Hardware Hacking 101

BSides Munich 2019

Radek Domanski, Johannes Wagner

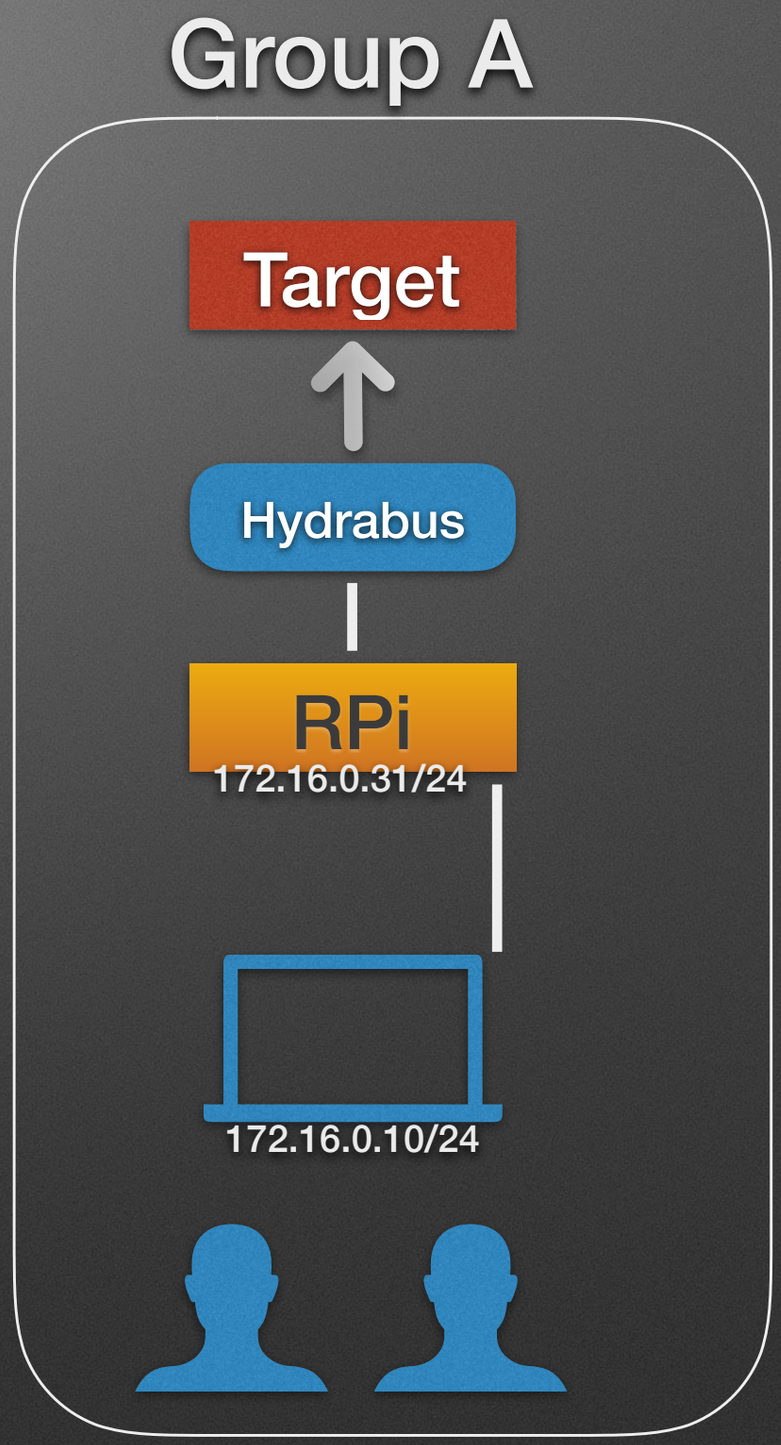
Introduction

Radek Domanski
@RabbitPro

Johannes Wagner
@ickyphuz

Course Structure

- 2 sessions - morning && afternoon
- Working in teams of 2
- 4 exercises modules
- Brake after exercise 2 and 3
- Mix hands-on vs theory



Gifts!!!

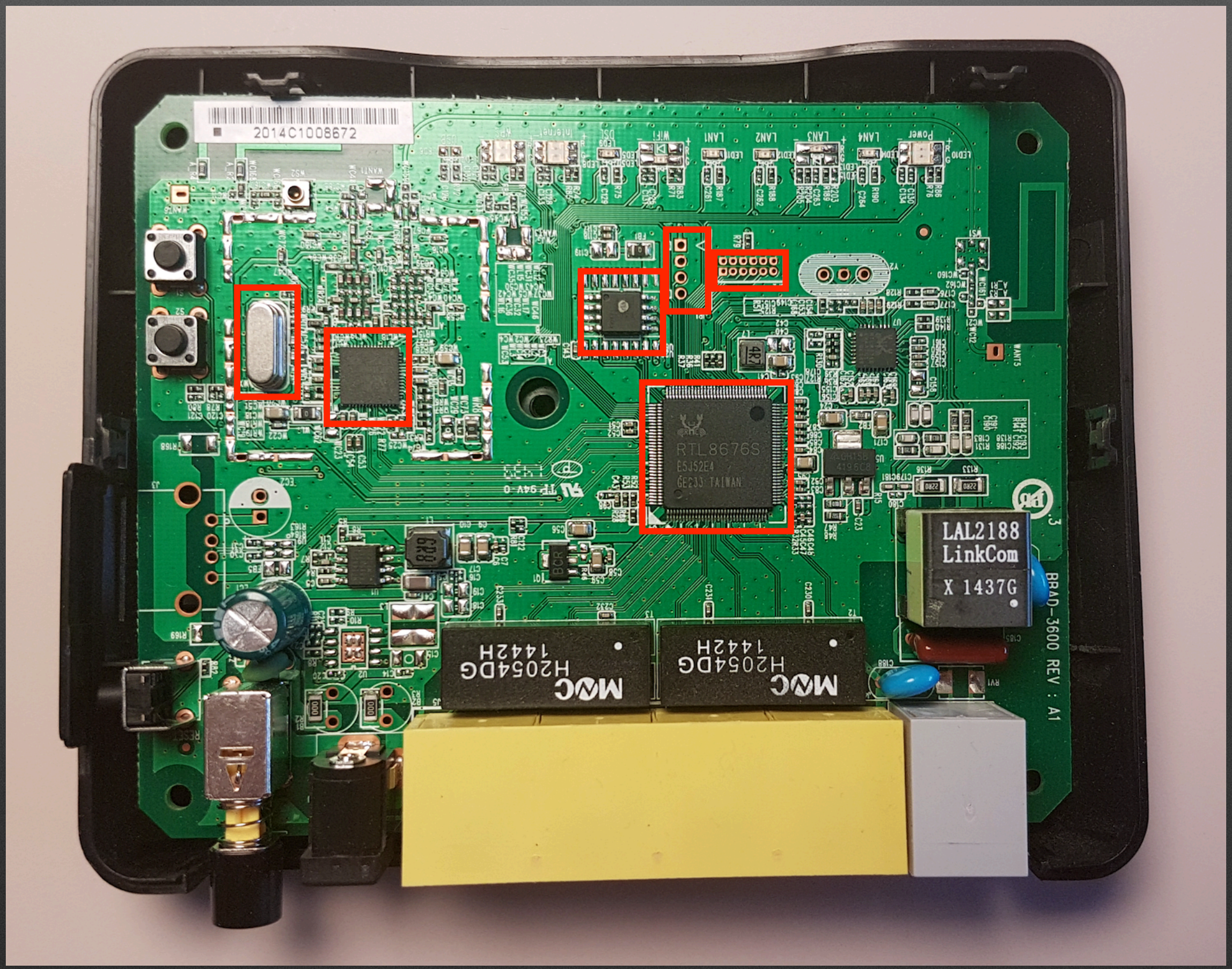
- Every participant receives a Hydrabus as a gift. Yes! You can take it home and continue hacking!
- On top of that we have 8 HydraNFC shields that we will distribute in a raffle. 4 pieces per session.

- Thank you to the gift sponsors:

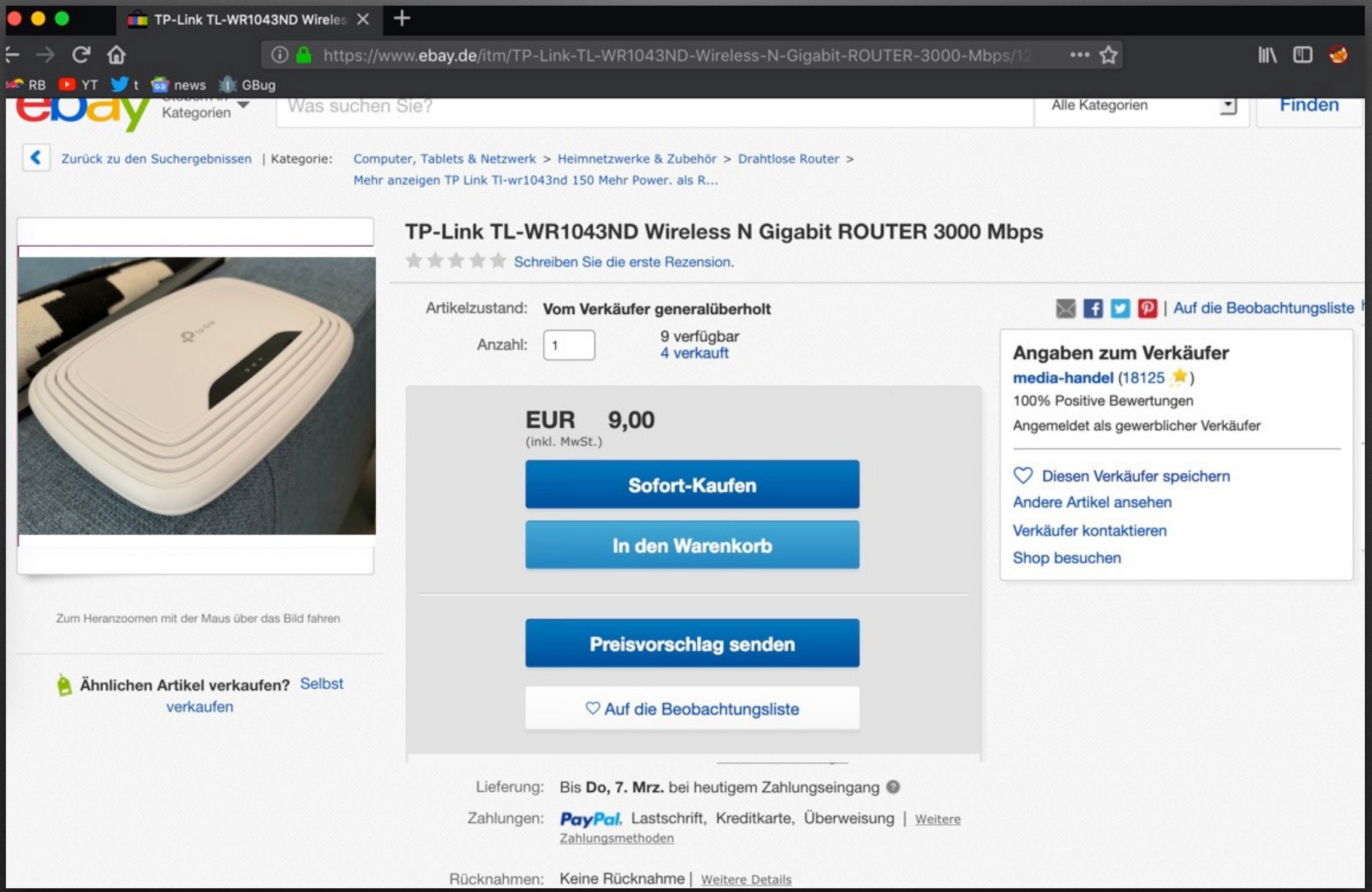
What is hardware hacking?

