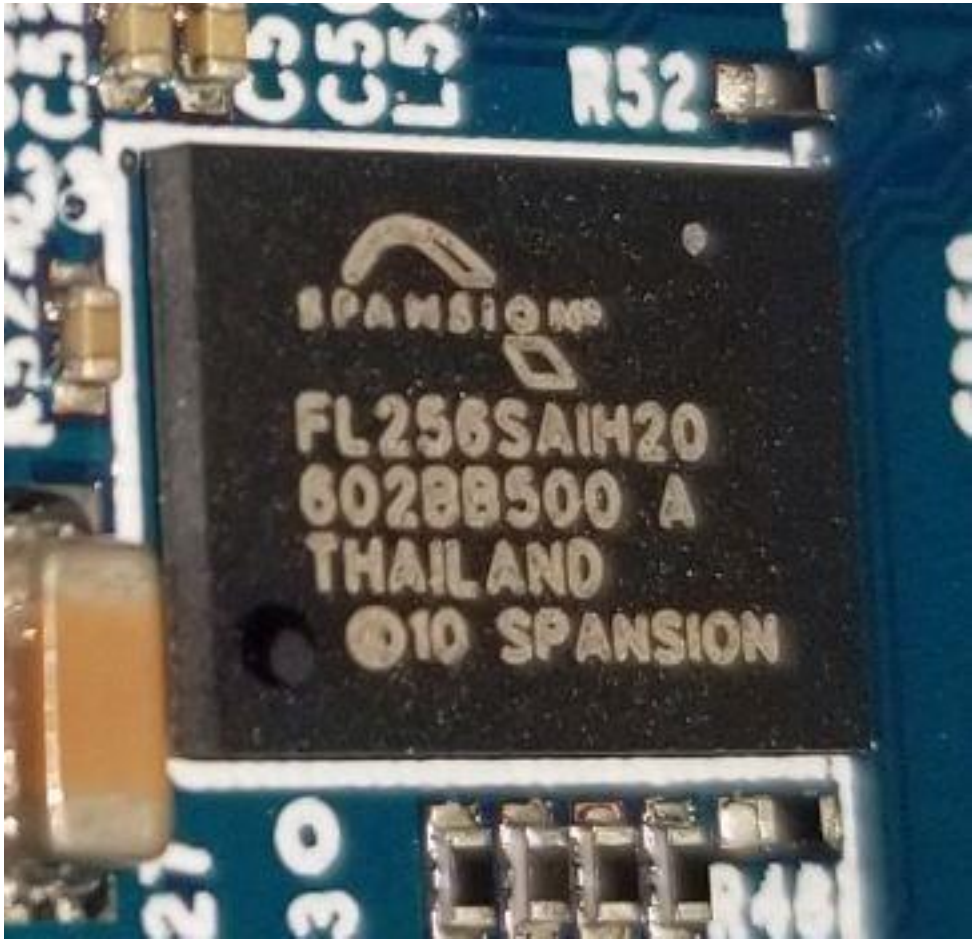
Find Each Line of Text on the Chip (Logo is Important)

Google the First X Number of Characters of Top Line

Try Just Clumps of Alpha-Numeric (TC, TC58, TC58NVG, TC58NVG0S3), Find Family of Chips

Basic Firmware Dumping – Lesson 1

Example Chip (Phone Camera w/ Sharp Angle)

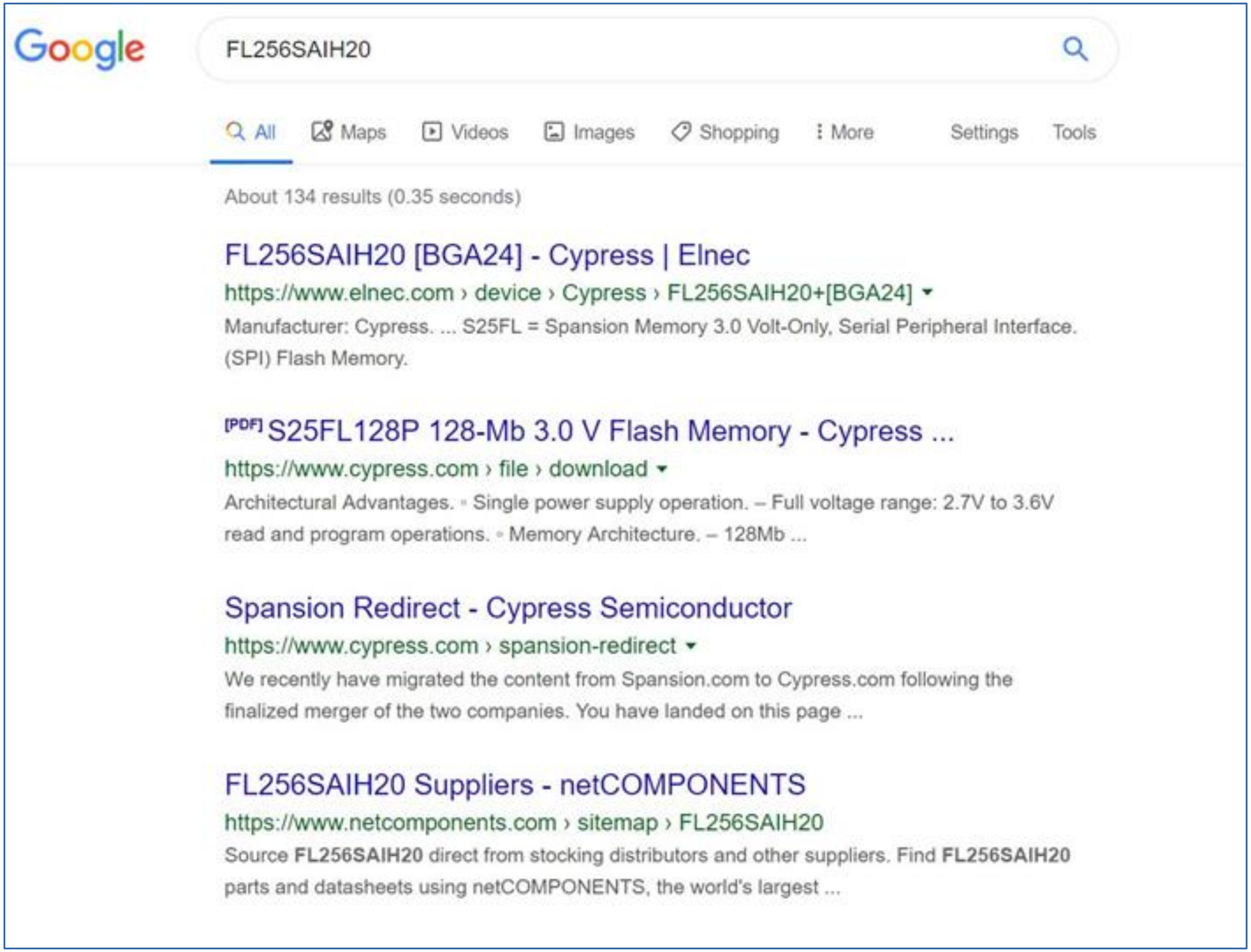


Chip Maker: Spansion

Part Number: FL256SAIH20

Orientation: Bottom-Left

Basic Firmware Dumping – Lesson 1



About 134 results (0.35 seconds)

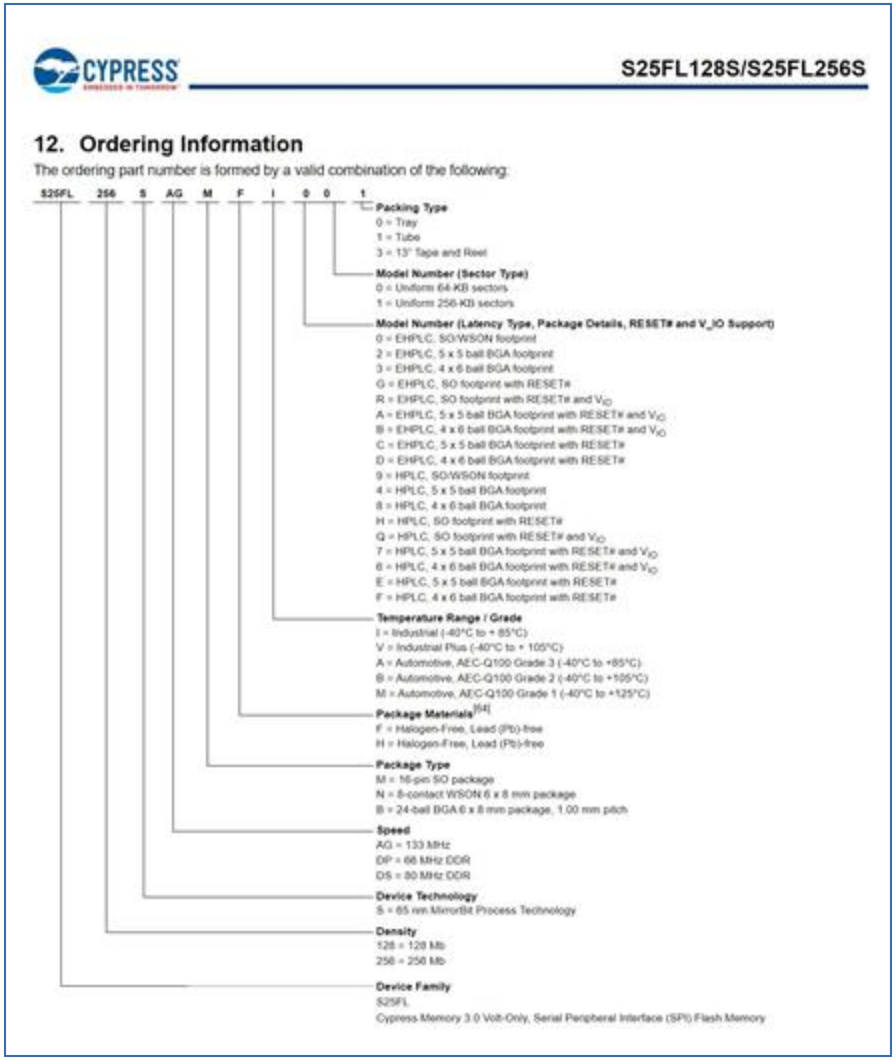
FL256SAIH20 [BGA24] - Cypress | Elnec
https://www.elnec.com › device › Cypress › FL256SAIH20+[BGA24] ▾
Manufacturer: Cypress. ... S25FL = Spansion Memory 3.0 Volt-Only, Serial Peripheral Interface. (SPI) Flash Memory.

[PDF] S25FL128P 128-Mb 3.0 V Flash Memory - Cypress ...
https://www.cypress.com › file › download ▾
Architectural Advantages. ▪ Single power supply operation. – Full voltage range: 2.7V to 3.6V read and program operations. ▪ Memory Architecture. – 128Mb ...

Spansion Redirect - Cypress Semiconductor
https://www.cypress.com › spansion-redirect ▾
We recently have migrated the content from Spansion.com to Cypress.com following the finalized merger of the two companies. You have landed on this page ...

FL256SAIH20 Suppliers - netCOMPONENTS
https://www.netcomponents.com › sitemap › FL256SAIH20
Source **FL256SAIH20** direct from stocking distributors and other suppliers. Find **FL256SAIH20** parts and datasheets using netCOMPONENTS, the world's largest ...

Basic Firmware Dumping – Lesson 1



S25FL128S/S25FL256S

12. Ordering Information

The ordering part number is formed by a valid combination of the following:

S25FL 256 S AG M F I 0 0 1

Packing Type
0 = Tray
1 = Tube
3 = 13" Tape and Reel

Model Number (Sector Type)
0 = Uniform 64-KB sectors
1 = Uniform 256-KB sectors

Model Number (Latency Type, Package Details, RESET# and V_IO Support)
0 = EHPLC, SO/WSON footprint
2 = EHPLC, 5 x 5 ball BGA footprint
3 = EHPLC, 4 x 6 ball BGA footprint
G = EHPLC, SO footprint with RESET#
R = EHPLC, SO footprint with RESET# and V_{IO}
A = EHPLC, 5 x 5 ball BGA footprint with RESET# and V_{IO}
B = EHPLC, 4 x 6 ball BGA footprint with RESET# and V_{IO}
C = EHPLC, 5 x 5 ball BGA footprint with RESET#
D = EHPLC, 4 x 6 ball BGA footprint with RESET#
9 = HPLC, SO/WSON footprint
4 = HPLC, 5 x 5 ball BGA footprint
8 = HPLC, 4 x 6 ball BGA footprint
H = HPLC, SO footprint with RESET#
Q = HPLC, SO footprint with RESET# and V_{IO}
7 = HPLC, 5 x 5 ball BGA footprint with RESET# and V_{IO}
6 = HPLC, 4 x 6 ball BGA footprint with RESET# and V_{IO}
E = HPLC, 5 x 5 ball BGA footprint with RESET#
F = HPLC, 4 x 6 ball BGA footprint with RESET#

Temperature Range / Grade
I = Industrial (-40°C to + 85°C)
V = Industrial Plus (-40°C to + 105°C)
A = Automotive, AEC-Q100 Grade 3 (-40°C to +85°C)
B = Automotive, AEC-Q100 Grade 2 (-40°C to +105°C)
M = Automotive, AEC-Q100 Grade 1 (-40°C to +125°C)

Package Materials[64]
F = Halogen-Free, Lead (Pb)-free
H = Halogen-Free, Lead (Pb)-free

Package Type
M = 16-pin SO package
N = 8-contact WSON 6 x 8 mm package
B = 24-ball BGA 6 x 8 mm package, 1.00 mm pitch

Speed
AG = 133 MHz
DP = 66 MHz DDR
DS = 80 MHz DDR

Device Technology
S = 65 nm MirrorBit Process Technology

Density
128 = 128 Mb
256 = 256 Mb

Device Family
S25FL
Cypress Memory 3.0 Volt-Only, Serial Peripheral Interface (SPI) Flash Memory

Translate to Cypress Order:
FL256SAIH20

FL – Device Family (S25FL)
256 – 256 Mb Density
S – 65 nm MirrorBit
AIH – Probably Package/Operating
2 – 5x5 Ball BGA
0 – Uniform 64-KB Sectors

Could Be Wrong, YOLO

Basic Firmware Dumping – Lesson 1

Usually “128” or “256” are size in Mbytes

Often You Identify a Non-Flash Chip

Work Through the Chips

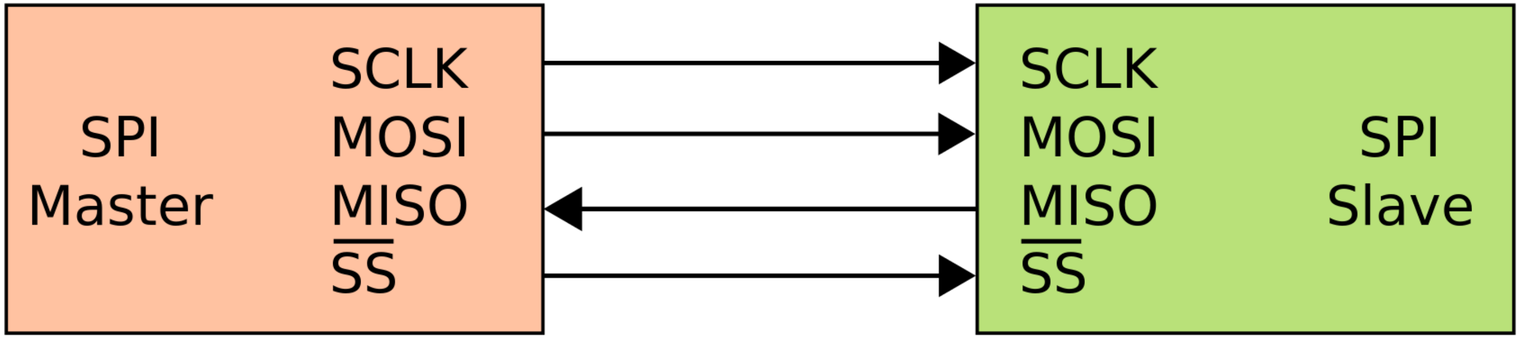
Chinese Clones (STLink->CMS, Etc.)

Dumping Chip In-Circuit with SPI

(Pronounced Spy)

Basic Firmware Dumping – Lesson 2

BusPirate is SPI Master, Flash Chip is SPI Slave

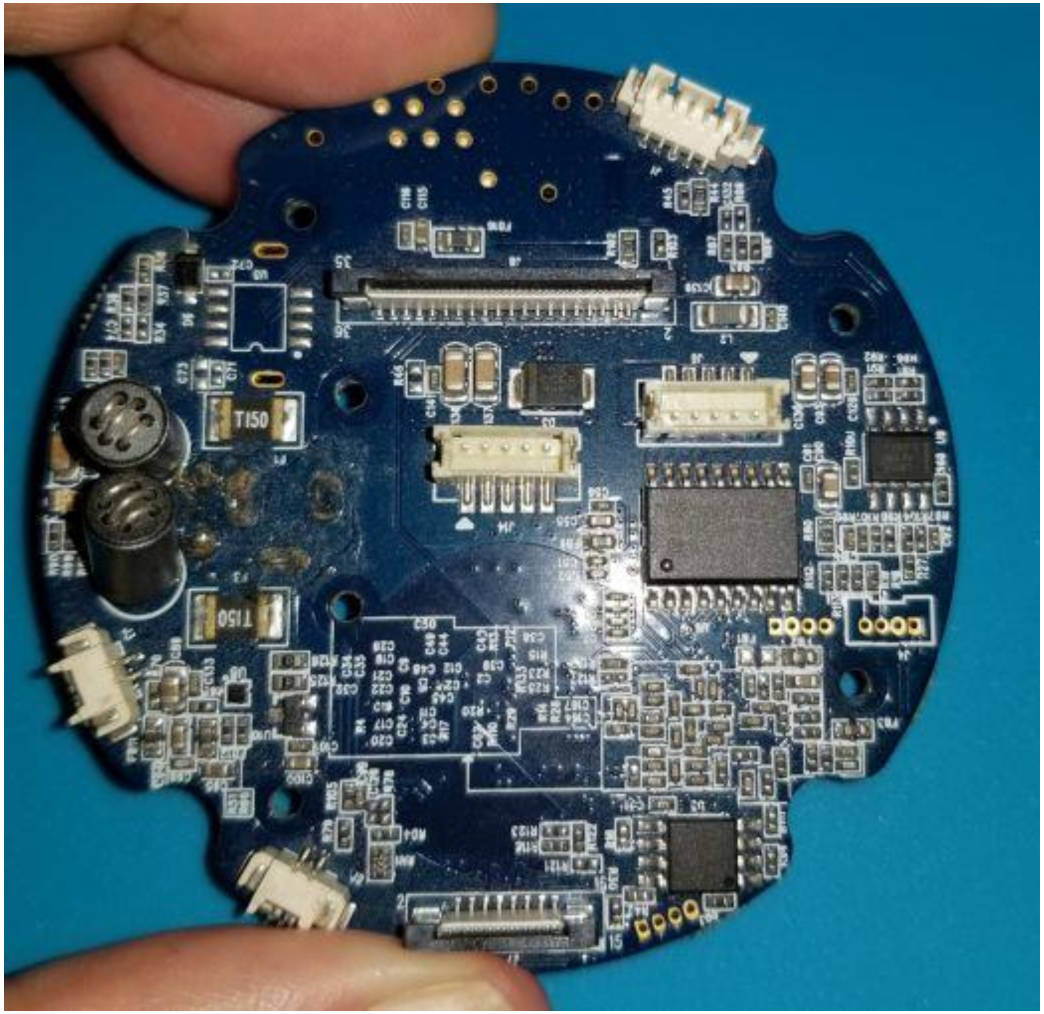


https://en.wikipedia.org/wiki/Serial_Peripheral_Interface

Basic Firmware Dumping – Lesson 2

Which Chip is Flash?

Guesses Anyone?



Basic Firmware Dumping – Lesson 2

Find a SPI Flash Chip with Available Pins/Connections

Datasheet says Interface (I2C, SPI, Etc.)

Connect SPI Interface (BusPirate) to MOSI, MISO, CS#, CLK, VCC, (Connect/Bridge Hold & WP to VCC)

“flashrom” to Interface and Dump Chip

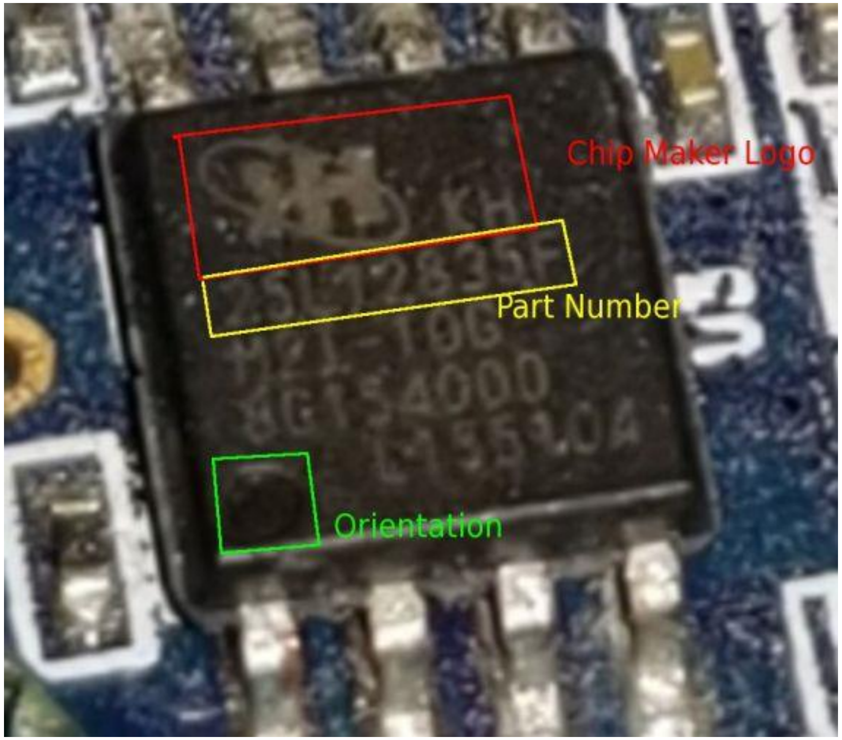
Basic Firmware Dumping – Lesson 2

Zmodo IoT 360 Wi-Fi Camera

“KH” Brand, 25L12835F, 8 Pins

Clearly Marked SPI Flash Chip

Easy to Reach Pins, Good Example



Basic Firmware Dumping – Lesson 2

KH25L12835F

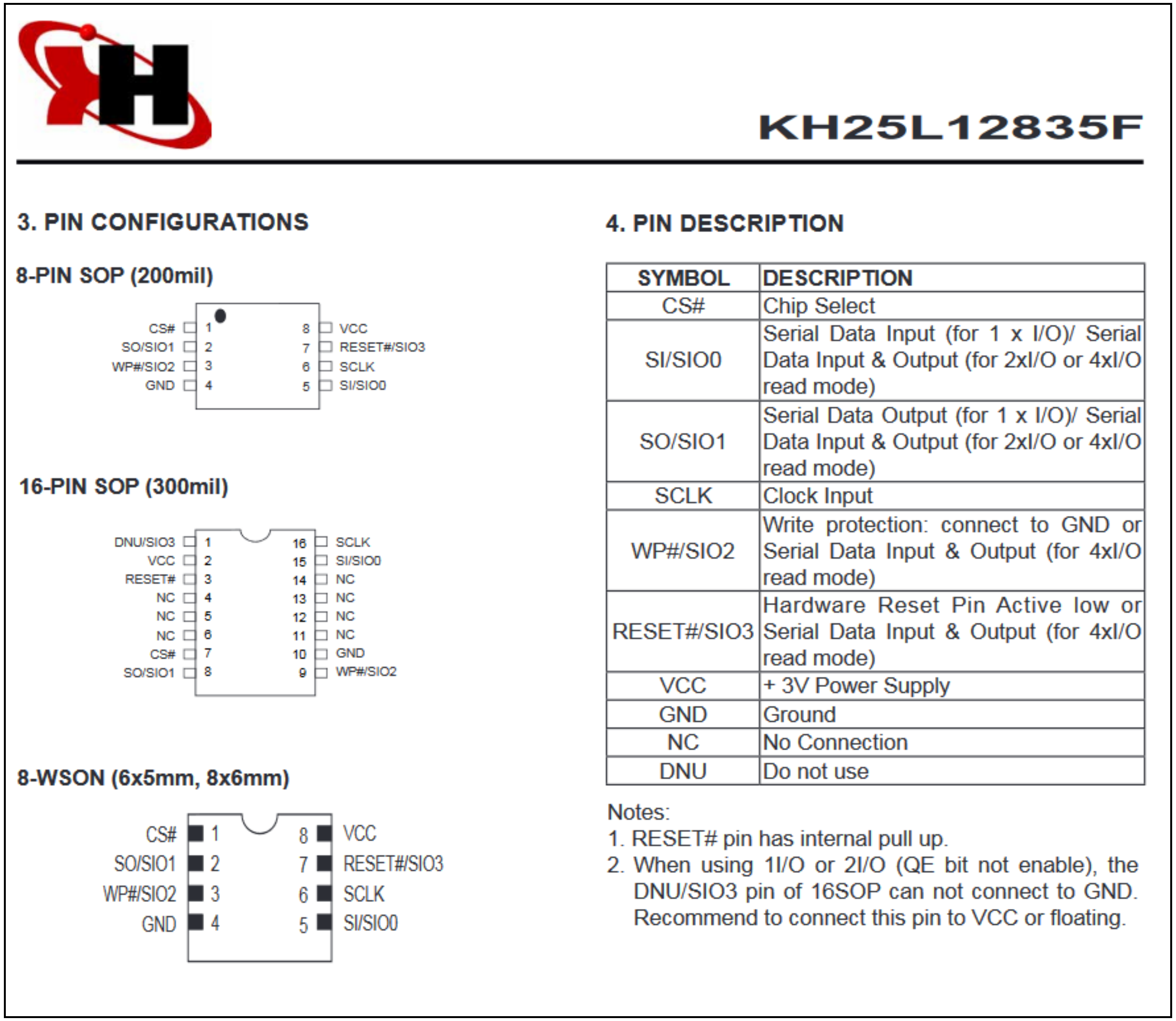
8-PIN SOP

Notes:

SO/SIO1 = MISO

SI/SIO0 = MOSI

WP/RESET = Bridge VCC



KH25L12835F

3. PIN CONFIGURATIONS

8-PIN SOP (200mil)

CS# 1
SO/SIO1 2
WP#/SIO2 3
GND 4
8 VCC
7 RESET#/SIO3
6 SCLK
5 SI/SIO0

16-PIN SOP (300mil)

DNU/SIO3 1
VCC 2
RESET# 3
NC 4
NC 5
NC 6
CS# 7
SO/SIO1 8
16 SCLK
15 SI/SIO0
14 NC
13 NC
12 NC
11 NC
10 GND
9 WP#/SIO2

8-WSON (6x5mm, 8x6mm)

CS# 1
SO/SIO1 2
WP#/SIO2 3
GND 4
8 VCC
7 RESET#/SIO3
6 SCLK
5 SI/SIO0

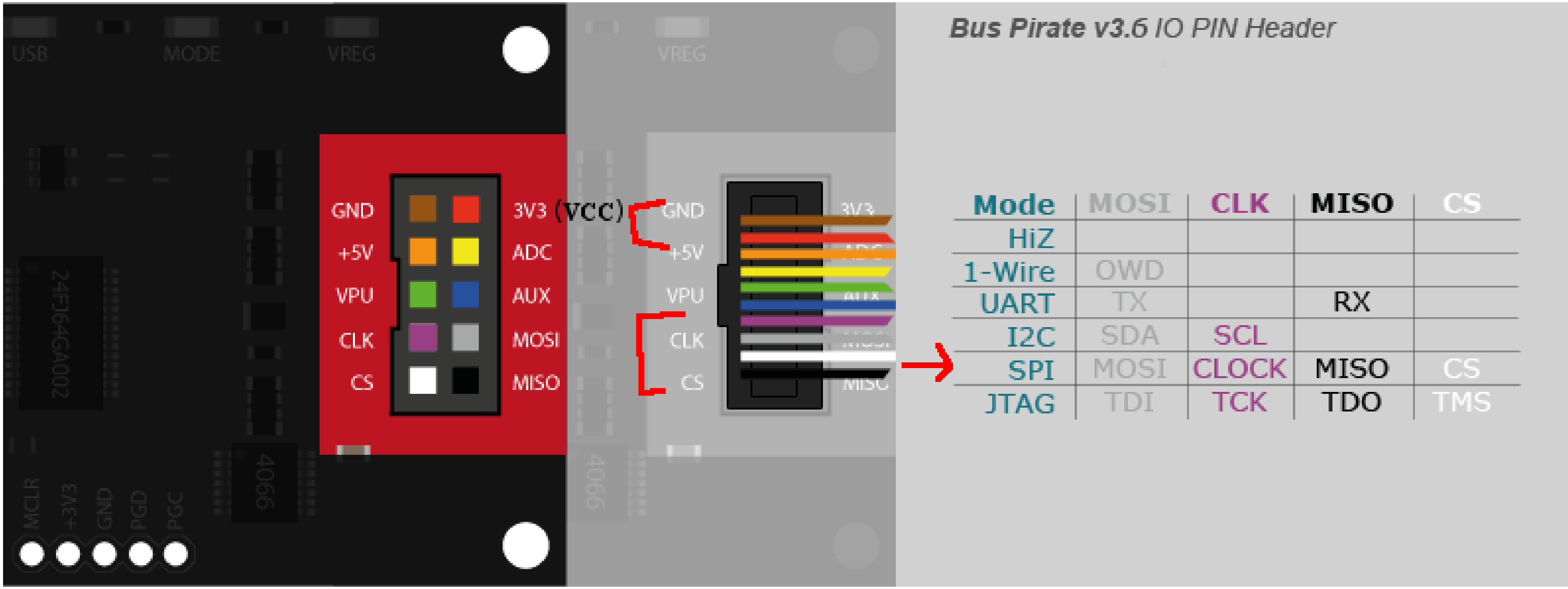
4. PIN DESCRIPTION

SYMBOL	DESCRIPTION
CS#	Chip Select
SI/SIO0	Serial Data Input (for 1 x I/O)/ Serial Data Input & Output (for 2xI/O or 4xI/O read mode)
SO/SIO1	Serial Data Output (for 1 x I/O)/ Serial Data Input & Output (for 2xI/O or 4xI/O read mode)
SCLK	Clock Input
WP#/SIO2	Write protection: connect to GND or Serial Data Input & Output (for 4xI/O read mode)
RESET#/SIO3	Hardware Reset Pin Active low or Serial Data Input & Output (for 4xI/O read mode)
VCC	+ 3V Power Supply
GND	Ground
NC	No Connection
DNU	Do not use

Notes:
1. RESET# pin has internal pull up.
2. When using 1I/O or 2I/O (QE bit not enable), the DNU/SIO3 pin of 16SOP can not connect to GND. Recommend to connect this pin to VCC or floating.

Basic Firmware Dumping – Lesson 2

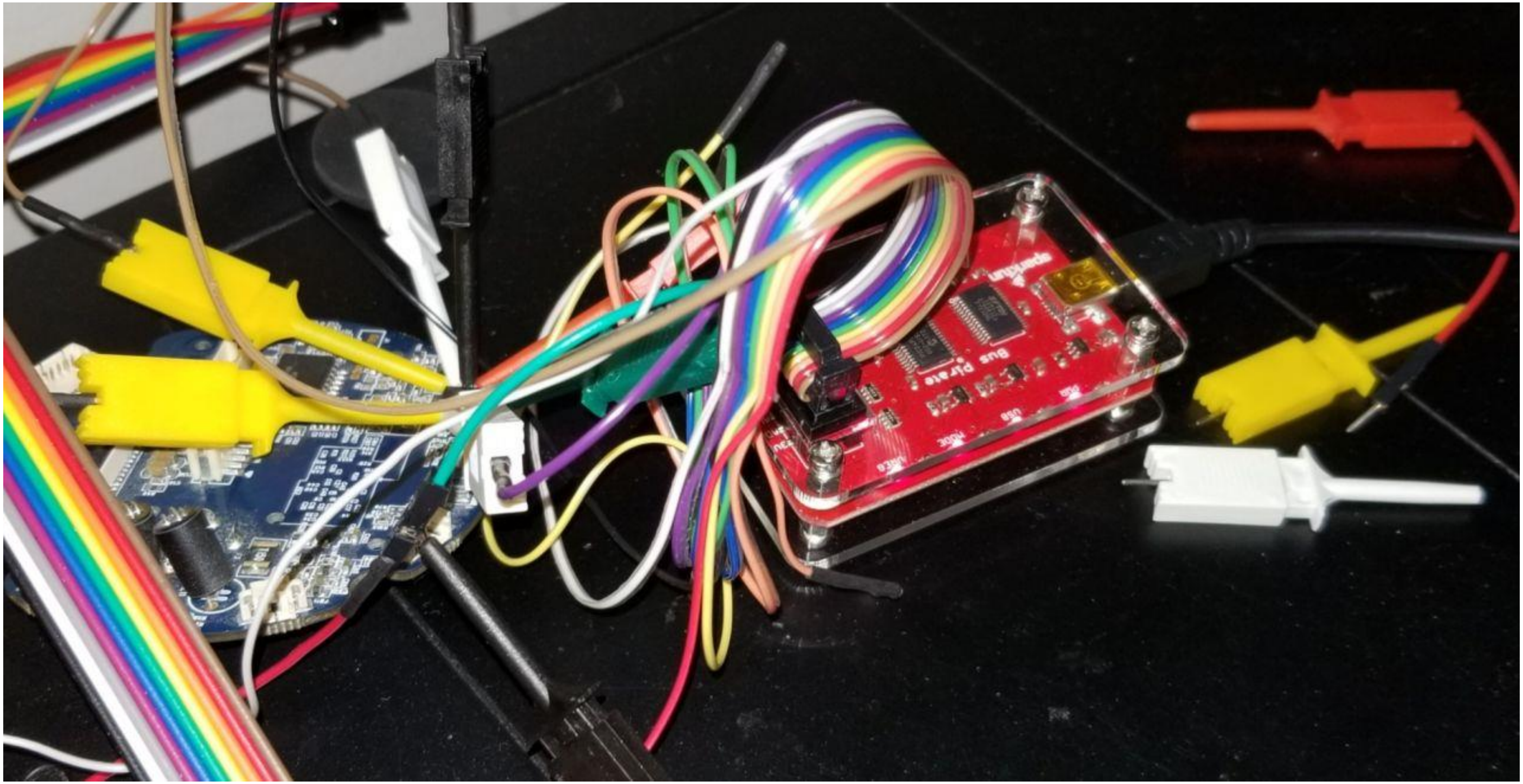
BusPirate v3.6 (SPI Interface)