- Locating debug interfaces
- Dumping firmware
- Glitching
- PCB/Hardware reverse engineering
- Bypassing security restrictions by modifying hardware
- Hardware implants

Inside a Router



Scenario


TP-Link TL-WR1043ND Wireles
https://www.ebay.de/itm/TP-Link-TL-WR1043ND-Wireless-N-Gigabit-ROUTER-3000-Mbps/12
RB YT t news GBug
Kategorien
Was suchen Sie?
Alle Kategorien
Finden
Zurück zu den Suchergebnissen | Kategorie: Computer, Tablets & Netzwerk > Heimnetzwerke & Zubehör > Drahtlose Router >
Mehr anzeigen TP Link Tl-wr1043nd 150 Mehr Power. als R...
TP-Link TL-WR1043ND Wireless N Gigabit ROUTER 3000 Mbps
Schreiben Sie die erste Rezension.
Artikelzustand: Vom Verkäufer generalüberholt
Anzahl: 1
9 verfügbar
4 verkauft
Auf die Beobachtungsliste
EUR 9,00
(inkl. MwSt.)
Sofort-Kaufen
In den Warenkorb
Preisvorschlag senden
Auf die Beobachtungsliste
Angaben zum Verkäufer
media-handel (18125)
100% Positive Bewertungen
Angemeldet als gewerblicher Verkäufer
Diesen Verkäufer speichern
Andere Artikel ansehen
Verkäufer kontaktieren
Shop besuchen
Zum Heranzoomen mit der Maus über das Bild fahren
Ähnlichen Artikel verkaufen? Selbst verkaufen
Lieferung: Bis Do, 7. Mrz. bei heutigem Zahlungseingang
Zahlungen: PayPal, Lastschrift, Kreditkarte, Überweisung | Weitere Zahlungsmethoden
Rücknahmen: Keine Rücknahme | Weitere Details

Exercise 1: Inspect the Router

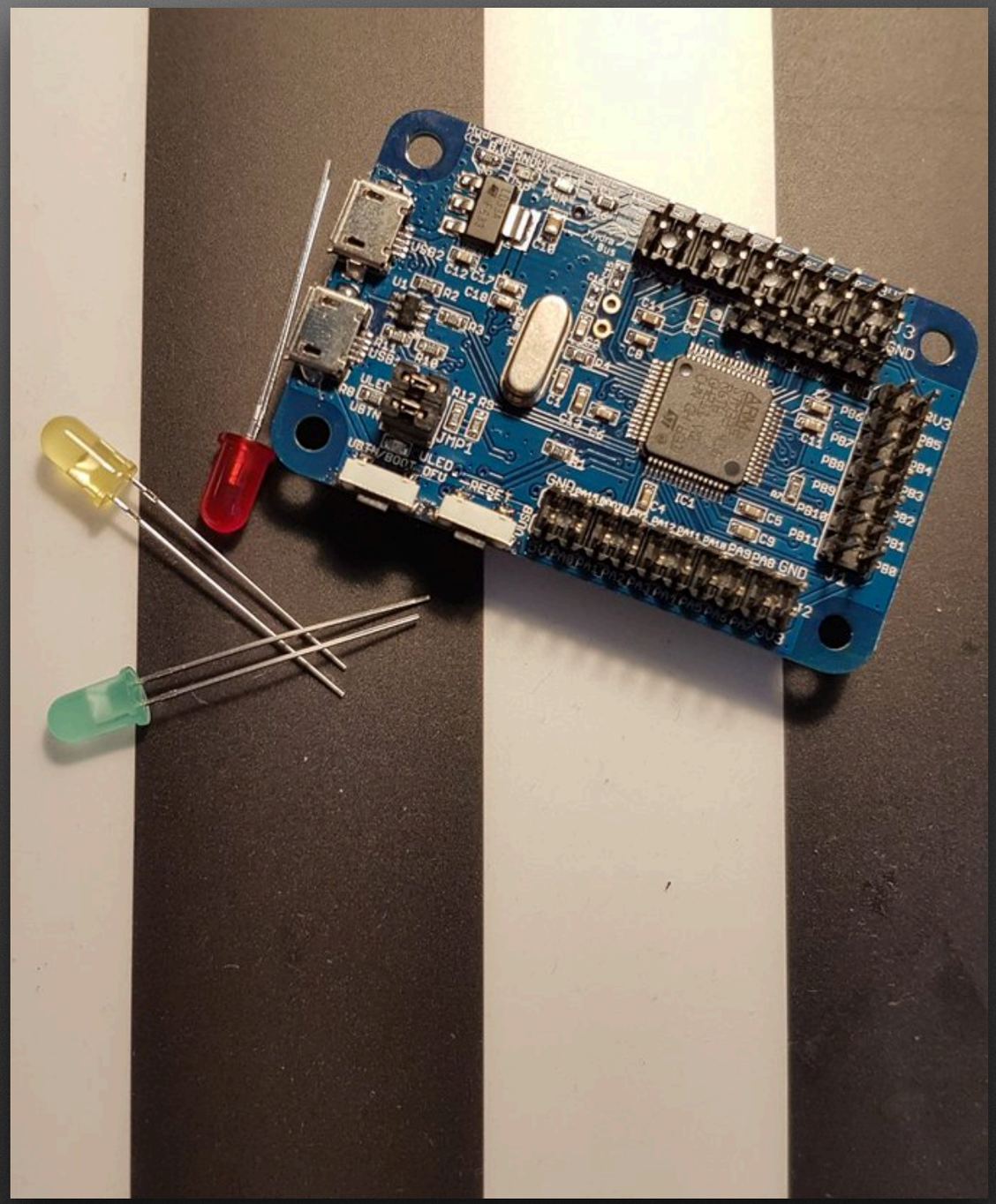
Exercise 2:

Exercise 3:

Exercise 4:

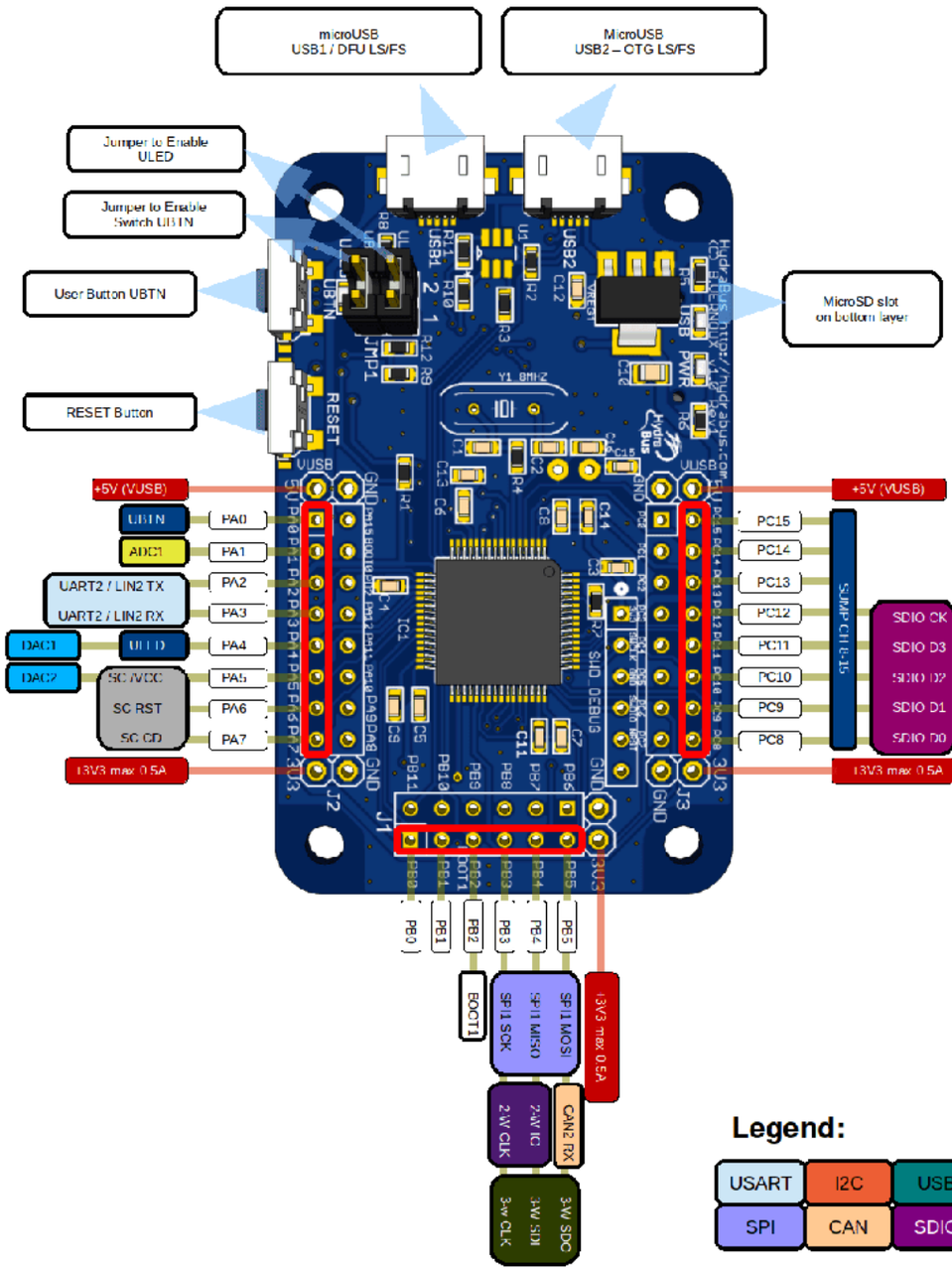
Hydrabus

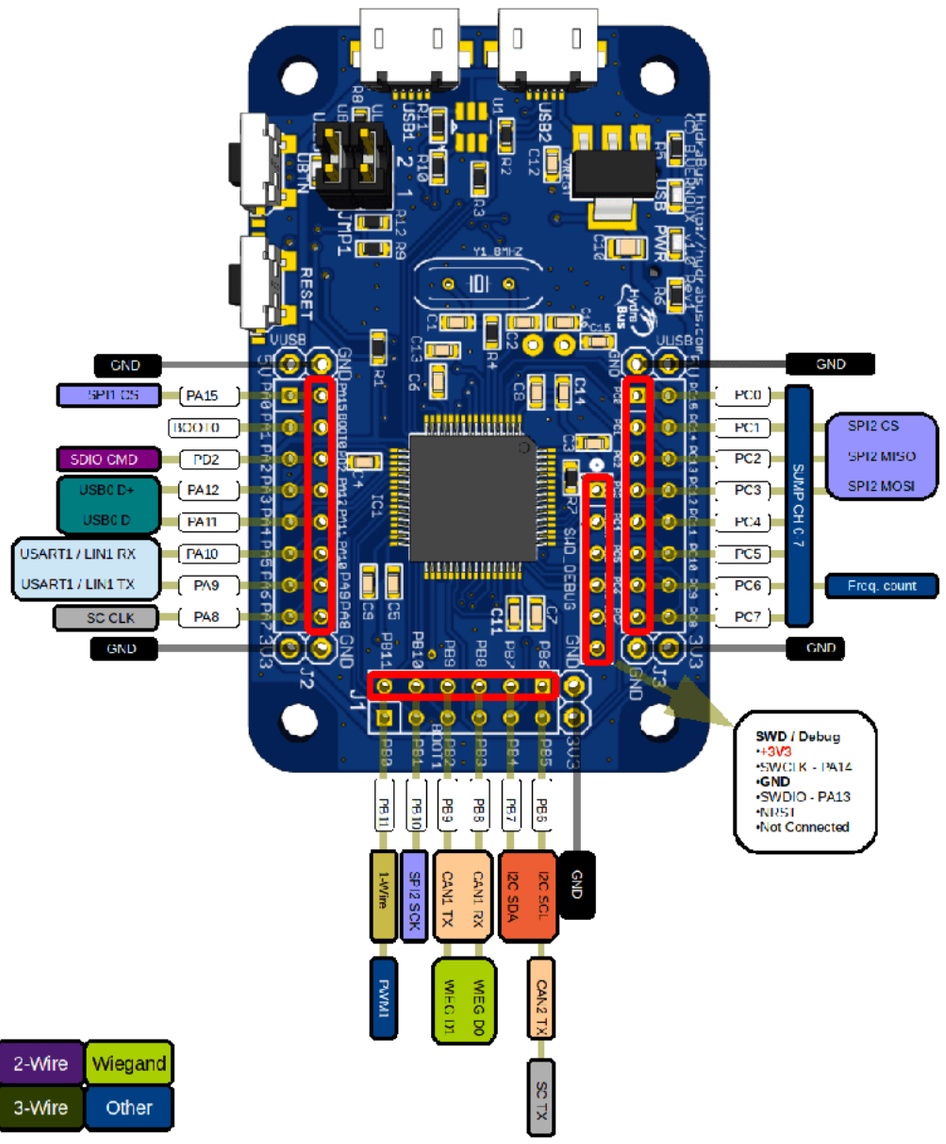
- OpenSource multi-tool hardware
- STM32F415 32bits@168MHz (40x faster than an Arduino)
- Over 15 protocols supported (JTAG, NAND, SPI, CAN, UART, I2C, etc.)
- menu mode and binary mode
- Support for NFC with HydraNFC
- website: https://hydrabus.com/



Hydrabus

Hydra Bus Hydra FW Pin Assignment





Legend:

USART	I2C	USB	ADC	Smartcard	2-Wire	Wiegand
SPI	CAN	SDIO	DAC	1-Wire	3-Wire	Other

Hydrabus

(how to connect)