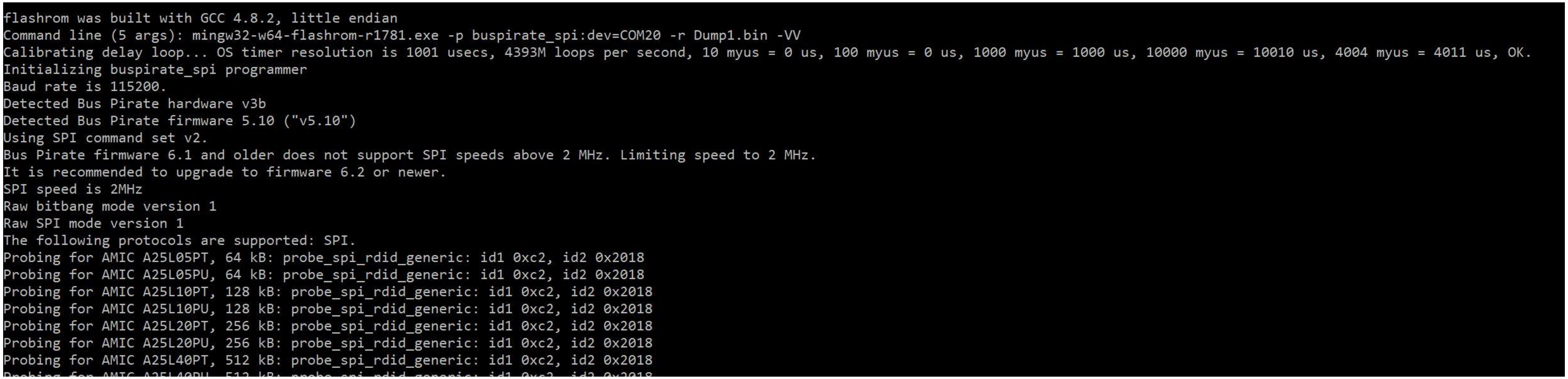
Bus Pirate v3.6 IO PIN Header

Mode	MOSI	CLK	MISO	CS
HiZ				
1-Wire	OWD			
UART	TX		RX	
I2C	SDA	SCL		
SPI	MOSI	CLOCK	MISO	CS
JTAG	TDI	TCK	TDO	TMS

Basic Firmware Dumping – Lesson 2

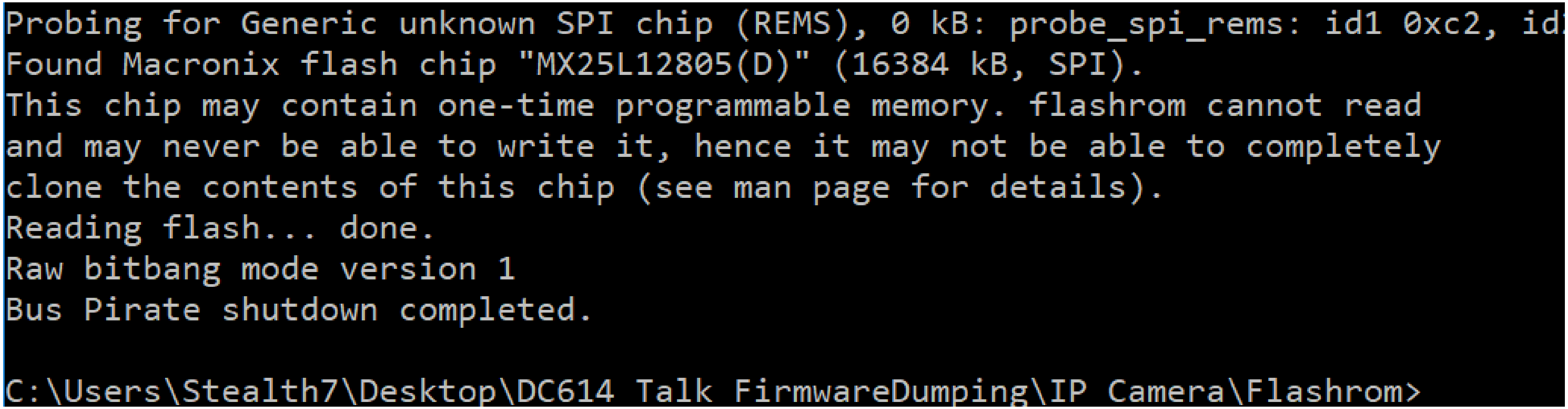


Basic Firmware Dumping – Lesson 2



```
flashrom was built with GCC 4.8.2, little endian
Command line (5 args): mingw32-w64-flashrom-r1781.exe -p buspirate_spi:dev=COM20 -r Dump1.bin -VV
Calibrating delay loop... OS timer resolution is 1001 usecs, 4393M loops per second, 10 myus = 0 us, 100 myus = 0 us, 1000 myus = 1000 us, 10000 myus = 10010 us, 4004 myus = 4011 us, OK.
Initializing buspirate_spi programmer
Baud rate is 115200.
Detected Bus Pirate hardware v3b
Detected Bus Pirate firmware 5.10 ("v5.10")
Using SPI command set v2.
Bus Pirate firmware 6.1 and older does not support SPI speeds above 2 MHz. Limiting speed to 2 MHz.
It is recommended to upgrade to firmware 6.2 or newer.
SPI speed is 2MHz
Raw bitbang mode version 1
Raw SPI mode version 1
The following protocols are supported: SPI.
Probing for AMIC A25L05PT, 64 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2018
Probing for AMIC A25L05PU, 64 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2018
Probing for AMIC A25L10PT, 128 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2018
Probing for AMIC A25L10PU, 128 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2018
Probing for AMIC A25L20PT, 256 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2018
Probing for AMIC A25L20PU, 256 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2018
Probing for AMIC A25L40PT, 512 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2018
Probing for AMIC A25L40PU, 512 kB: probe_spi_rdid_generic: id1 0xc2, id2 0x2018
```

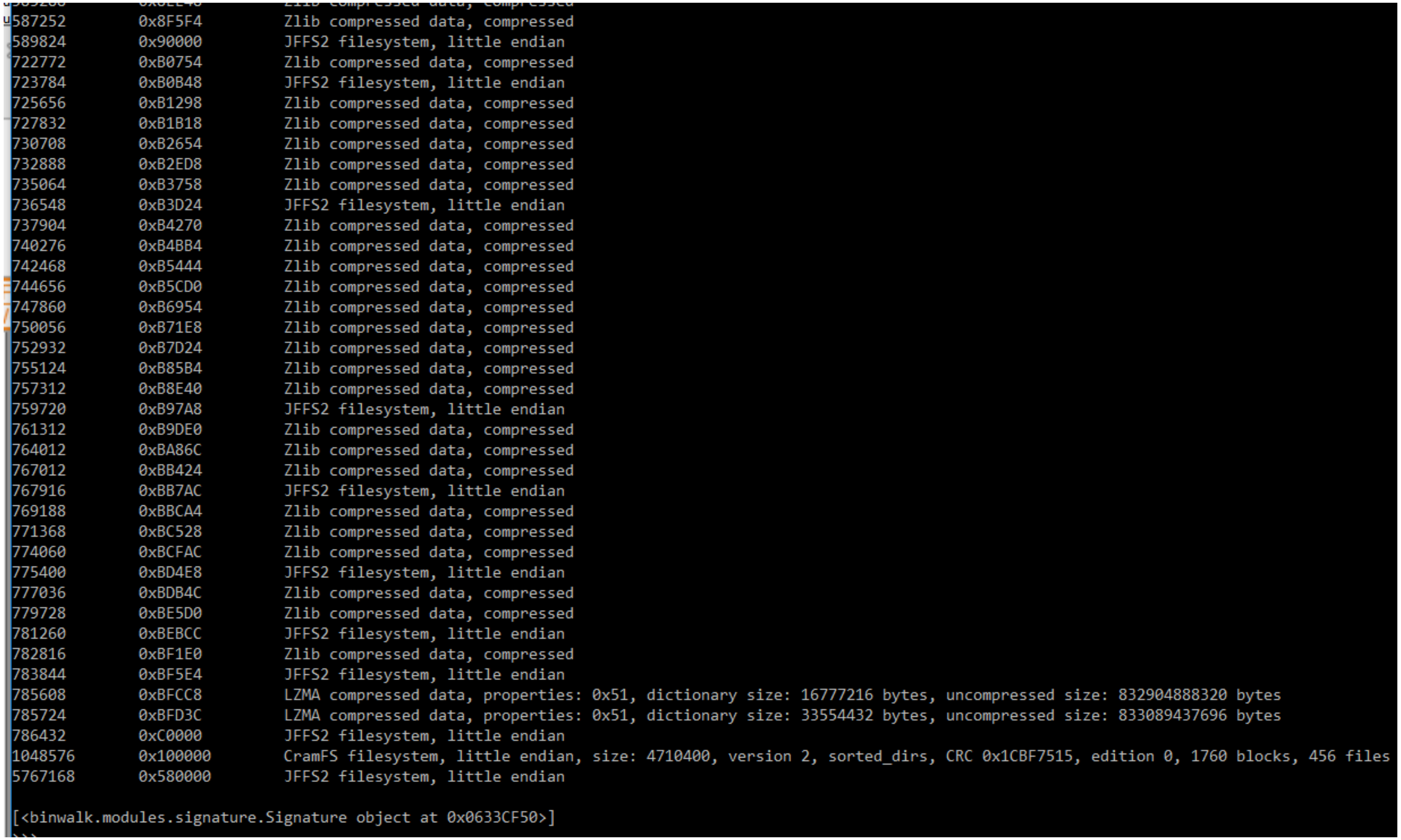
Basic Firmware Dumping – Lesson 2



```
Probing for Generic unknown SPI chip (REMS), 0 kB: probe_spi_rems: id1 0xc2, id
Found Macronix flash chip "MX25L12805(D)" (16384 kB, SPI).
This chip may contain one-time programmable memory. flashrom cannot read
and may never be able to write it, hence it may not be able to completely
clone the contents of this chip (see man page for details).
Reading flash... done.
Raw bitbang mode version 1
Bus Pirate shutdown completed.

C:\Users\Stealth7\Desktop\DC614 Talk FirmwareDumping\IP Camera\Flashrom>
```

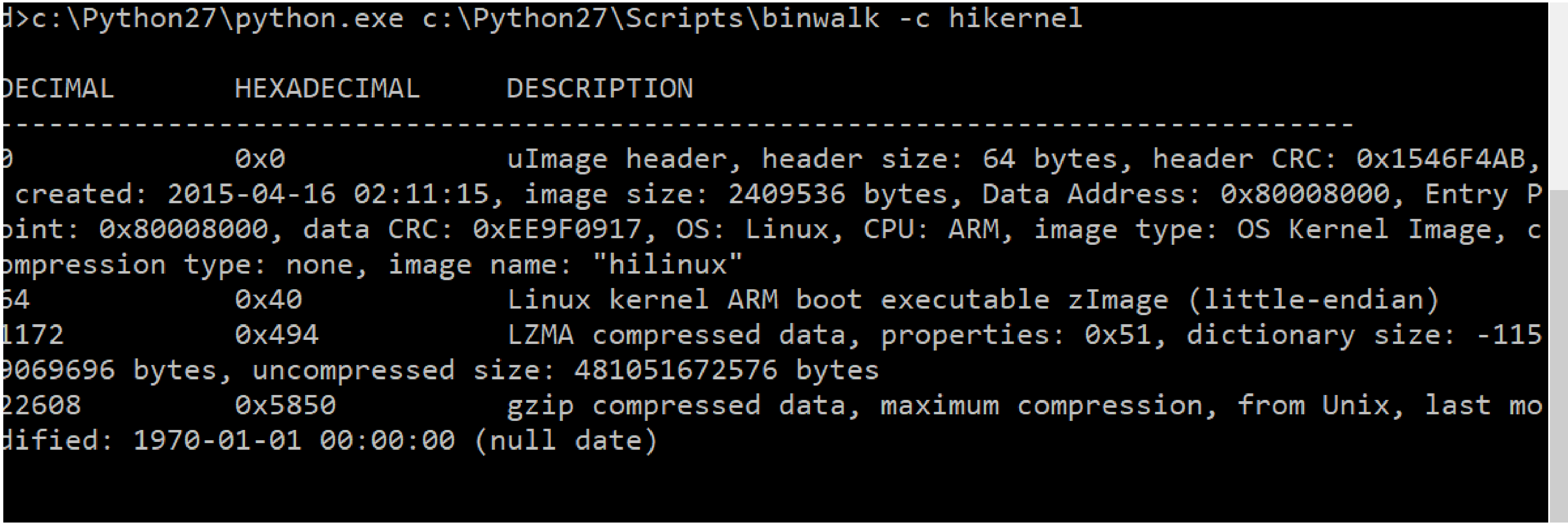
Basic Firmware Dumping – Lesson 2



```
587252        0x8F5F4       Zlib compressed data, compressed
589824        0x90000       JFFS2 filesystem, little endian
722772        0xB0754       Zlib compressed data, compressed
723784        0xB0B48       JFFS2 filesystem, little endian
725656        0xB1298       Zlib compressed data, compressed
727832        0xB1B18       Zlib compressed data, compressed
730708        0xB2654       Zlib compressed data, compressed
732888        0xB2ED8       Zlib compressed data, compressed
735064        0xB3758       Zlib compressed data, compressed
736548        0xB3D24       JFFS2 filesystem, little endian
737904        0xB4270       Zlib compressed data, compressed
740276        0xB4BB4       Zlib compressed data, compressed
742468        0xB5444       Zlib compressed data, compressed
744656        0xB5CD0       Zlib compressed data, compressed
747860        0xB6954       Zlib compressed data, compressed
750056        0xB71E8       Zlib compressed data, compressed
752932        0xB7D24       Zlib compressed data, compressed
755124        0xB85B4       Zlib compressed data, compressed
757312        0xB8E40       Zlib compressed data, compressed
759720        0xB97A8       JFFS2 filesystem, little endian
761312        0xB9DE0       Zlib compressed data, compressed
764012        0xBA86C       Zlib compressed data, compressed
767012        0xBB424       Zlib compressed data, compressed
767916        0xBB7AC       JFFS2 filesystem, little endian
769188        0xBBCA4       Zlib compressed data, compressed
771368        0xBC528       Zlib compressed data, compressed
774060        0xBCFAC       Zlib compressed data, compressed
775400        0xBD4E8       JFFS2 filesystem, little endian
777036        0xBDB4C       Zlib compressed data, compressed
779728        0xBE5D0       Zlib compressed data, compressed
781260        0xBEBCC       JFFS2 filesystem, little endian
782816        0xBF1E0       Zlib compressed data, compressed
783844        0xBF5E4       JFFS2 filesystem, little endian
785608        0xBFCC8       LZMA compressed data, properties: 0x51, dictionary size: 16777216 bytes, uncompressed size: 832904888320 bytes
785724        0xBFD3C       LZMA compressed data, properties: 0x51, dictionary size: 33554432 bytes, uncompressed size: 833089437696 bytes
786432        0xC0000       JFFS2 filesystem, little endian
1048576       0x100000      CramFS filesystem, little endian, size: 4710400, version 2, sorted_dirs, CRC 0x1CBF7515, edition 0, 1760 blocks, 456 files
5767168       0x580000      JFFS2 filesystem, little endian

[<binwalk.modules.signature.Signature object at 0x0633CF50>]
```

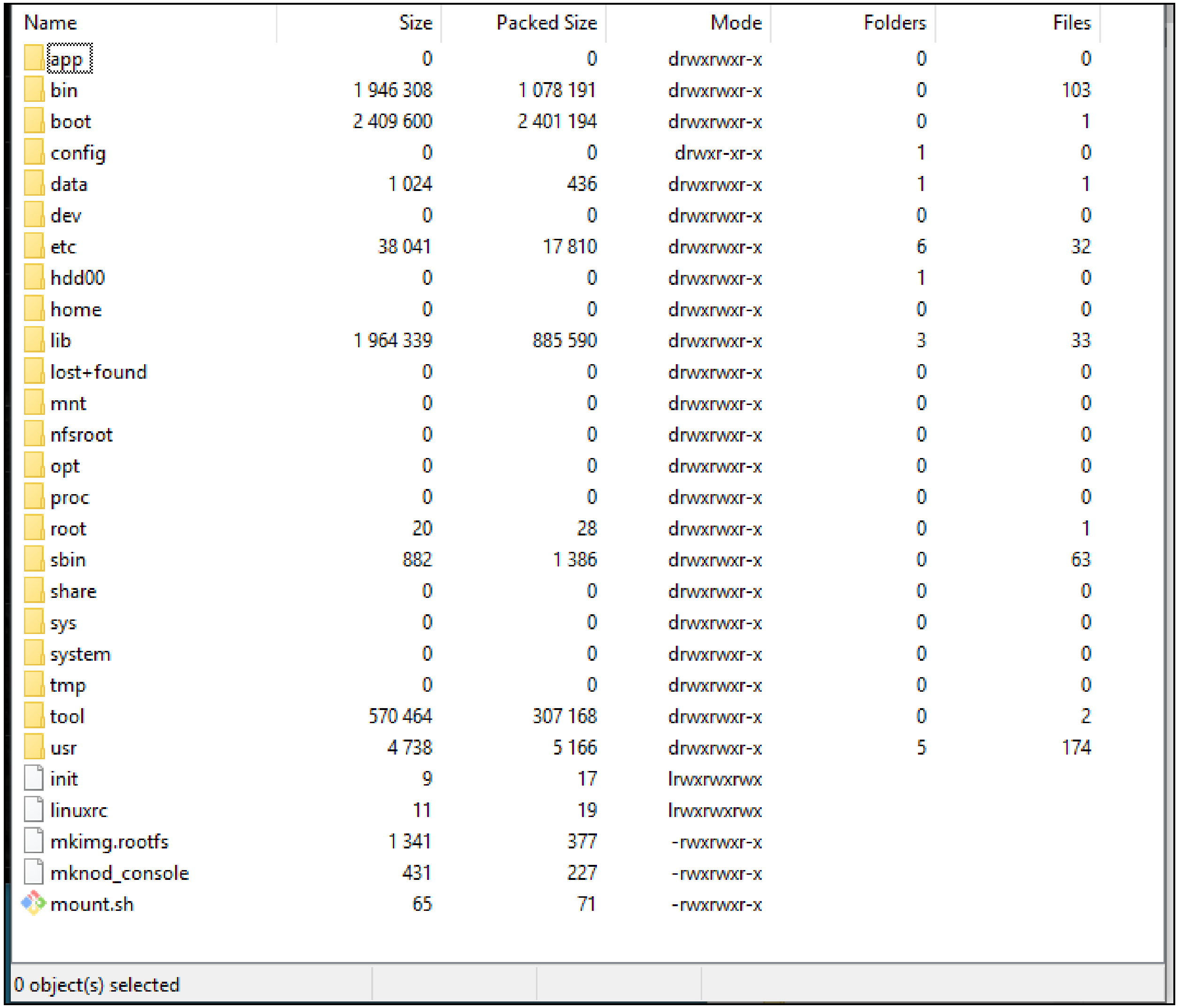
Basic Firmware Dumping – Lesson 2



```
d>c:\Python27\python.exe c:\Python27\Scripts\binwalk -c hikernel

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             uImage header, header size: 64 bytes, header CRC: 0x1546F4AB,
 created: 2015-04-16 02:11:15, image size: 2409536 bytes, Data Address: 0x80008000, Entry P
oint: 0x80008000, data CRC: 0xEE9F0917, OS: Linux, CPU: ARM, image type: OS Kernel Image, c
ompression type: none, image name: "hilinux"
64            0x40            Linux kernel ARM boot executable zImage (little-endian)
1172          0x494           LZMA compressed data, properties: 0x51, dictionary size: -115
9069696 bytes, uncompressed size: 481051672576 bytes
22608         0x5850          gzip compressed data, maximum compression, from Unix, last mo
dified: 1970-01-01 00:00:00 (null date)
```

Basic Firmware Dumping – Lesson 2

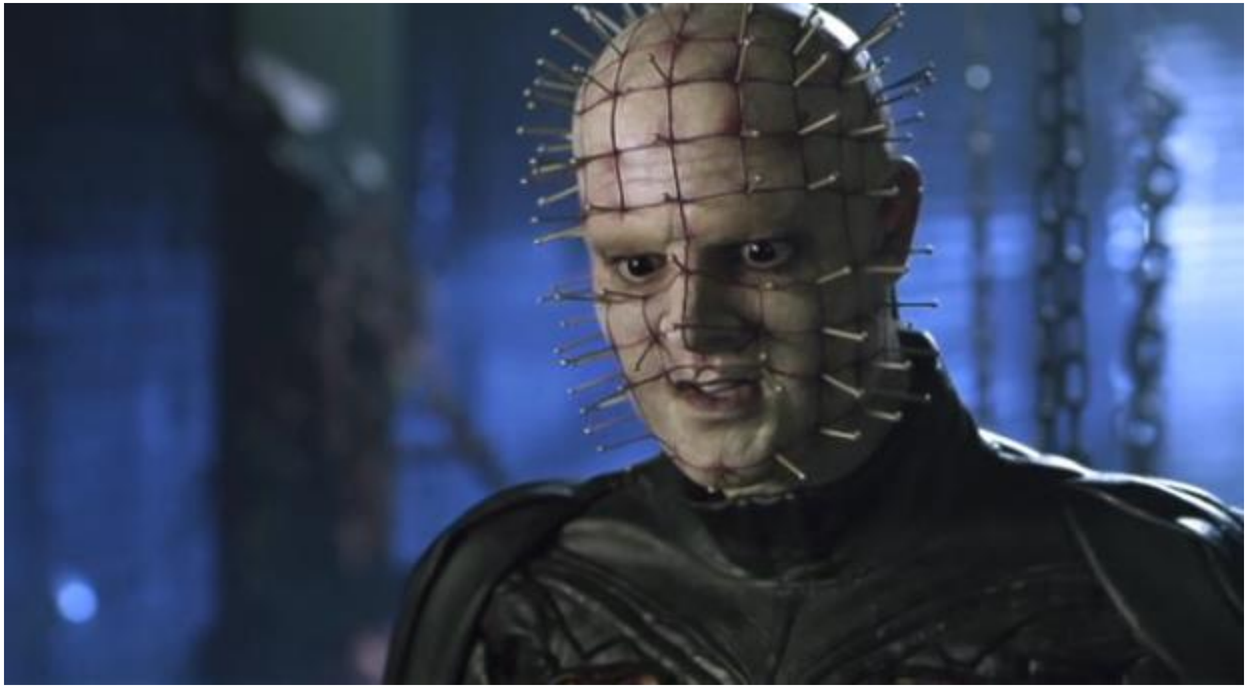


Name	Size	Packed Size	Mode	Folders	Files
app	0	0	drwxrwxr-x	0	0
bin	1 946 308	1 078 191	drwxrwxr-x	0	103
boot	2 409 600	2 401 194	drwxrwxr-x	0	1
config	0	0	drwxr-xr-x	1	0
data	1 024	436	drwxrwxr-x	1	1
dev	0	0	drwxrwxr-x	0	0
etc	38 041	17 810	drwxrwxr-x	6	32
hdd00	0	0	drwxrwxr-x	1	0
home	0	0	drwxrwxr-x	0	0
lib	1 964 339	885 590	drwxrwxr-x	3	33
lost+found	0	0	drwxrwxr-x	0	0
mnt	0	0	drwxrwxr-x	0	0
nfsroot	0	0	drwxrwxr-x	0	0
opt	0	0	drwxrwxr-x	0	0
proc	0	0	drwxrwxr-x	0	0
root	20	28	drwxrwxr-x	0	1
sbin	882	1 386	drwxrwxr-x	0	63
share	0	0	drwxrwxr-x	0	0
sys	0	0	drwxrwxr-x	0	0
system	0	0	drwxrwxr-x	0	0
tmp	0	0	drwxrwxr-x	0	0
tool	570 464	307 168	drwxrwxr-x	0	2
usr	4 738	5 166	drwxrwxr-x	5	174
init	9	17	lrwxrwxrwx		
linuxrc	11	19	lrwxrwxrwx		
mkimg.rootfs	1 341	377	-rwxrwxr-x		
mknod_console	431	227	-rwxrwxr-x		
mount.sh	65	71	-rwxrwxr-x		

0 object(s) selected

Basic Firmware Dumping – Lesson 3

Finding Debug Pins



Basic Firmware Dumping – Lesson 3

Look for Open Contact Points / Test Points

Rows of 4x2 or 4x1 are Usually Chip Testing/Boot Testing

Photoshop / GIMP Traces on Front/Back

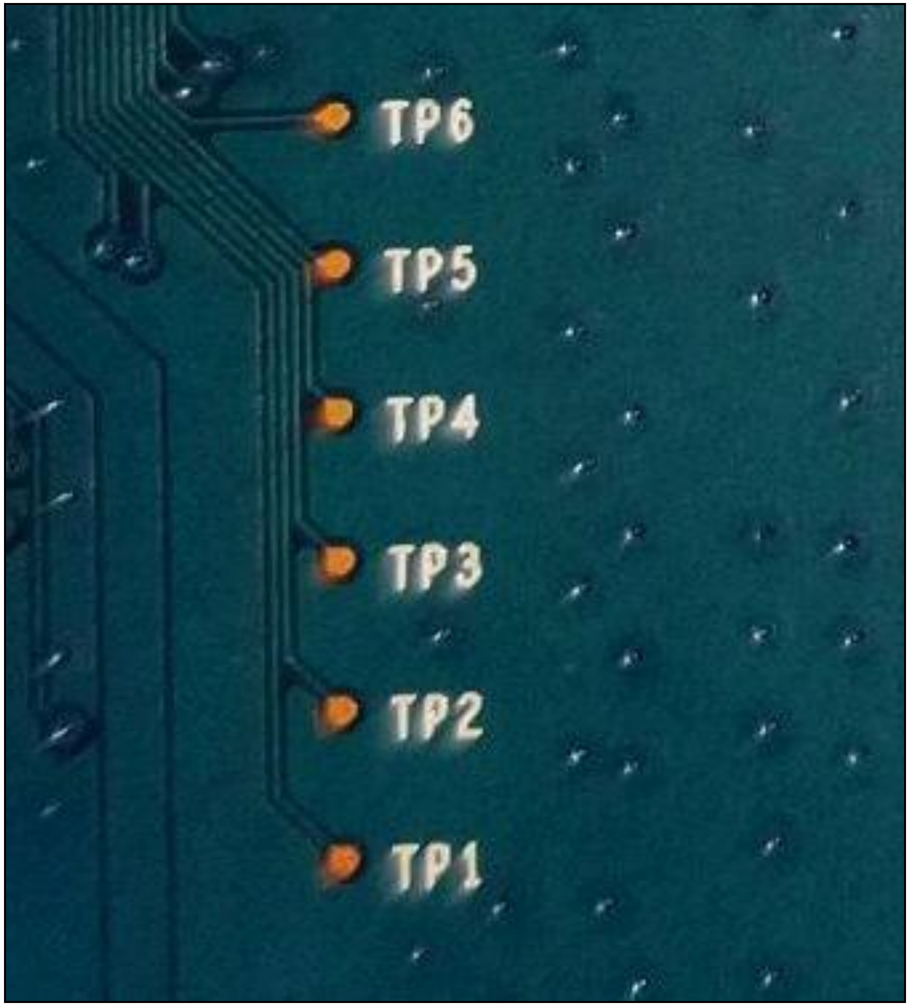
Sometimes YOLO (Gut Feeling)