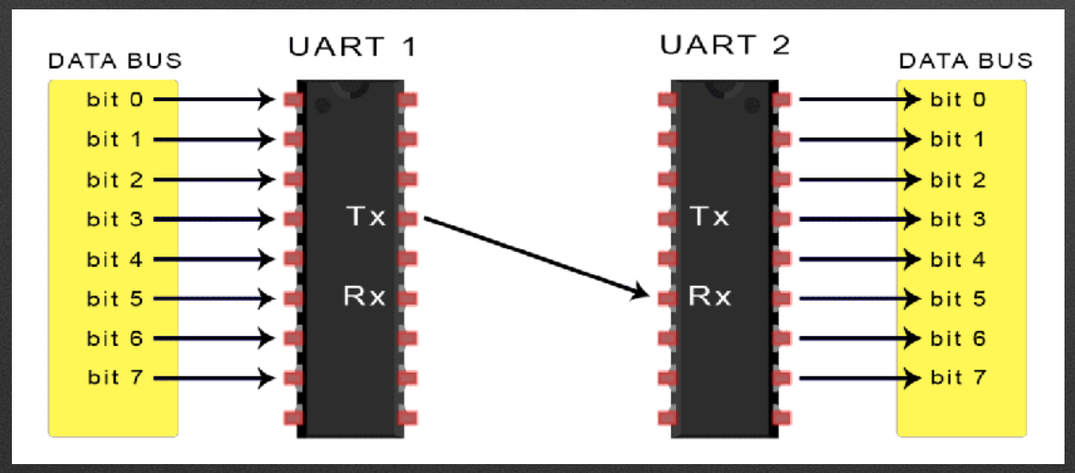
- screen /dev/tty<> 115200
- Sometimes it is required to reset the device by a RESET button

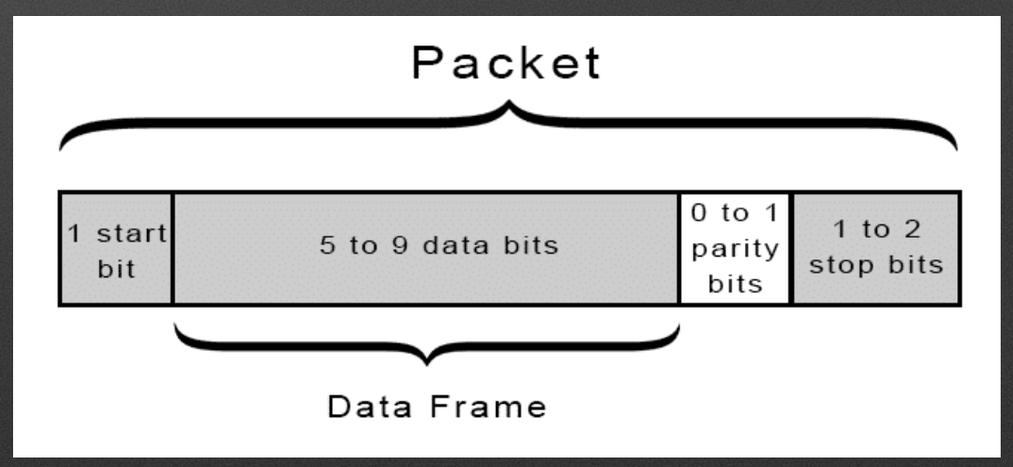
```
> help
Available commands
   help            Available commands
   history         Command history
   clear           Clear screen
   show            Show information
   logging         Turn logging on or off
   sd              SD card management
   adc             Read analog values
   dac             Write analog values
   pwm             Write PWM
   frequency       Read frequency
   gpio            Get or set GPIO pins
   spi             SPI mode
   i2c             I2C mode
   1-wire          1-wire mode
   2-wire          2-wire mode
   3-wire          3-wire mode
   uart            UART mode
   nfc             NFC mode
   can             CAN mode
   sump            SUMP mode
   jtag            JTAG mode
   random          Random number
   flash           NAND flash mode
   debug           Debug mode
>
```

UART

Universal Asynchronous Receiver/Transmiter

- Serial communication
- Synchronized by software
- No clock signal is needed only RX,TX, VCC (power) and GND (ground)
- On the RX pin you should see fluctuating values around 3,14 V - 3,3 V with the multimeter (incoming data)
- TX pin should have 0 V and no connection to GND
- You can also use a logic analyzer but multimeter is way faster
- WARNING! Some UART connectors use 5V instead of 3,3 for High value. In that case, you need a so called „logic level converter“ which basically converts the 5V to a 3,3V signal or you may brick your hydrabus/buspirate/shikra/etc. . But the most common ones are using 3,3 V



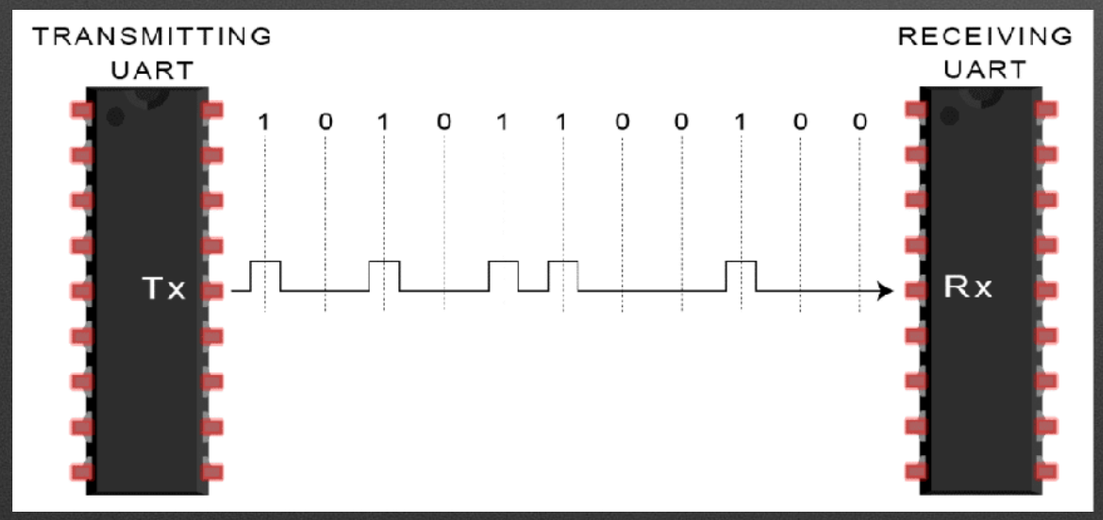


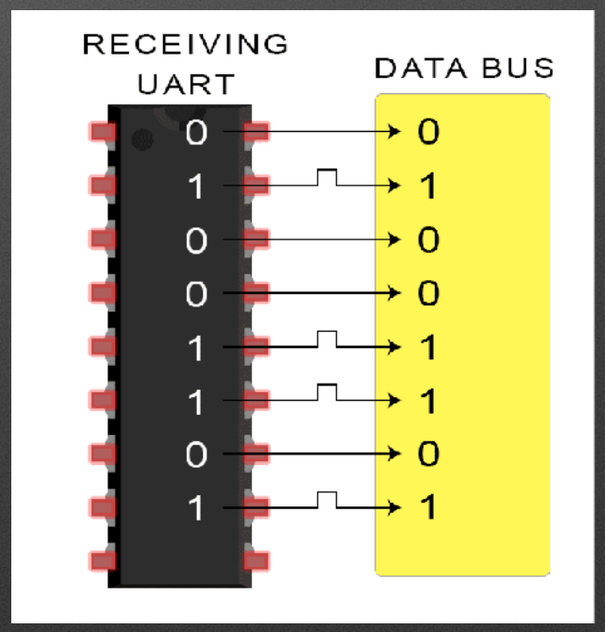
src: http://www.circuitbasics.com/basics-uart-communication/

UART

Universal Asynchronous Receiver/Transmiter

- Data frame is limited to 9 bit
- Baudrate has to be selected correctly otherwise encoding errors occur





src: http://www.circuitbasics.com/basics-uart-communication/

Multimeter

don’t brick it!

- 2 modes we need today
  - Conductivity test (the beep thingy)
    - One test-pin to ground (enclosure metal) and one test-pin to the board pin
    - It beeps when there is a connection
  - Voltage measurement
    - One test-pin to ground pin and one to the board pin you want to test

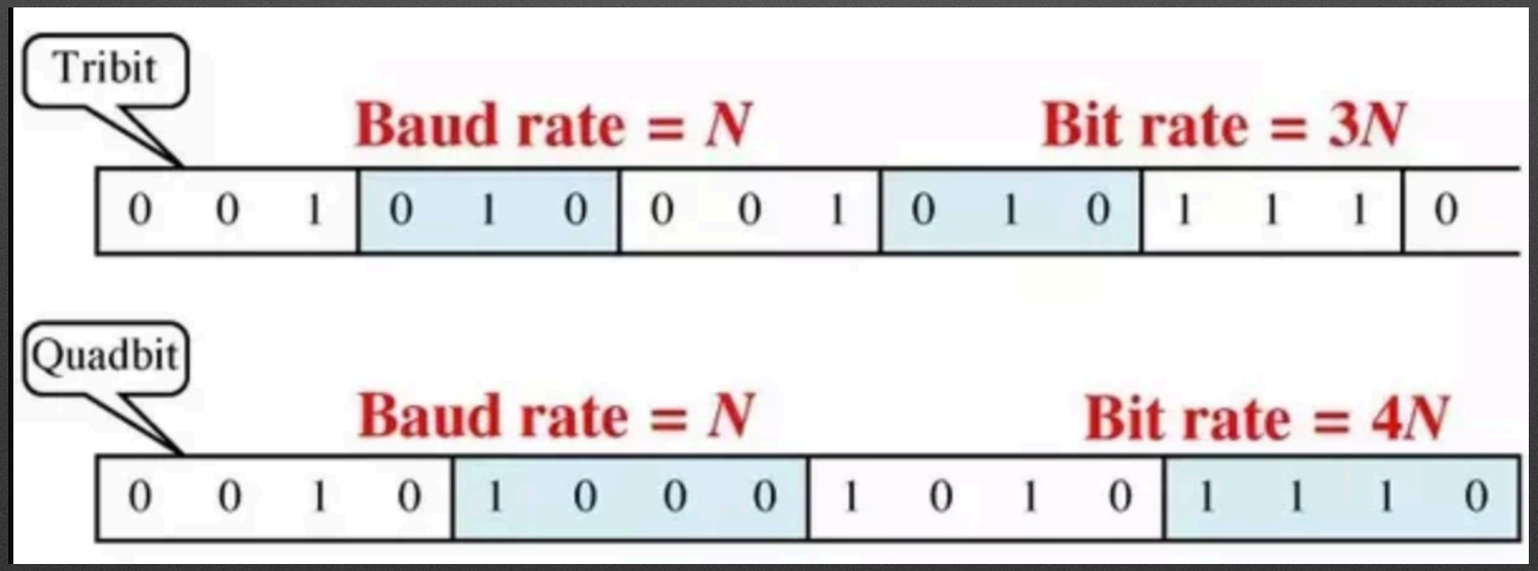
Conductivity

Voltage (DC)

Baudrate
What is it?

- It means basically how many symbol changes (not bits) per second on the transmission medium are possible. In other words, the speed.

- Q: Why do we need it? A: With the wrong baud, we get weird stuff on the terminal which nobody can read. Try it!



Baudrate
What is it?

- Q: How do i find the right baudrate?
- Most common is 115200 baud
- Other options but not limited by are: 9600, 19200, 38400, 57600

src: http://www.circuitbasics.com/basics-uart-communication/

Exercise 1: Inspect the Router

Exercise 2: UART

Exercise 3:

Exercise 4:

POST- Power On Selftest

Loading the boot code from flash over SPI to the RAM

UBOOT loads initramfs and Boots the system kernel

Partitions are mounted

SquashFS mounted

NVRAM loads user specific Configuration files

```
[04030C0C][04030C0C][87870000][22224444][00222245]
DU Setting Cal Done
```

```
U-Boot 1.1.3 (Mar 19 2018 - 15:36:42)

Board: Ralink APSoC DRAM:  32 MB
relocate_code Pointer at: 81fc0000
flash manufacture id: c8, device id 40 16
find flash: GD25Q32B
============================================
Ralink UBoot Version: 4.3.0.0
--------------------------------------------
ASIC 7628_MP (Port5<->None)
DRAM component: 256 Mbits DDR, width 16
DRAM bus: 16 bit
Total memory: 32 MBytes
Flash component: SPI Flash
Date:Mar 19 2018  Time:15:36:42
============================================
icache: sets:512, ways:4, linesz:32 ,total:65536
dcache: sets:256, ways:4, linesz:32 ,total:32768

 ##### The CPU freq = 580 MHZ ####
 estimate memory size =32 Mbytes
RESET MT7628 PHY!!!!!!
continue to starting system.
 0
disable switch phyport...
```

```
3: System Boot system code via Flash.(0xbc010000)
do_bootm:argc=2, addr=0xbc010000
## Booting image at bc010000 ...
   Uncompressing Kernel Image ... OK
No initrd
## Transferring control to Linux (at address 8000c150) ...
## Giving linux memsize in MB, 32

Starting kernel ...
```

```
GD25Q32B(c8 40160000) (4096 Kbytes)
mtd .name = raspi, .size = 0x00400000 (4M) .erasesize = 0x00010000 (64K) .numeraseregions = 0
Creating 5 MTD partitions on "raspi":
0x000000000000-0x000000010000 : "boot"
0x000000010000-0x000000100000 : "kernel"
0x000000100000-0x0000003e0000 : "rootfs"
mtd: partition "rootfs" set to be root filesystem
0x0000003e0000-0x0000003f0000 : "config"
0x0000003f0000-0x000000400000 : "radio"
Register flash device:flash0
```