Basic Firmware Dumping – Lesson 3

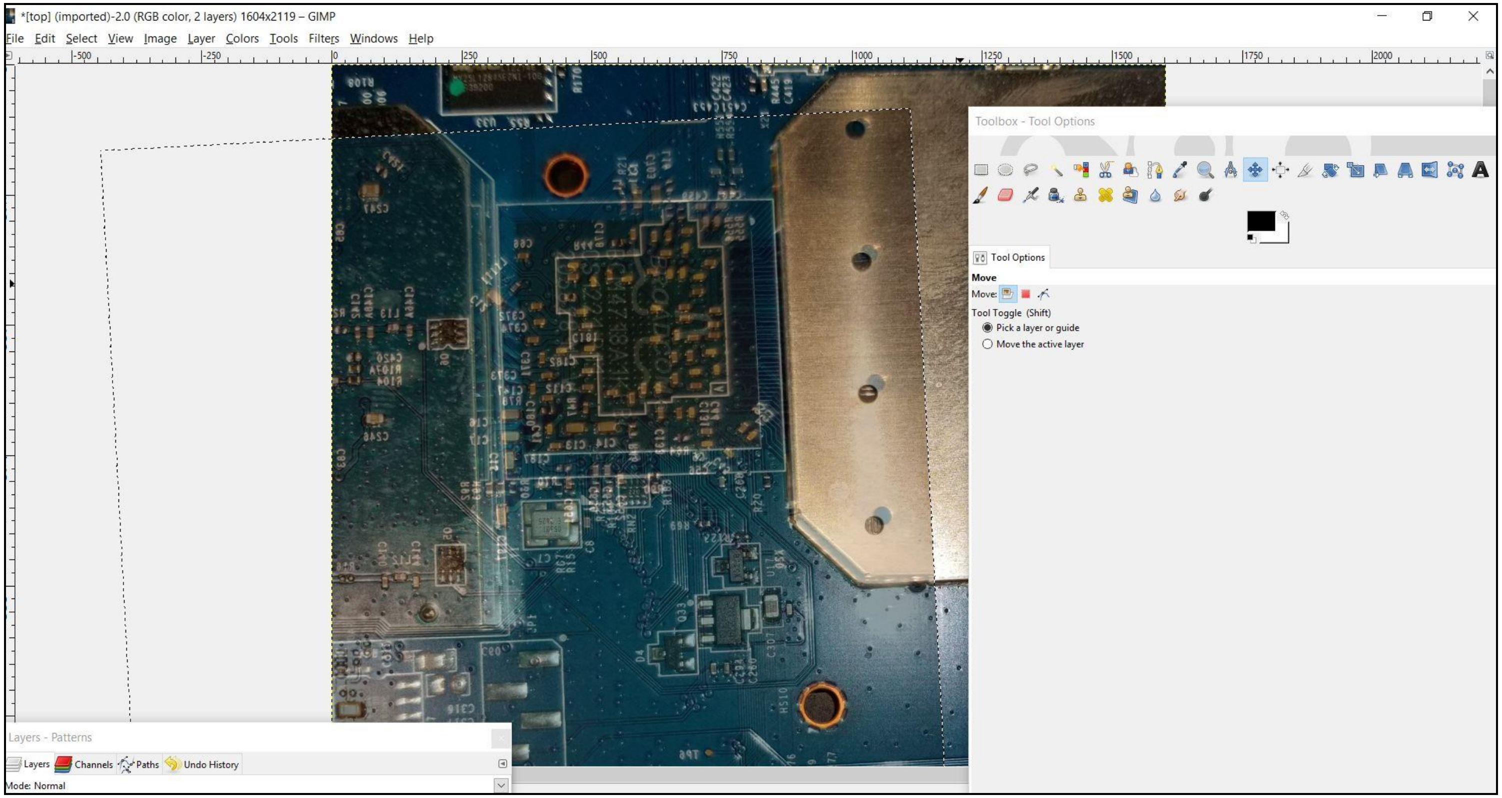
Netgear Router has mysterious TP1-6 Pins (Test Point?)

Take a Clear Picture of the Front and Back

Place Both Layers in GIMP/Photoshop



Basic Firmware Dumping – Lesson 3



*[top] (imported)-2.0 (RGB color, 2 layers) 1604x2119 – GIMP

Move

Move:

Tool Toggle (Shift)

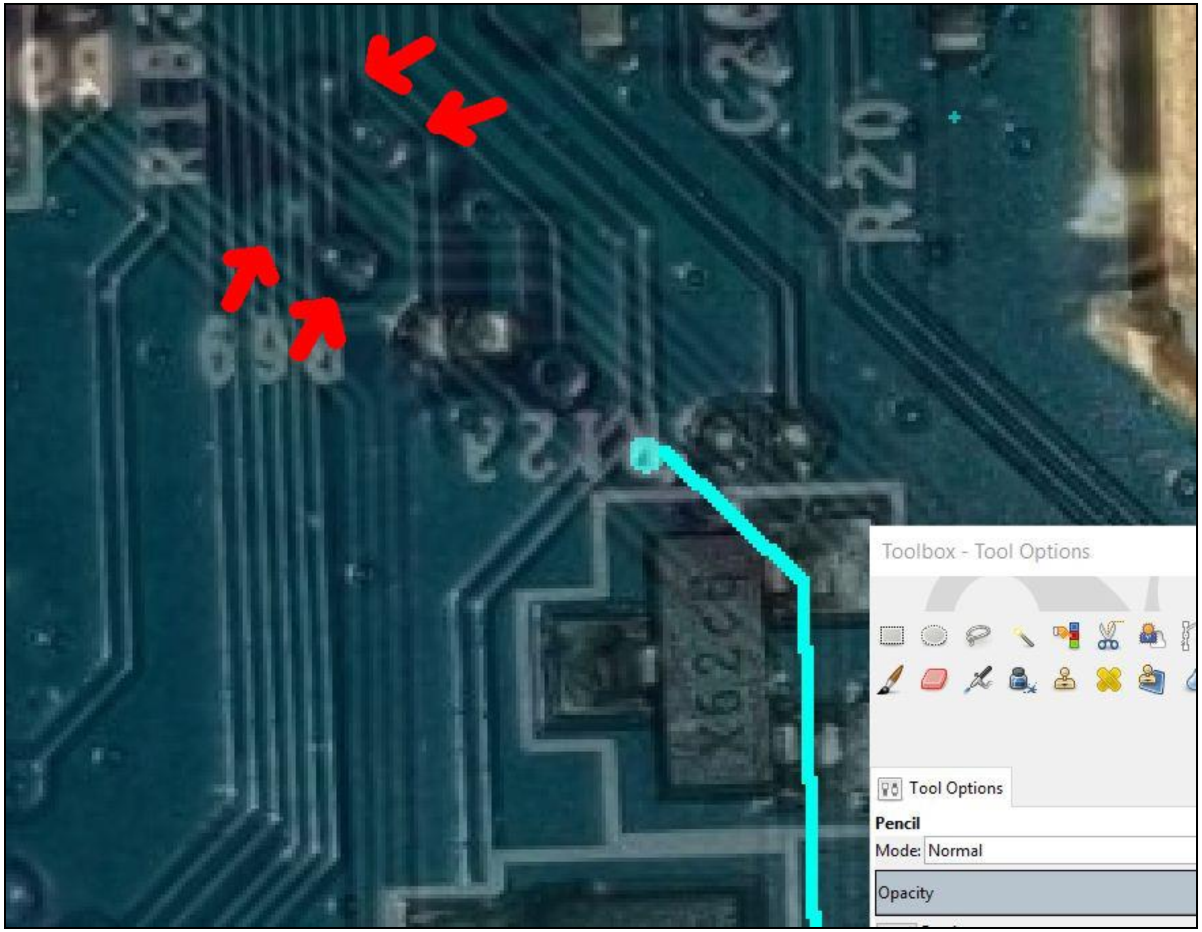
● Pick a layer or guide

○ Move the active layer

Mode: Normal

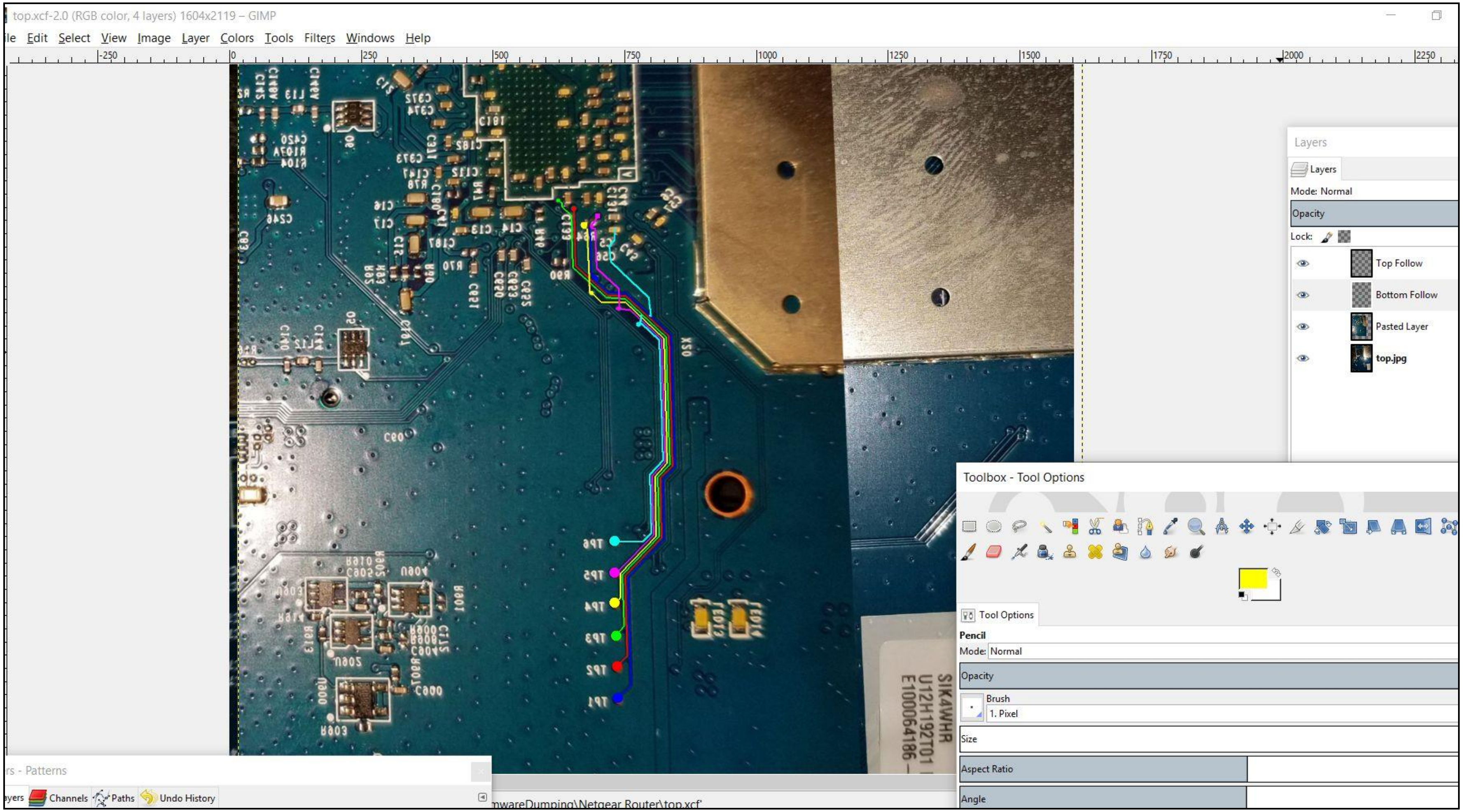
Top Layer as "Bottom", Bottom Layer as "Top", Rotate/Scale

Basic Firmware Dumping – Lesson 3



Pins are Not Aligned (Had to Adjust as Lines Made)

Basic Firmware Dumping – Lesson 3



top.xcf-2.0 (RGB color, 4 layers) 1604x2119 – GIMP

Layers

Mode: Normal

Top Follow

Bottom Follow

Pasted Layer

top.jpg

Toolbox - Tool Options

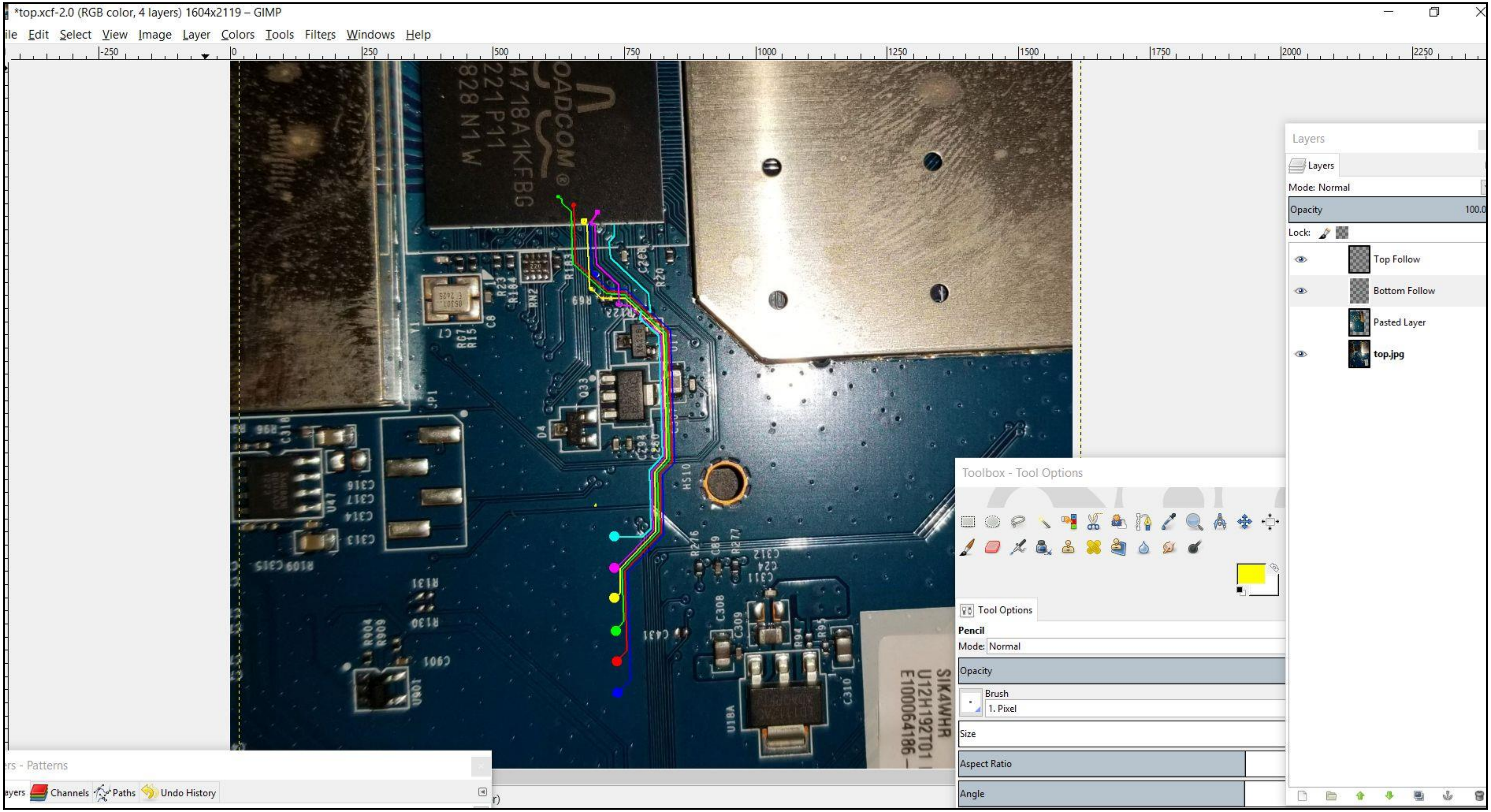
Pencil

Mode: Normal

Brush: 1. Pixel

mwareDumping\Netgear Router\top.xcf'