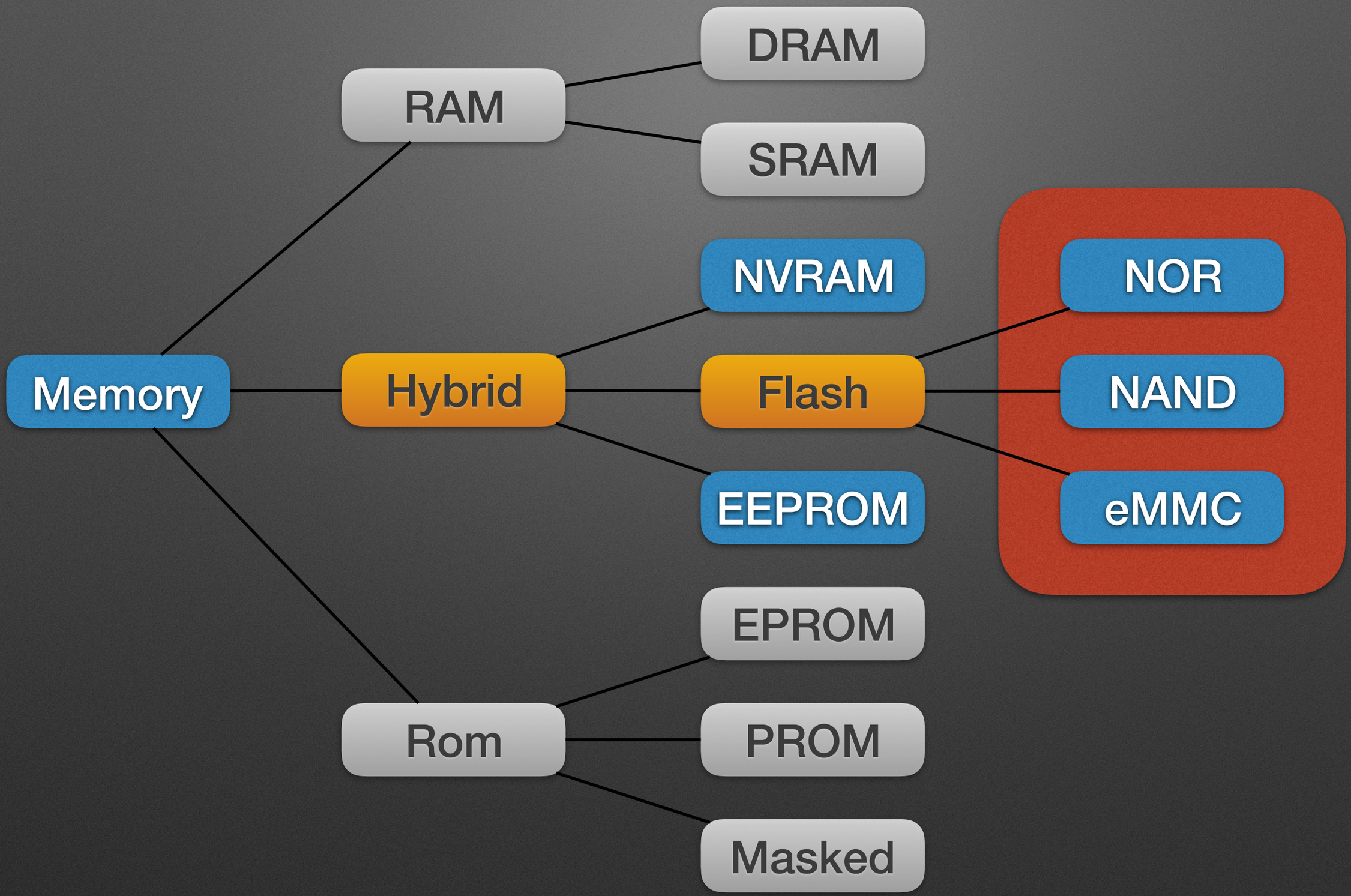
```
FS: Mounted root (squashfs filesystem) readonly on device 31:2.
```

```
spiflash_ioctl_read, Read from 0x003e0000 length 0x10000, ret 0, retlen 0x10000
spiflash_ioctl_read, Read from 0x003e0000 length 0x10, ret 0, retlen 0x10
 dm_loadCfg ] 2365:  software version is not match, in config, version = 0
 dm_readFile ] 2061:  can not open xml file /var/tmp/pc/default_config.xml!, about to open file /etc/default_config.xml
 parseConfigNode ] 525:  Meet unrecognized object node "PhDDNSCfg", skip the node
 parseConfigNode ] 530:  Meet unrecognized parameter node "PhDDNSCfg", skip the node
 parseConfigNode ] 525:  Meet unrecognized object node "SnmpCfg", skip the node
 parseConfigNode ] 525:  Meet unrecognized object node "ACL", skip the node
 parseConfigNode ] 530:  Meet unrecognized parameter node "ACL", skip the node
 parseConfigNode ] 530:  Meet unrecognized parameter node "MACAddressControlEnabled", skip the node
 parseConfigNode ] 530:  Meet unrecognized parameter node "X_TP_MACAddressControlRule", skip the node
 parseConfigNode ] 530:  Meet unrecognized parameter node "Vlan", skip the node
```

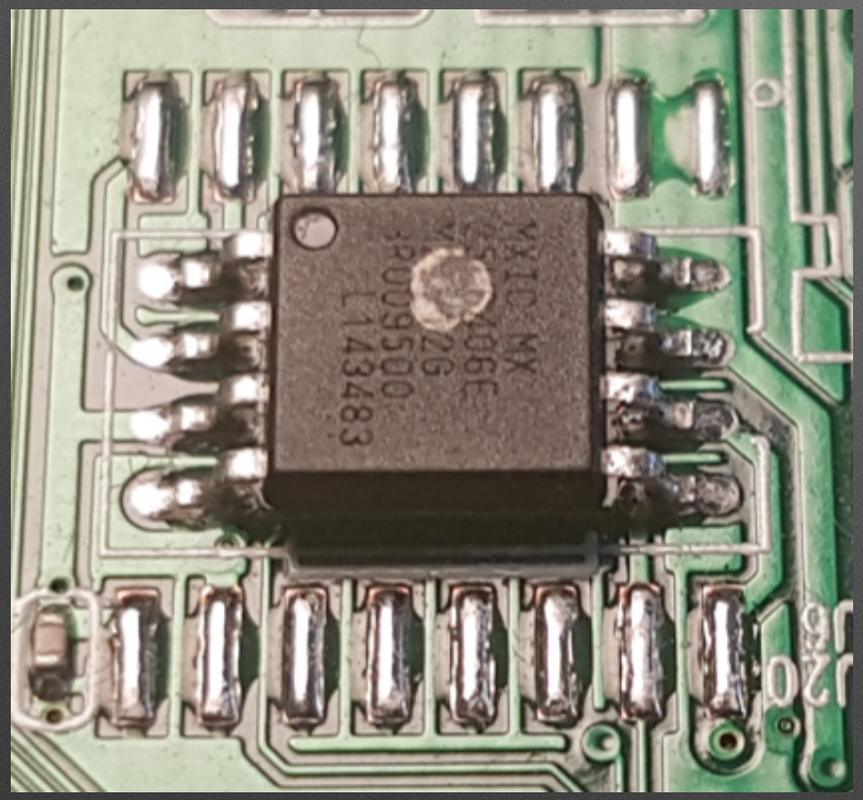
Firmware

```
flash manufacture id: c8, device id 40 16
GD25Q32B(c8 40160000) (4096 Kbytes)
mtd .name = raspi, .size = 0x00400000 (4M) .erasesize = 0x00010000 (64K) .numeraseregions = 0
Creating 5 MTD partitions on "raspi":
0x000000000000-0x000000010000 : "boot"
0x000000010000-0x000000100000 : "kernel"
0x000000100000-0x0000003e0000 : "rootfs"
mtd: partition "rootfs" set to be root filesystem
0x0000003e0000-0x0000003f0000 : "config"
0x0000003f0000-0x000000400000 : "radio"
Register flash device:flash0
PPP generic driver version 2.4.2
```

Memory types



Flash memory types

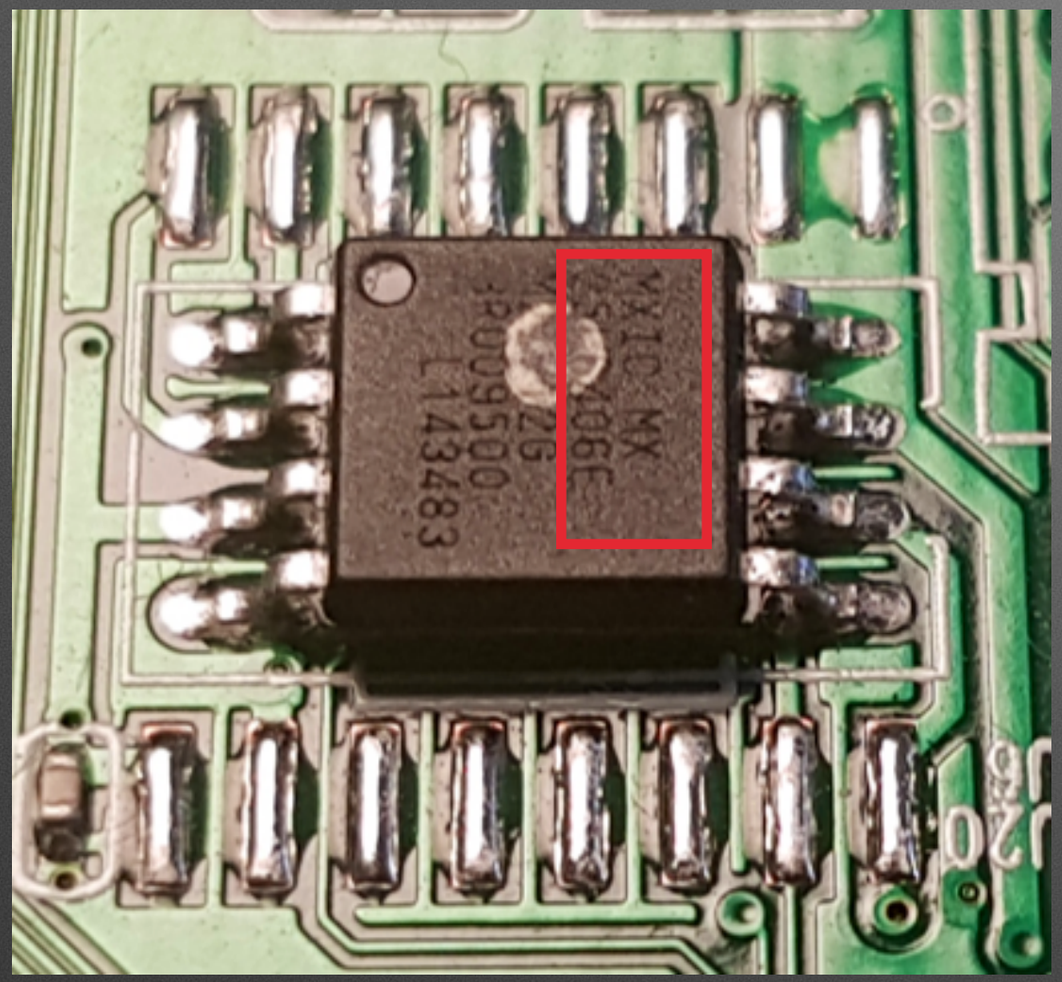


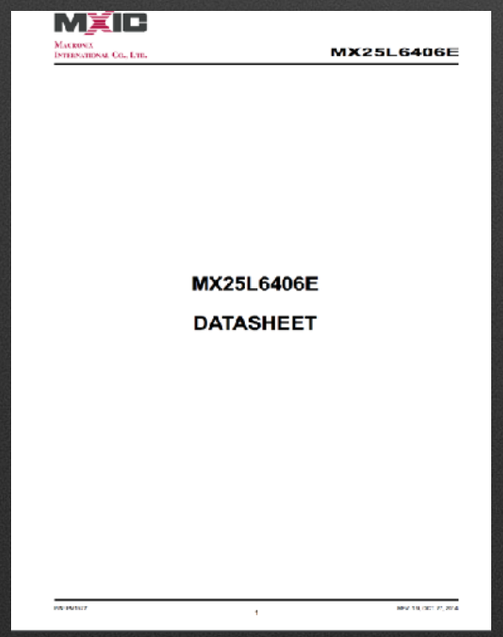
- NOR FLASH
- SOIC8 package

- NAND FLASH
- TSOP48 package

- eMMC FLASH
- BGA{153} package

NOR Flash (SPI)

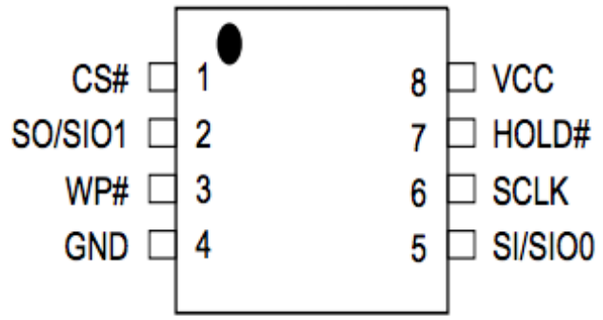



MXIC
MACRONIX INTERNATIONAL CO., LTD.
MX25L6406E

MX25L6406E

DATASHEET

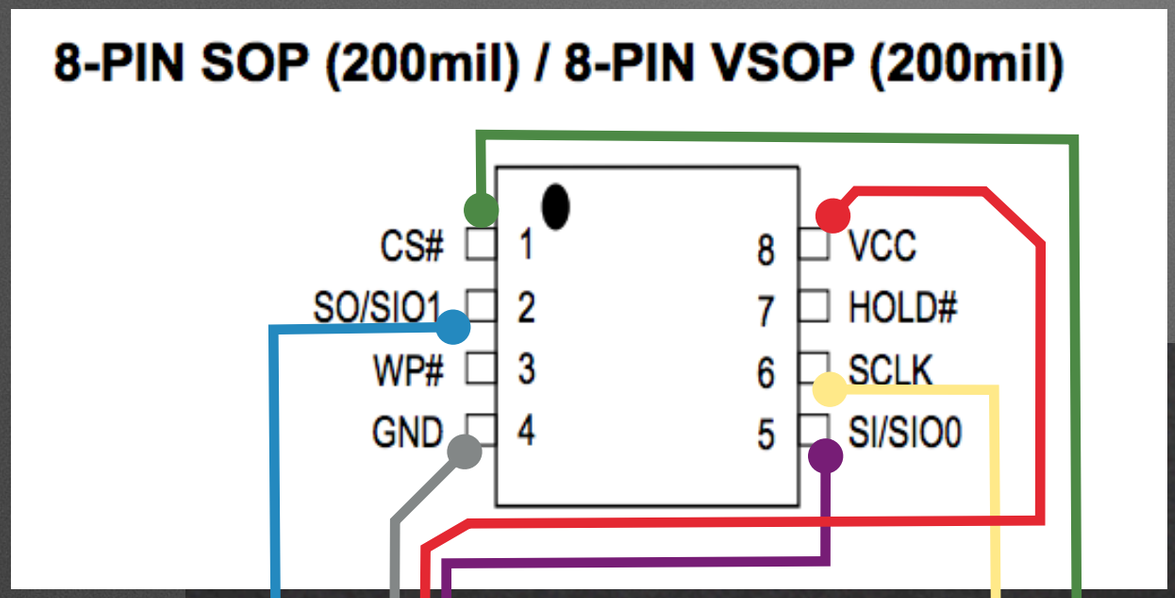
8-PIN SOP (200mil) / 8-PIN VSOP (200mil)



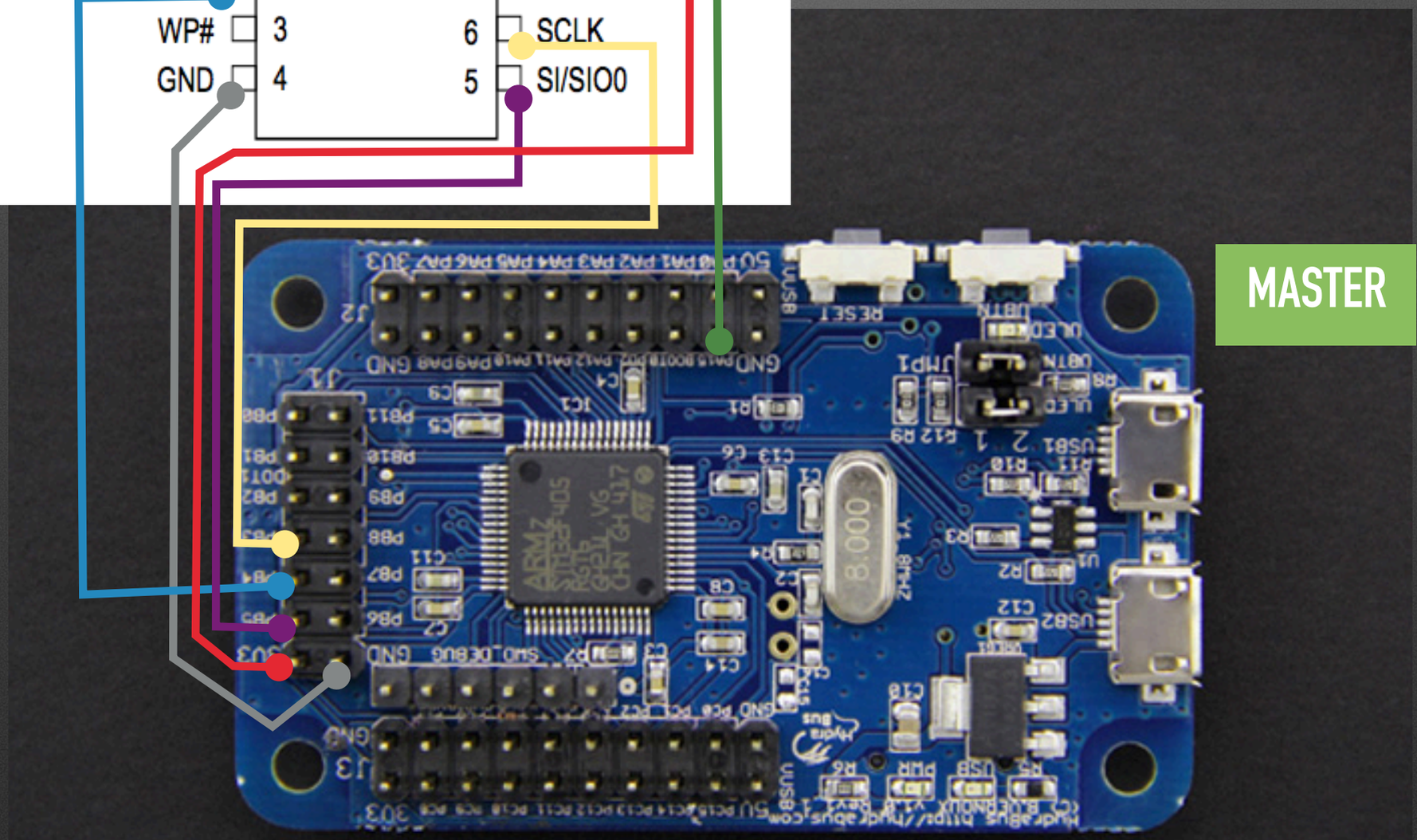
4. PIN DESCRIPTION

SYMBOL	DESCRIPTION
CS#	Chip Select
SI/SIO0	Serial Data Input (for 1 x I/O)/ Serial Data Input & Output (for Dual Output mode)
SO/SIO1	Serial Data Output (for 1 x I/O)/ Serial Data Output (for Dual Output mode)
SCLK	Clock Input
WP#	Write protection
HOLD#	Hold, to pause the device without deselecting the device
VCC	+ 3.3V Power Supply
GND	Ground

NOR to Hydrabus (SPI)