Basic Firmware Dumping – Lesson 3



*top.xcf-2.0 (RGB color, 4 layers) 1604x2119 – GIMP

Mode: Normal

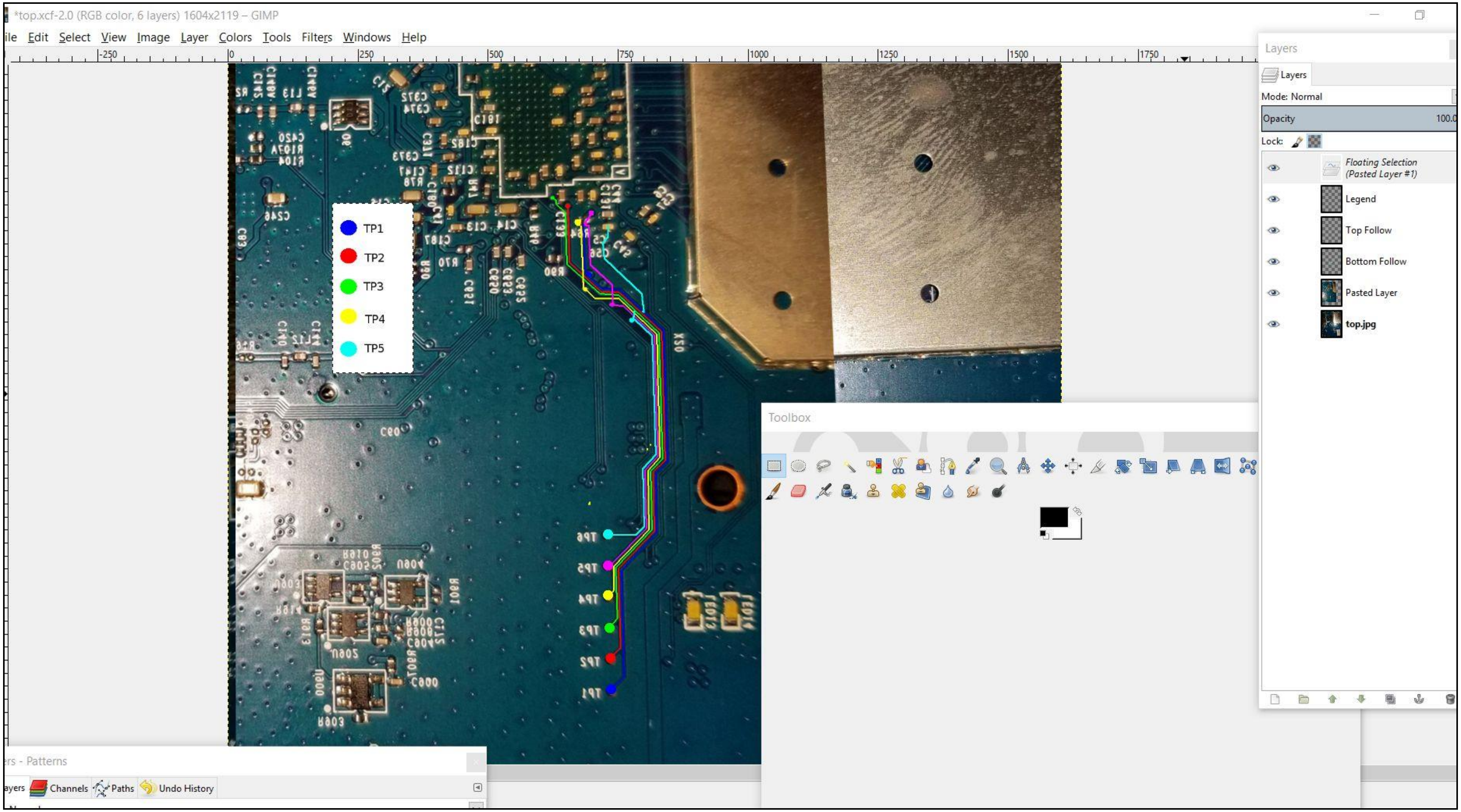
Opacity 100.0

Pencil

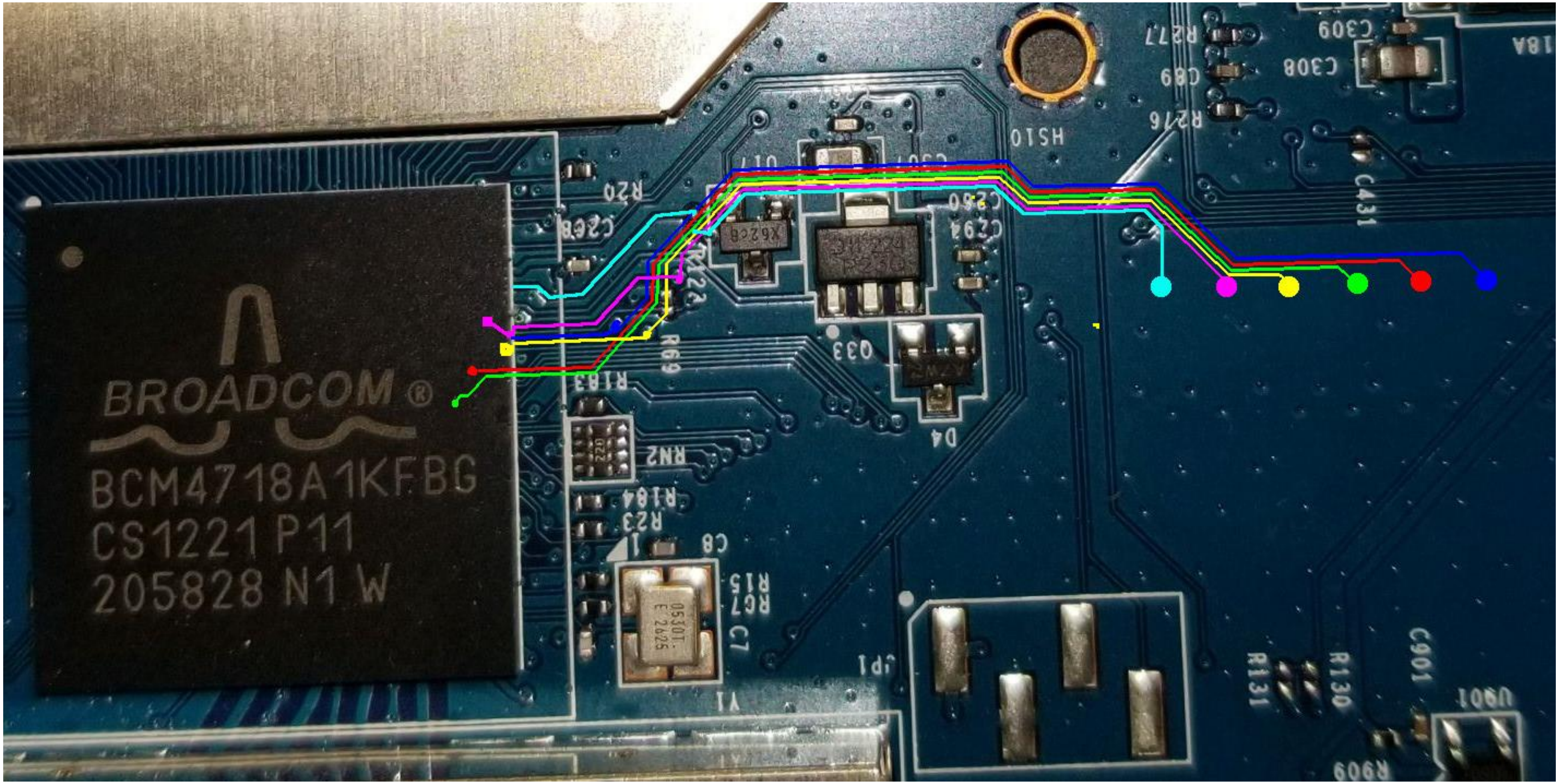
Mode: Normal

Brush 1. Pixel

Basic Firmware Dumping – Lesson 3


*top.xcf-2.0 (RGB color, 6 layers) 1604x2119 – GIMP
ile Edit Select View Image Layer Colors Tools Filters Windows Help
TP1
TP2
TP3
TP4
TP5
TP6
Toolbox
Layers
Mode: Normal
Opacity 100.0
Lock:
Floating Selection (Pasted Layer #1)
Legend
Top Follow
Bottom Follow
Pasted Layer
top.jpg
rs - Patterns
ayers Channels Paths Undo History

Basic Firmware Dumping – Lesson 3



Basic Firmware Dumping – Lesson 3

These TPs are Going to the Broadcom…

The JTAG interface, collectively known as a Test Access Port, or TAP, uses the following signals to support the operation of boundary scan.

TCK (Test Clock) – this signal synchronizes the internal state machine operations.

TMS (Test Mode Select) – this signal is sampled at the rising edge of TCK to determine the next state.

TDI (Test Data In) – this signal represents the data shifted into the device’s test or programming logic. It is sampled at the rising edge of TCK when the internal state machine is in the correct state.

TDO (Test Data Out) – this signal represents the data shifted out of the device’s test or programming logic and is valid on the falling edge of TCK when the internal state machine is in the correct state.

TRST (Test Reset) – this is an optional pin which, when available, can reset the TAP controller’s state machine.

TL;DR 6 Pins are Most Likely JTAG

Basic Firmware Dumping – Lesson 3

Interfacing JTAG Gives CPU Level Access and Commands

OpenOCD on Pi is Awesome

Custom JTAG Interface Tools (> $2000)

“BlackMagic Probe”, Arm Cortex M0 (Defcon 27 Badge)

Basic Firmware Dumping – Lesson 3

Sniff the Signals with a Logic Analyzer (Salae Pro, DSLogic, etc.)

In OpenOCD: dump_image <filename> <starting address> <size>

Left as Exercise for the Reader… (I ran out of time)

https://openwrt.org/docs/techref/hardware/port.jtag

https://openwrt.org/docs/guide-user/hardware/debrick.ath79.using.jtag

UART Hunting

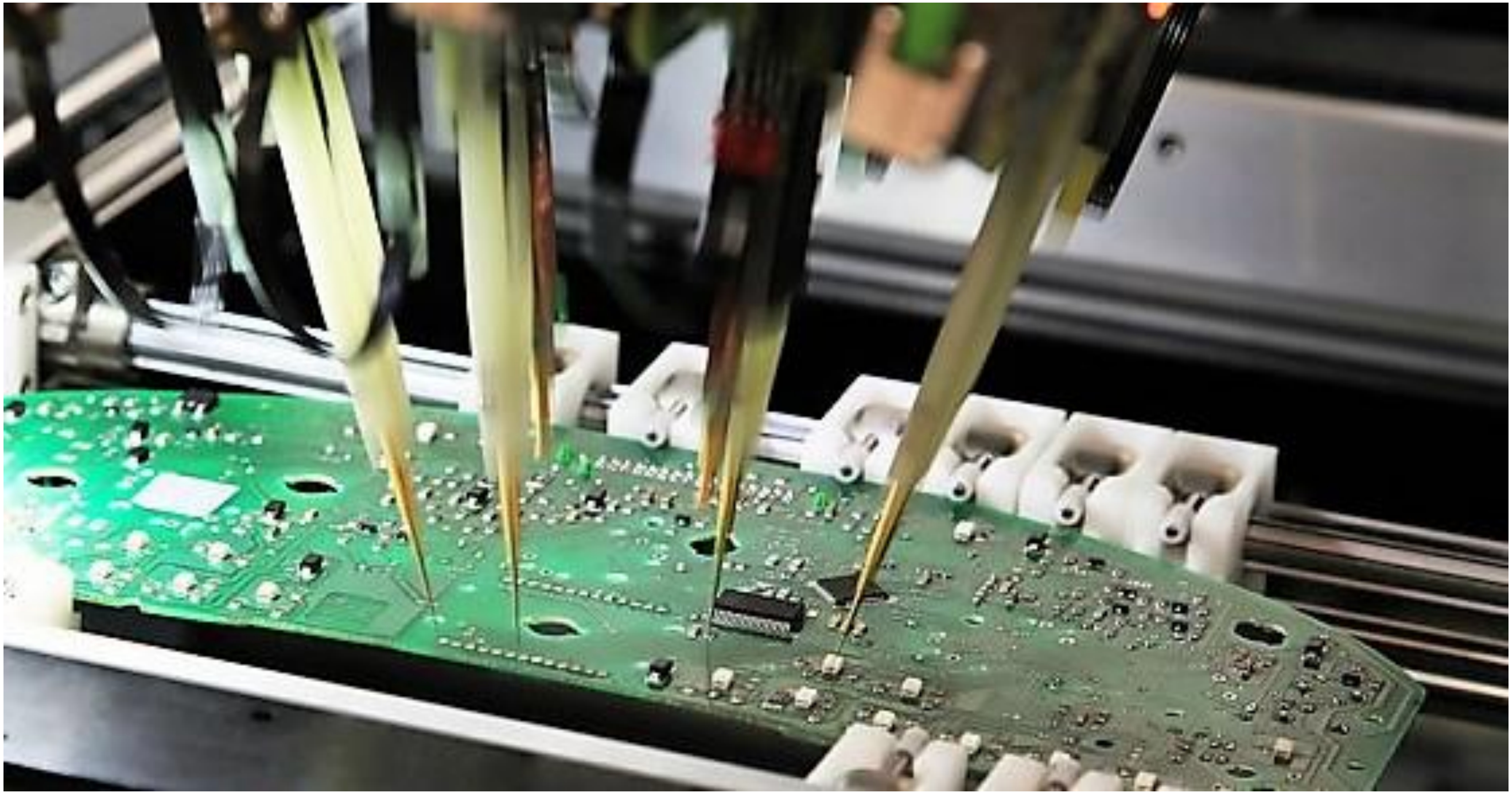
Serial Interface, TX/RX, Baud Rate, Common on Arduino Boards

Look for 4-Pin Connection Points (Imagine a Test Connector in a Factory)

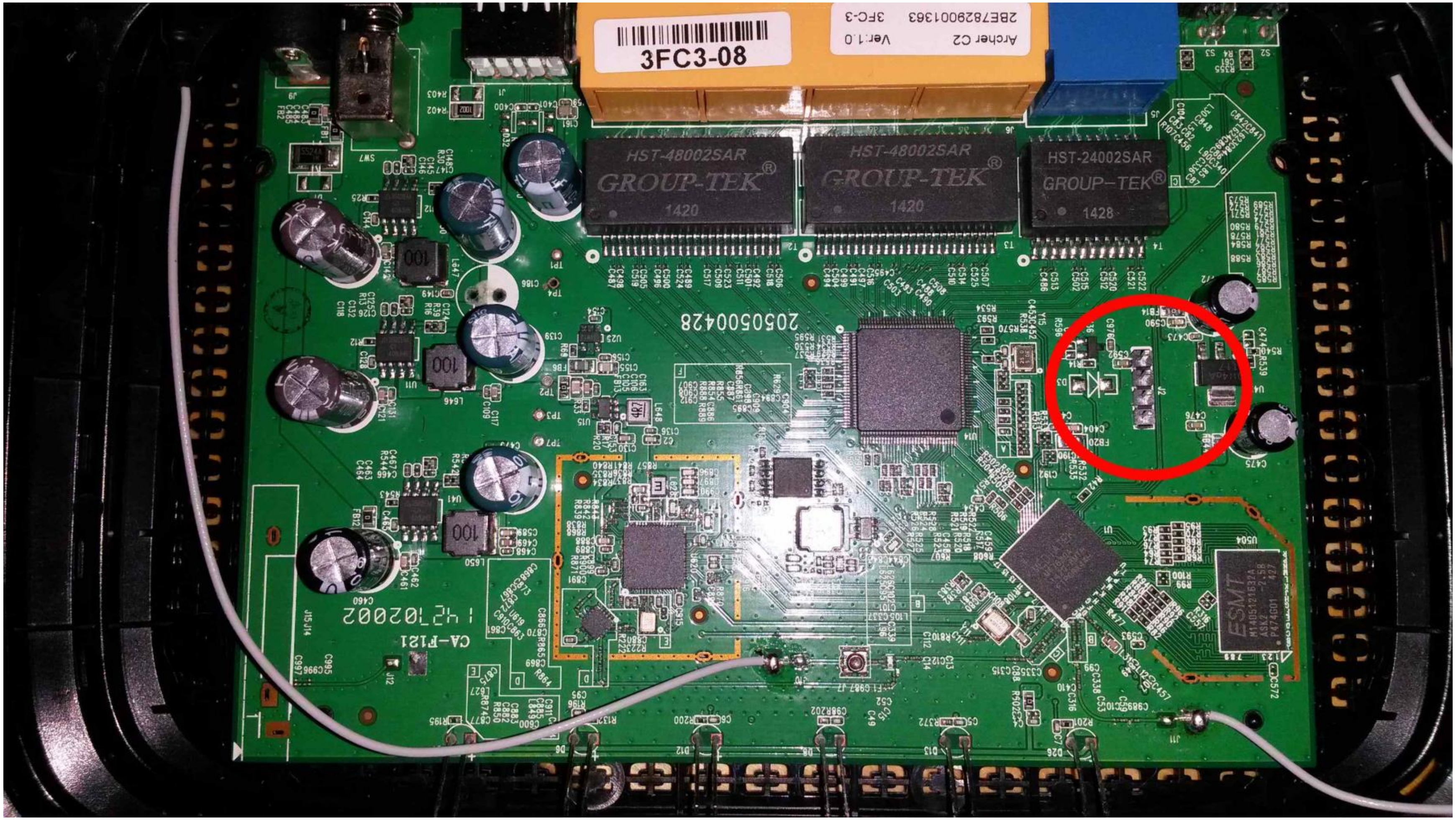
Using a Multimeter or Logic Analyzer, find the GND pin, and watch for 3v spikes on the other 3 pins (one should be VCC).

(Examples on Next Slides)

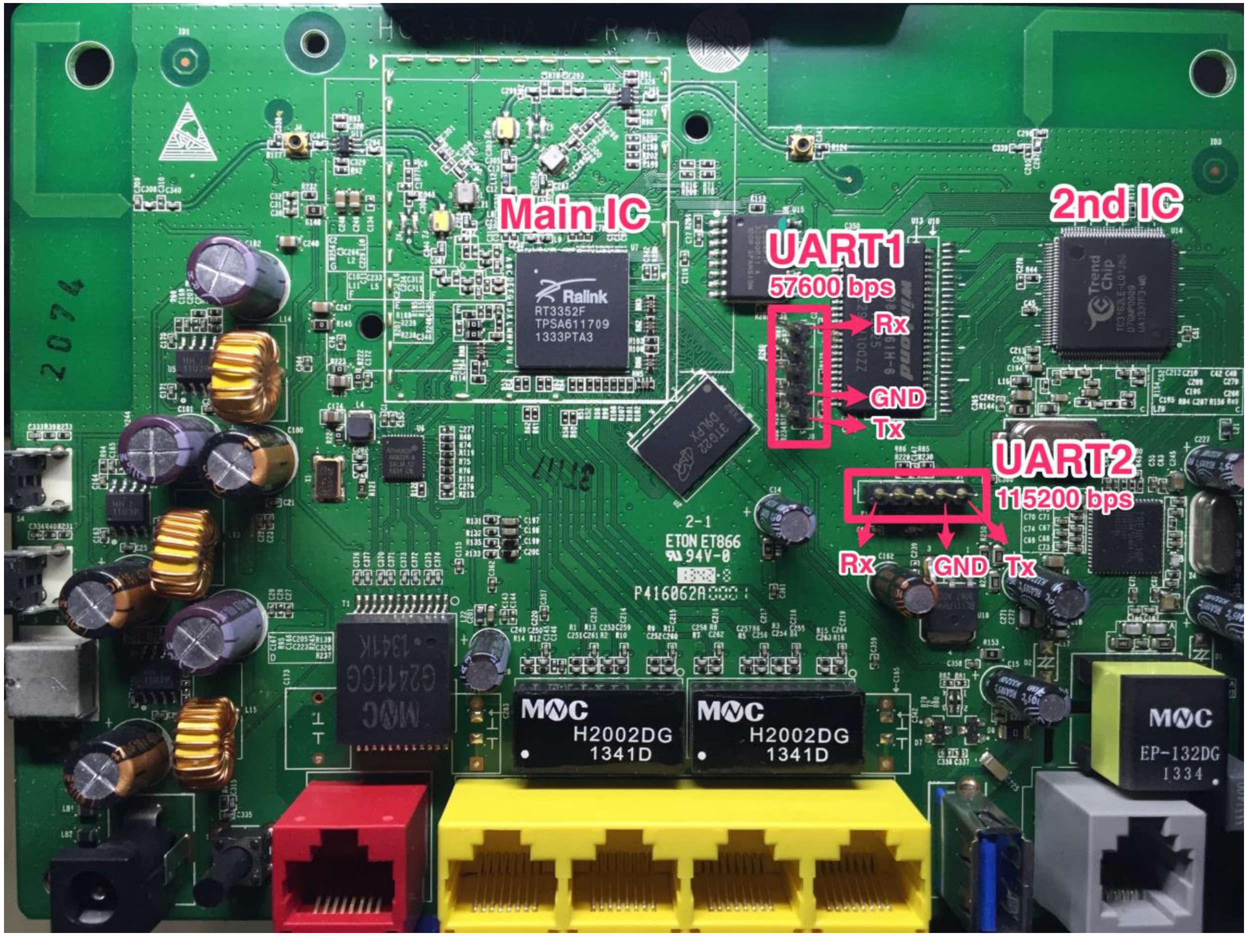
UART Hunting – Factory Test Pins



UART Hunting – Example (Not Mine)



UART Hunting – Example (Not Mine)

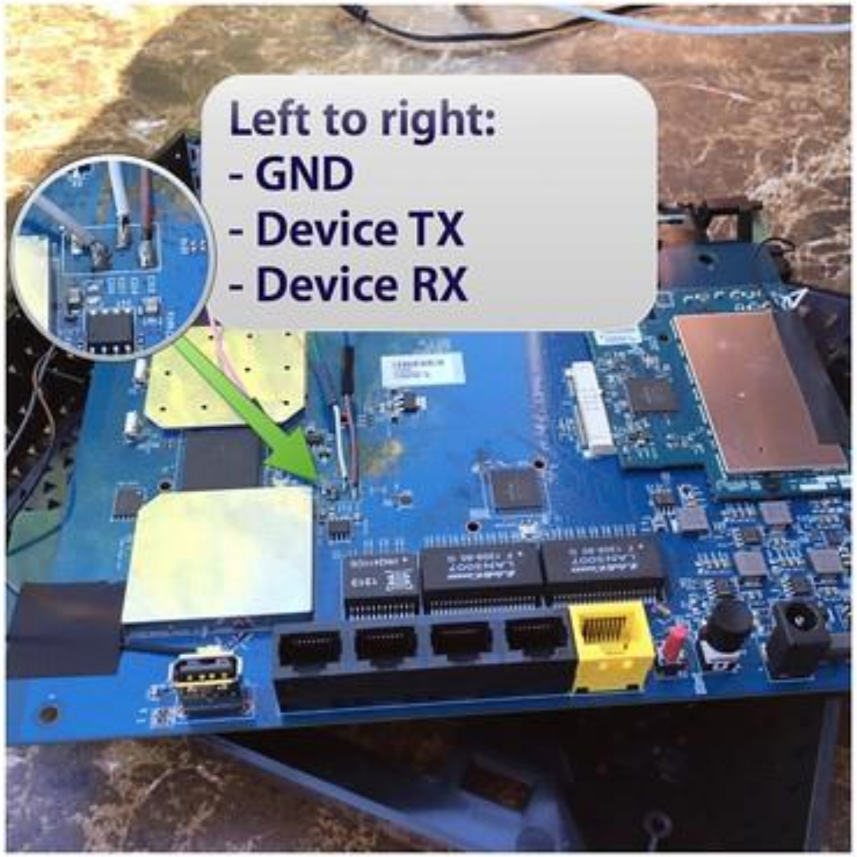


Google the Answer (Work Backwards)

Work Backwards From the Solution

These Were Next to Our TPs!

Pro-Tip: Hardware Hacking is a Rabbit Hole!



Left to right:
- GND
- Device TX
- Device RX

https://shadow-file.blogspot.com/2015/07/abandoned-part-10.html

Get To It!

Go Break Some Stuff! Dump Things!

Spend a Day Identifying Chips

Find Well Documented Boards, Walk-Through Tutorials

Get Comfortable with Being Uncomfortable

Takeaways

Firmware Updates are Often Partial Patches

Hardware is Not Easy. Practice!

Poke Things, Learn Tools, Get Swole

Don’t Break Your Vital Equipment (Life Lesson)

Questions?

Thank You!

https://github.com/arntsonl/Presentations