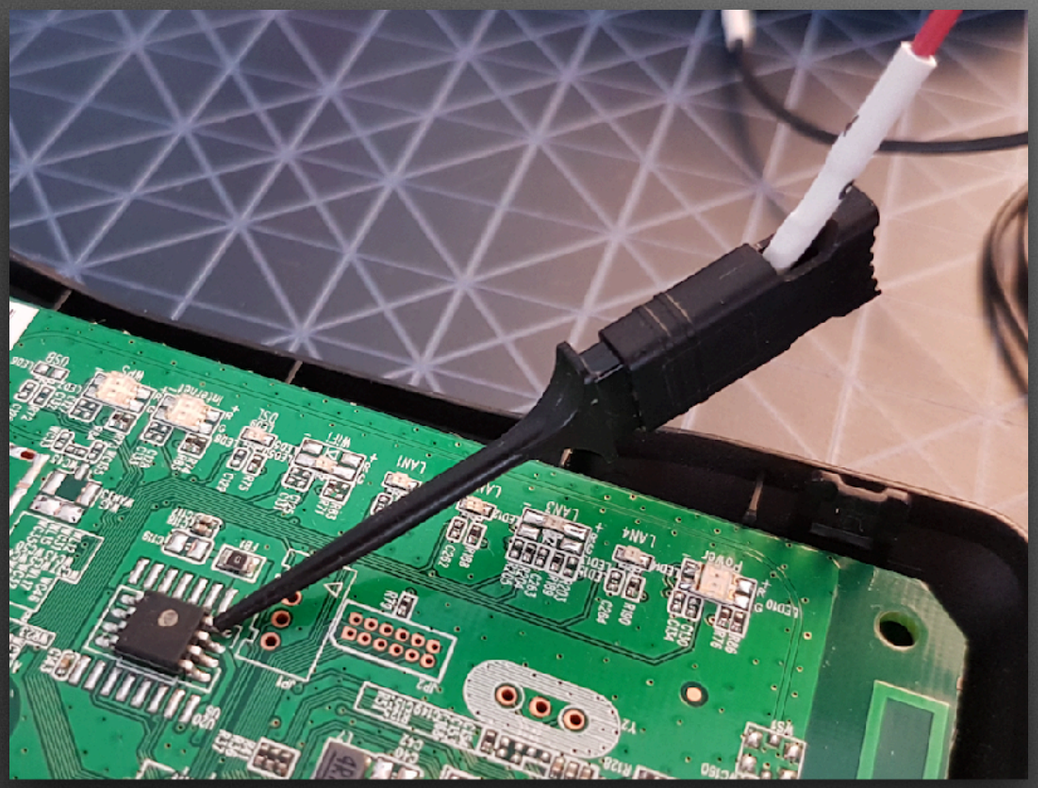


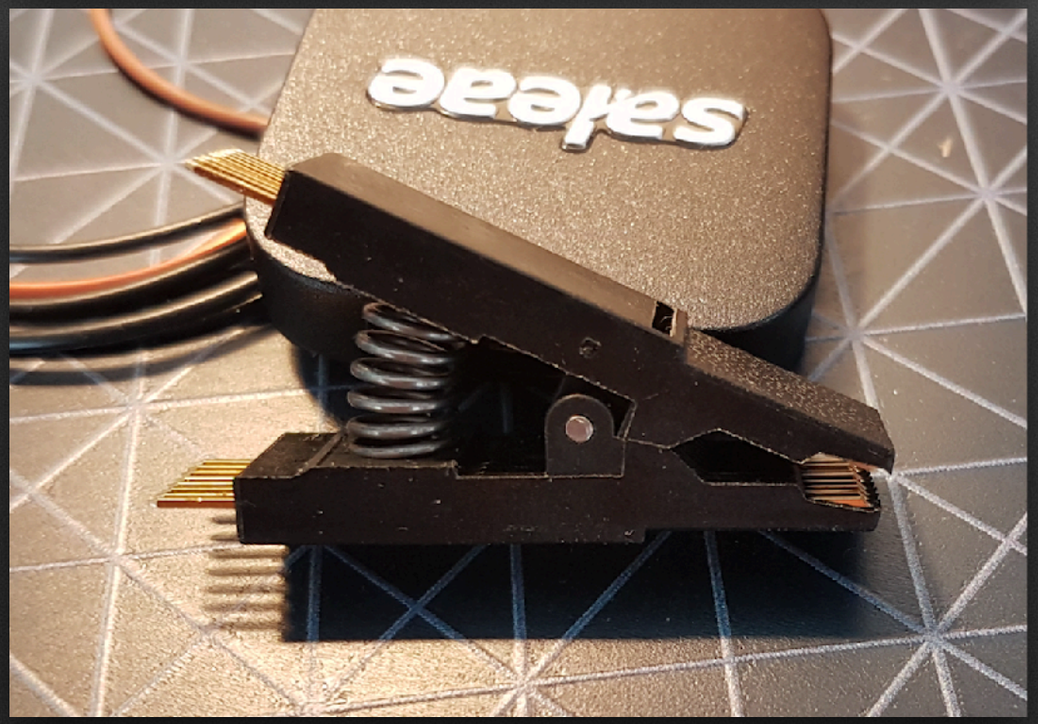
SLAVE

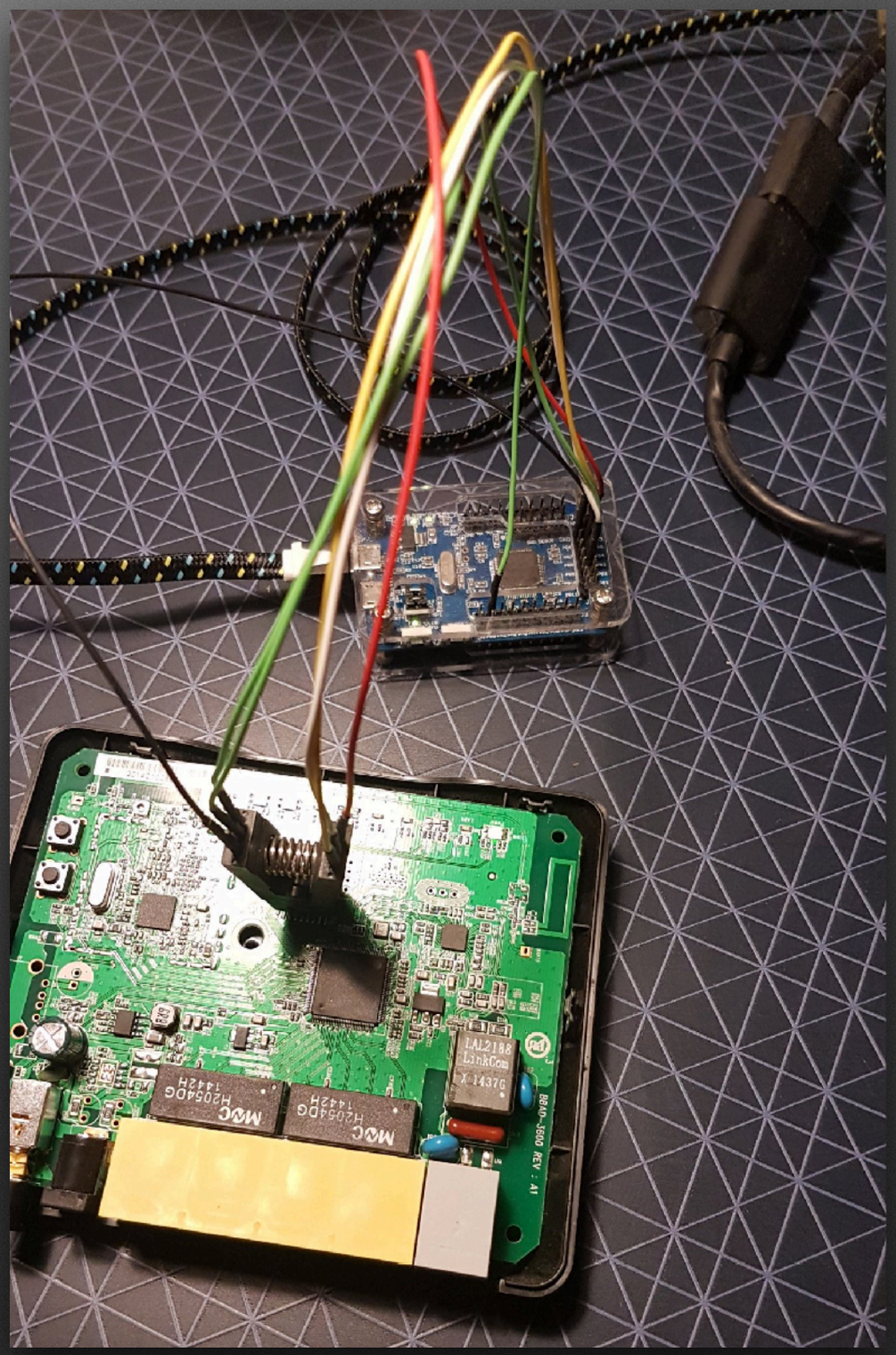


MASTER

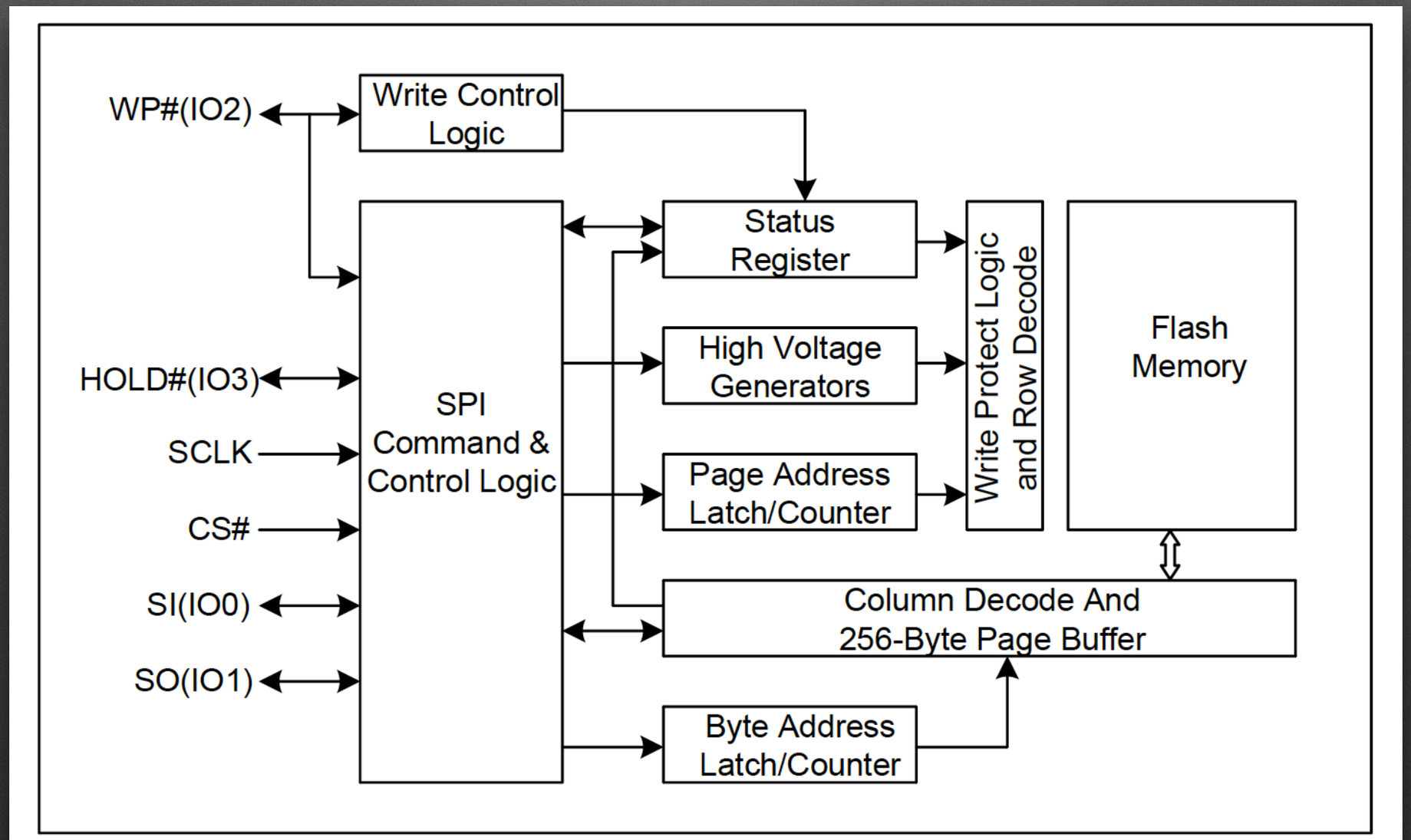
NOR to Hydrabus







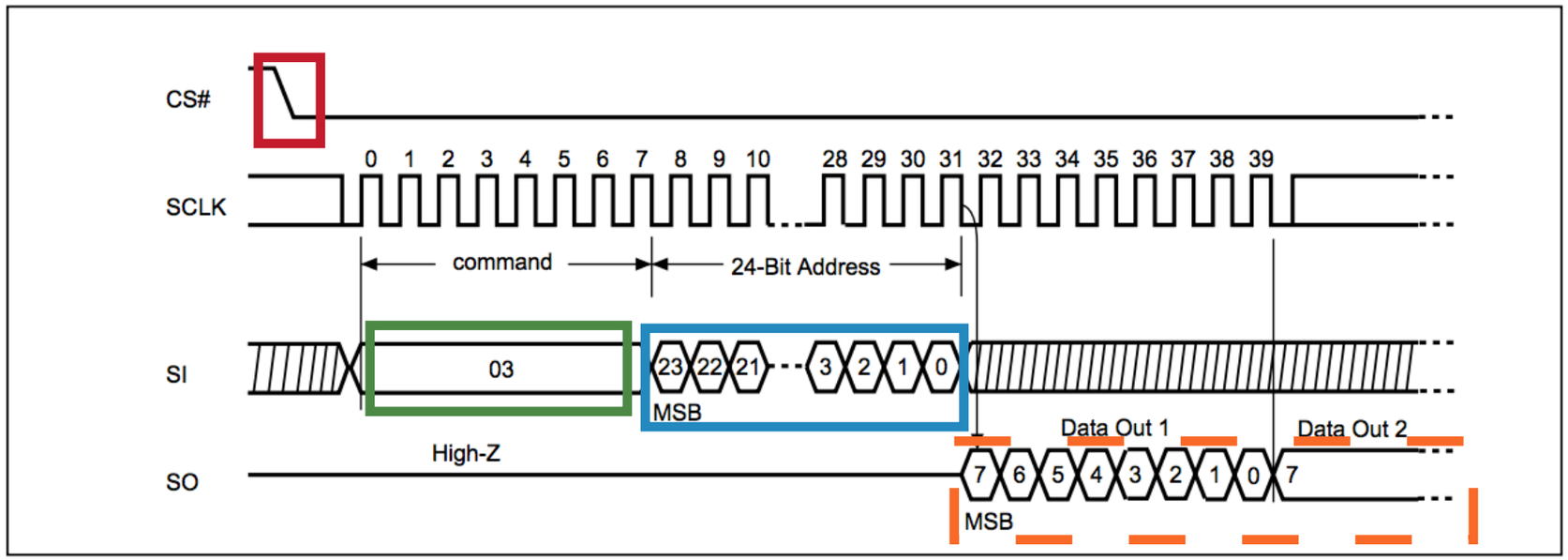
SPI Protocol



Sector Erase	20H	A23-A16	A15-A8	A7-A0
Block Erase(32K)	52H	A23-A16	A15-A8	A7-A0
Block Erase(64K)	D8H	A23-A16	A15-A8	A7-A0

Read Data from NOR Flash

Figure 16. Read Data Bytes (READ) Sequence (Command 03)




```
spi1> [ 0x03 0x00 0x00 0x00
/CS ENABLED
WRITE: 0x03 0x00 0x00 0x00
spi1> hd:128
10 00 00 FF 00 00 00 00  10 00 01 05 00 00 00 00  |  ................
00 00 00 00 00 00 00 00  10 00 FF F9 00 00 00 00  |  ................
10 00 FF F7 00 00 00 00  10 00 FF F5 00 00 00 00  |  ................
10 00 FF F3 00 00 00 00  10 00 FF F1 00 00 00 00  |  ................
10 00 FF EF 00 00 00 00  10 00 FF ED 00 00 00 00  |  ................
10 00 FF EB 00 00 00 00  10 00 FF E9 00 00 00 00  |  ................
10 00 FF E7 00 00 00 00  10 00 FF E5 00 00 00 00  |  ................
10 00 FF E3 00 00 00 00  10 00 FF E1 00 00 00 00  |  ................
spi1> ]
/CS DISABLED
spi1>
```

Exercise 1: Inspect the Router

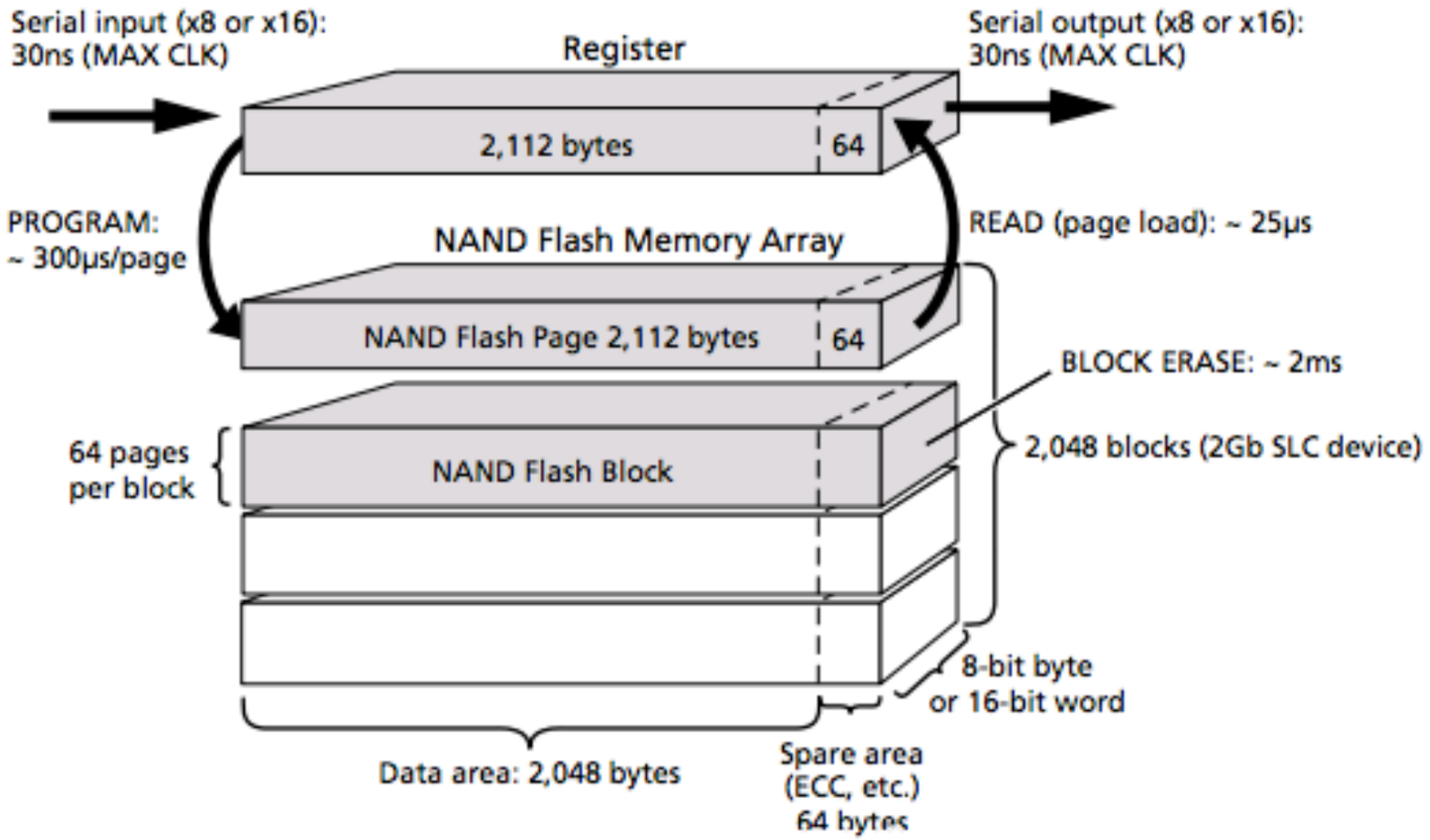
Exercise 2: UART

Exercise 3: NOR Flash & Router hacking

Exercise 4:

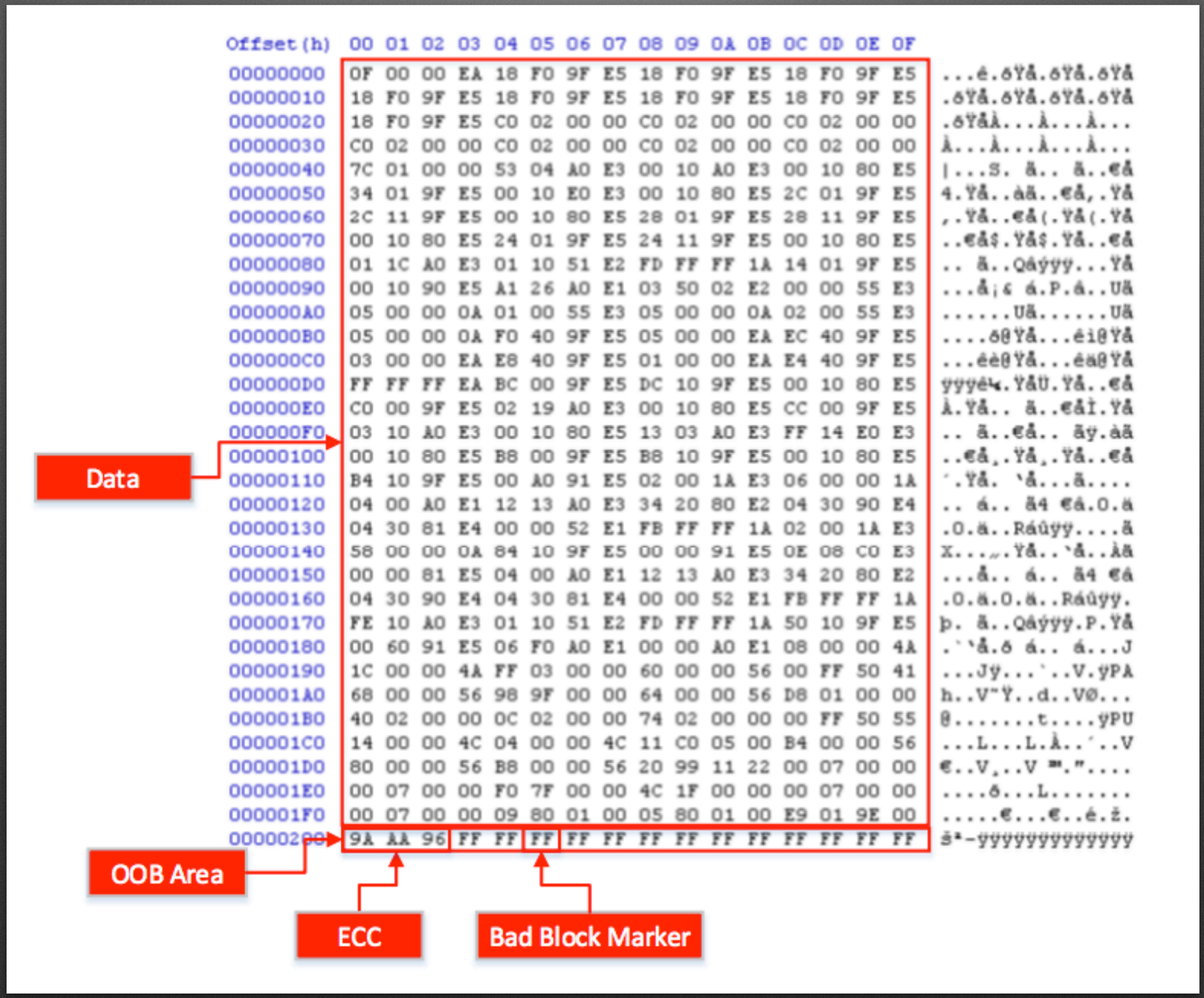
NAND Flash

2Gb NAND Flash Device Organized as 2,048 Blocks



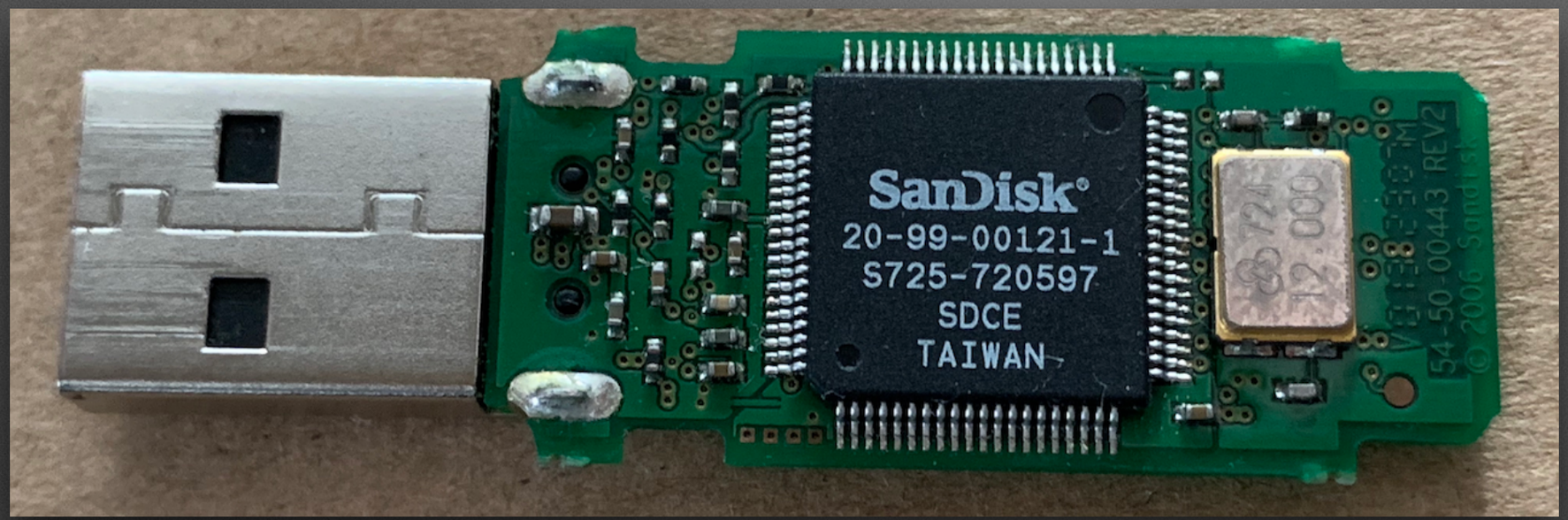
- Pages organised in blocks
- OOB and ECC introduced
- Raw access or controller access (bad block management)

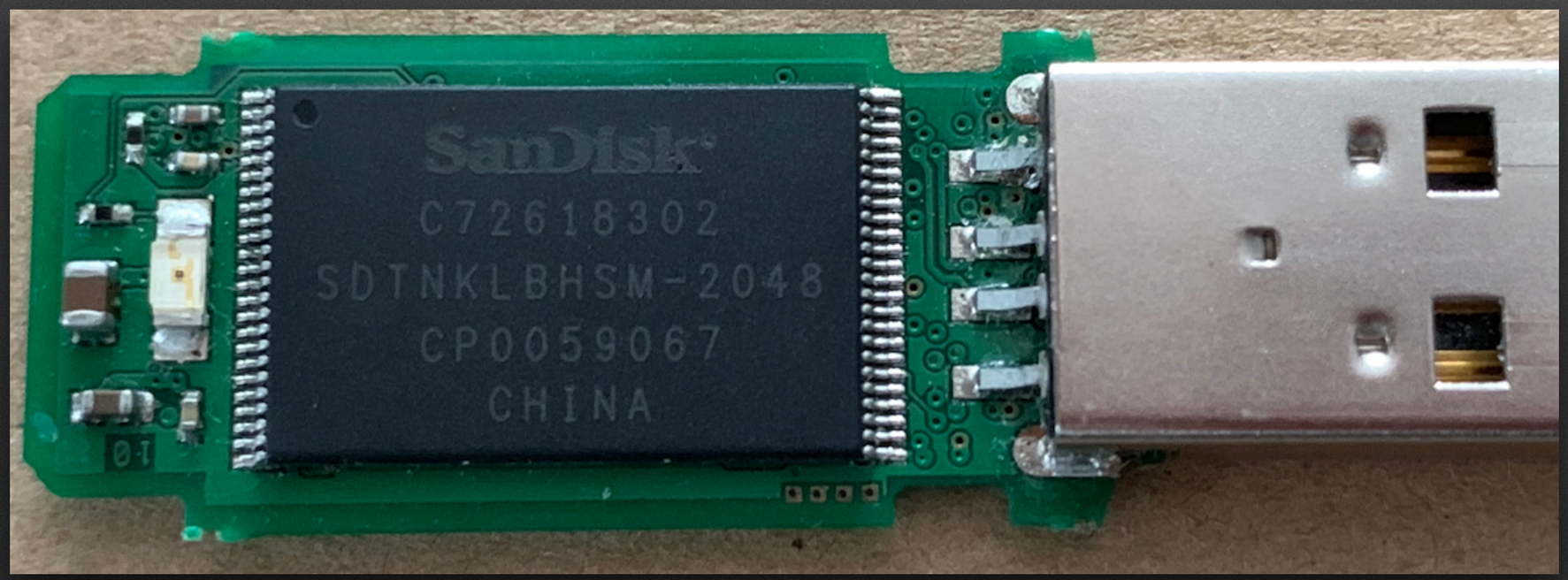
NAND Data organization

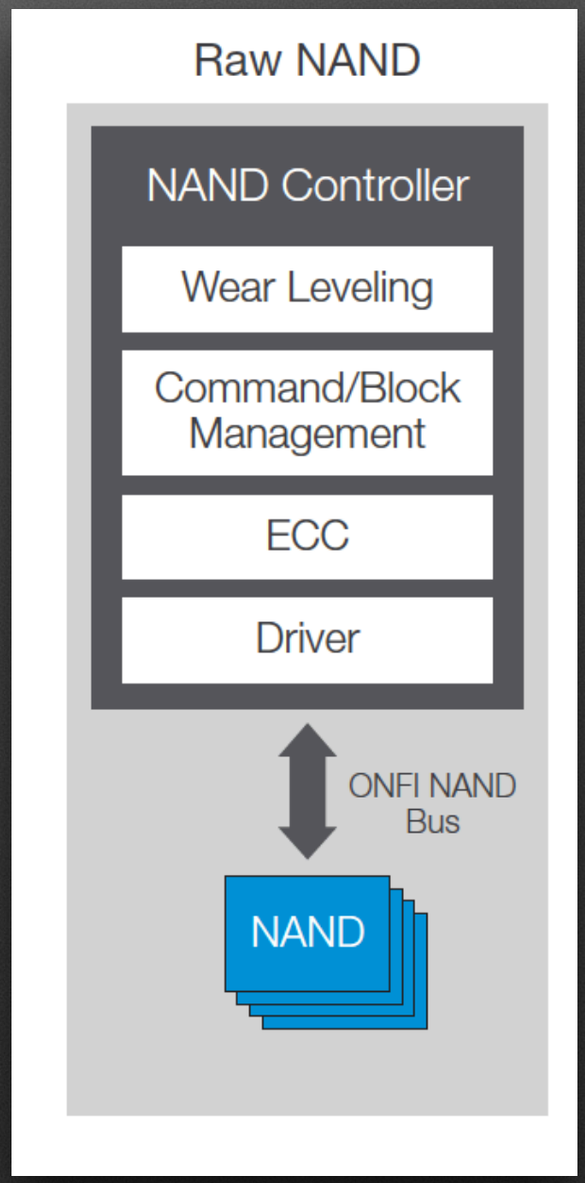


source: https://www.blackhat.com/docs/us-14/materials/us-14-Oh-Reverse-Engineering-Flash-Memory-For-Fun-And-Benefit-WP.pdf

NAND + Controller





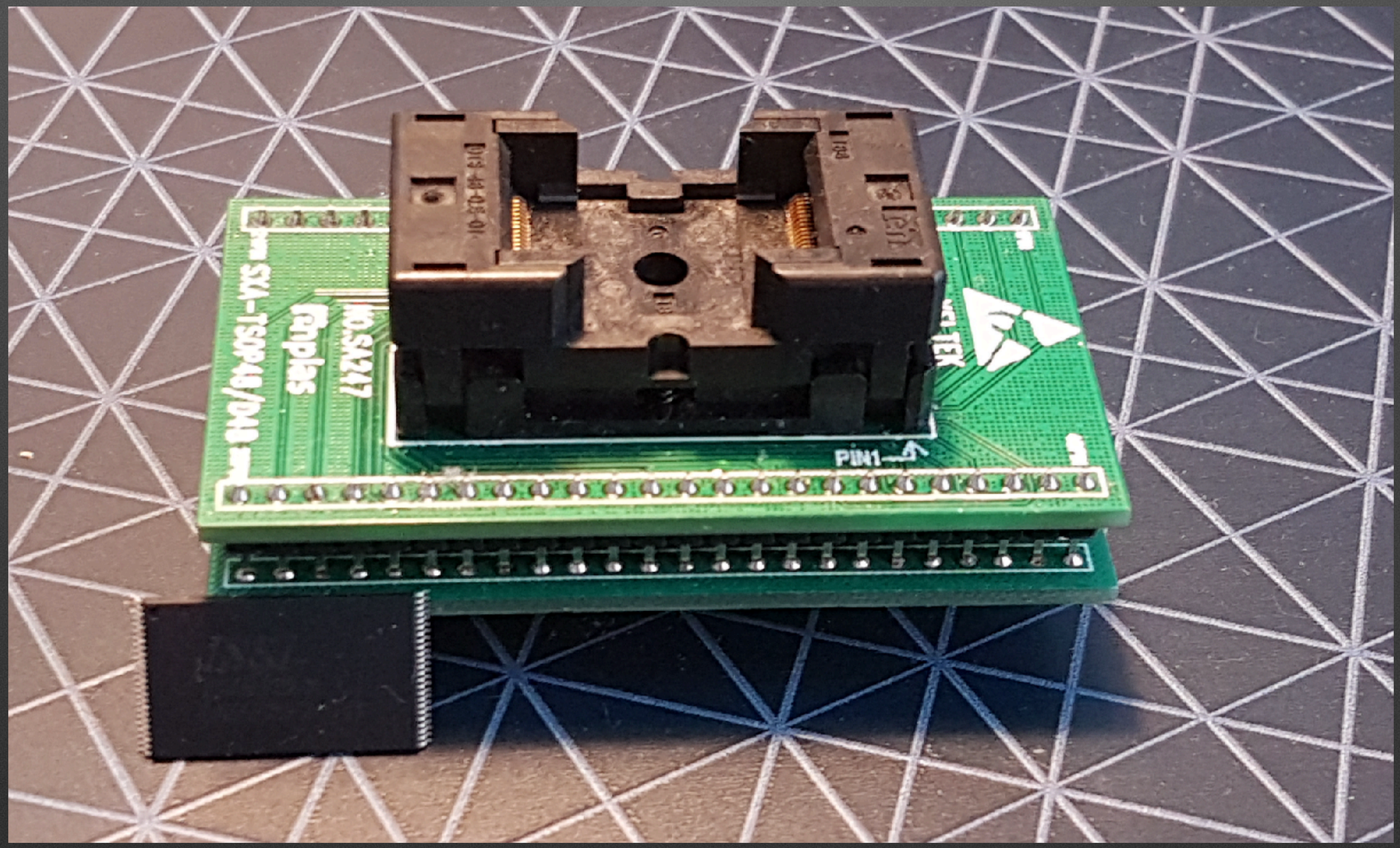


source: NOR Flash guide Micron

NAND chip extraction

https://www.youtube.com/watch?v=7VahHWI3pT8

NAND firmware extraction



Exercise 1: Inspect the Router

Exercise 2: UART

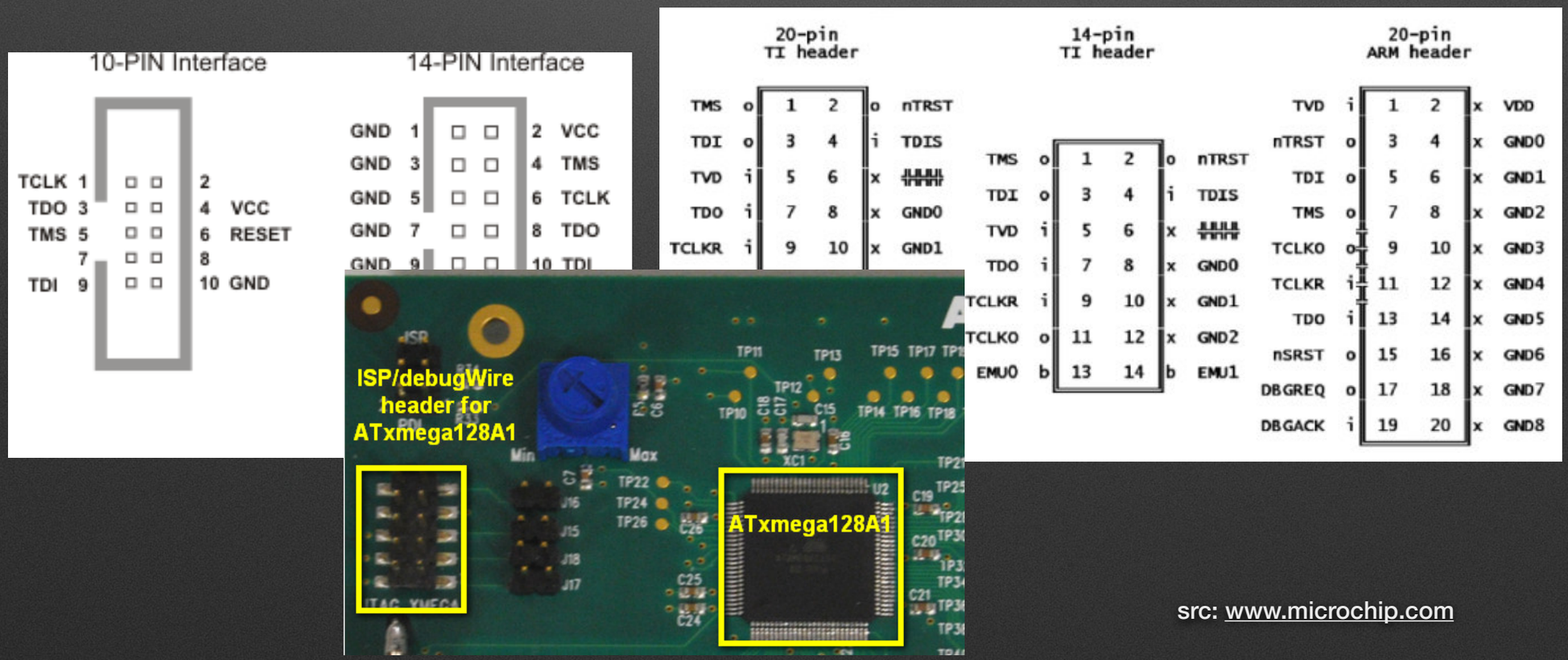
Exercise 3: NOR Flash & Router hacking

Exercise 4: NAND Flash

JTAG

Joint Test Action Group

- Debug interface for a processor
- Hardware equivalent to „root“ on the OS layer
- Comes in different pin layout schemes {10,14,20,…}-pin)

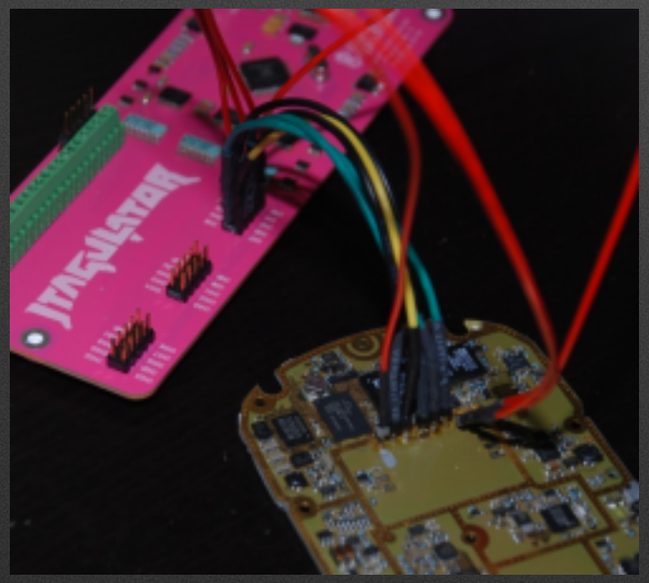


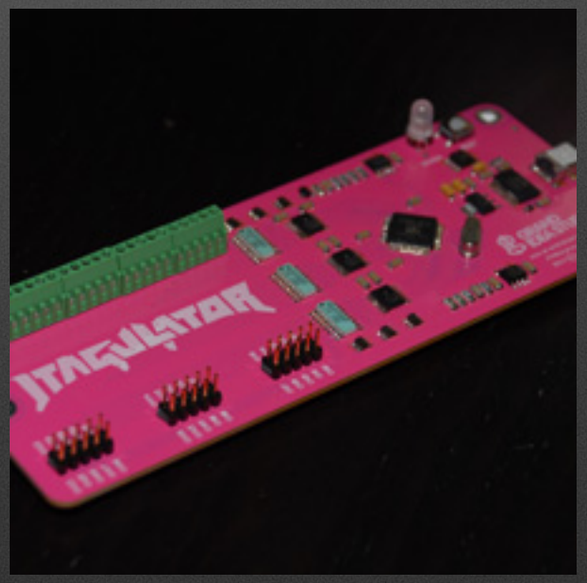
src: www.microchip.com

JTAG

Joint Test Action Group

- Finding the right pins is not trivial.
- Good indicators are labels on the PCB such as: TMS,TCK or TDI
- JTAGulator/JTAGenum/Hydrabus are devices to find the right pin layout from a lot of provided input pins
- It may be locked on your devices (!)





src: grandideastudio.com/jtagulator/

JTAG

Joint Test Action Group

- Once you found the layout successfully connect the following pins to the Hydrabus
  - TCK (Test Clock)
  - TDI (Test Data In)
  - TDO (Test Data Out)
  - TMS (Test Mode Select)
  - TRST (Test Reset) *optional*

JTAG / OpenOCD

Joint Test Action Group

- Debugging the hardware can be done with OpenOCD combined with GDB

- Starting the server and connect with GDB (example):

  - openocd -s share/openocd/scripts -f interface/ftdi/esp32_devkitj_v1.cfg -f board/esp-wroom-32.cfg

  - target remote IP:PORT i.e. target remote localhost:4444

after connecting with the correct configuration it is recomended to set the adapters
clock speed, enable single core debugging etc. a great guide for the configuration can be found on openocd.org
(http://www.openocd.org/doc/html/Config-File-Guidelines.html) and a complete guideline with openocd features etc.
can be found at (https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/jtag-debugging/index.html#jtag-debugging-configuring-esp32-target).

THANK YOU!

RADEK DOMANSKI

TWITTER: @RABBITPRO

JOHANNES WAGNER

TWITTER: @ICKYPHUZ