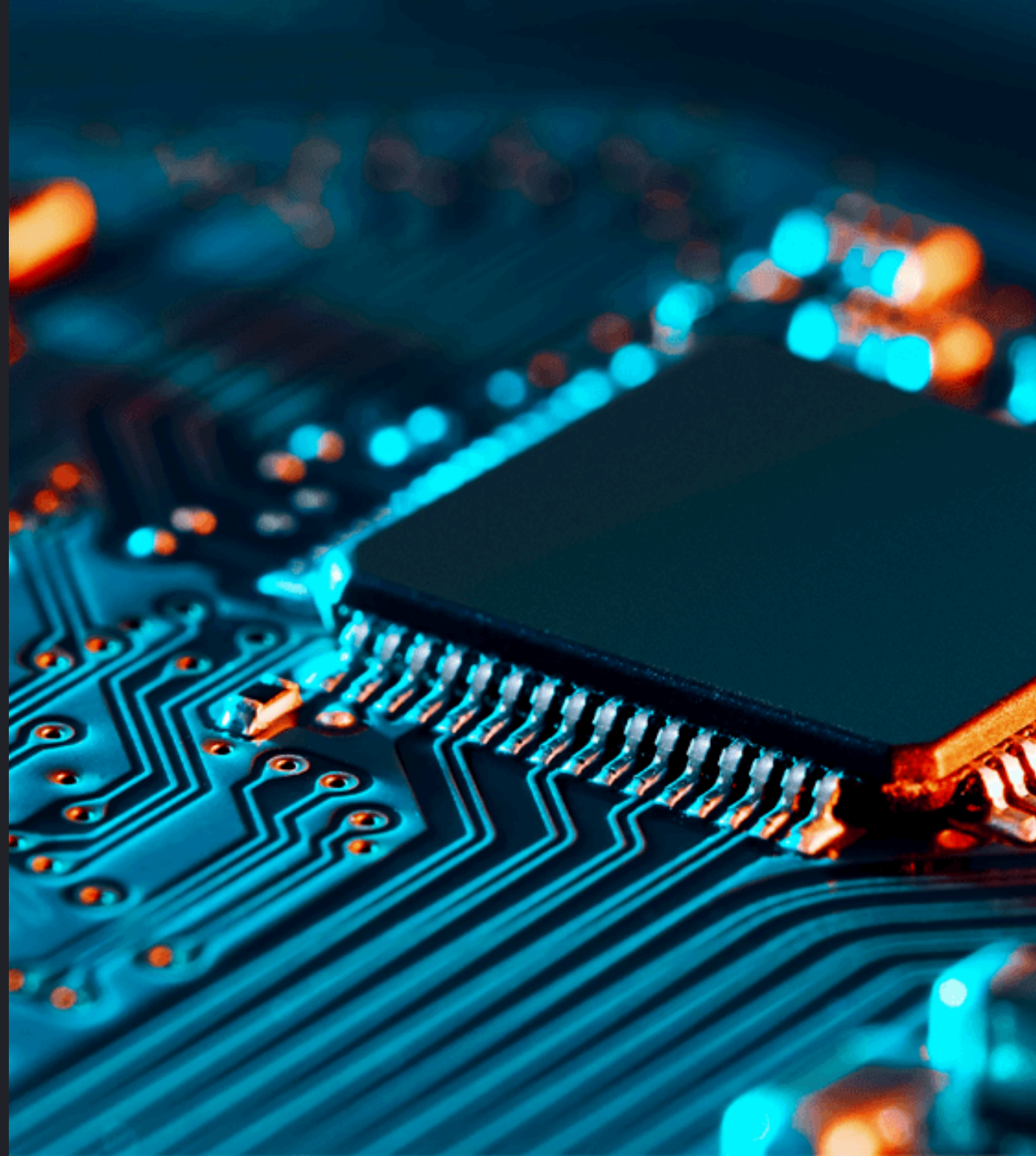
Rétro-ingénierie de systèmes embarqués

Sami Babigeon
Master 2 SSI
16 Septembre, 2024

d'après le cours de Damien Cauquil

Quarkslab



`$ whoami`

- Ancien du master SSI (Promo 2023/2024);
- Travaille @Quarkslab depuis 1 an;
- Passionné par le reverse et l'éléctronique;
- Organisateur du Jeanne d'Hack CTF (*shameless promotion*).

A **vous** de jouer!

Plan

1. Introduction au reverse hardware
2. Monter son labo
3. Electronique: les composants
4. Electronique: les lois
5. Electronique numérique
6. Rétro-ingénierie de PCB
7. Lecture de datasheets
8. Protocoles de communication
9. Interfaces de débogage

Introduction au reverse hardware

Objectifs

- **Comprendre le fonctionnement** d'un système embarqué, son architecture et ses moindres rouages;
- Retrouver le **schéma électronique** d'un circuit imprimé;
- Identifier et **analyser les interactions** entre composants;
- **Extraire le contenu** des composants possédant une mémoire pour une analyse ultérieure.

Objectifs (suite)

- **Reverser** les firmwares extraits;
- Rechercher des **vulnérabilités**, les exploiter;
- Développer du **code sur mesure** pour ce système.

"Cutting ties of a vinyl cutter": https://www.youtube.com/watch?v=o2CHK2ylqF0

Principe général

1. On débute par une **analyse globale** du système;
2. On identifie les **composants**;
3. On identifie **les interactions** entre composants;
4. On reconstitue les **schémas de principe et électronique**;
5. On extrait enfin le **contenu des mémoires**.

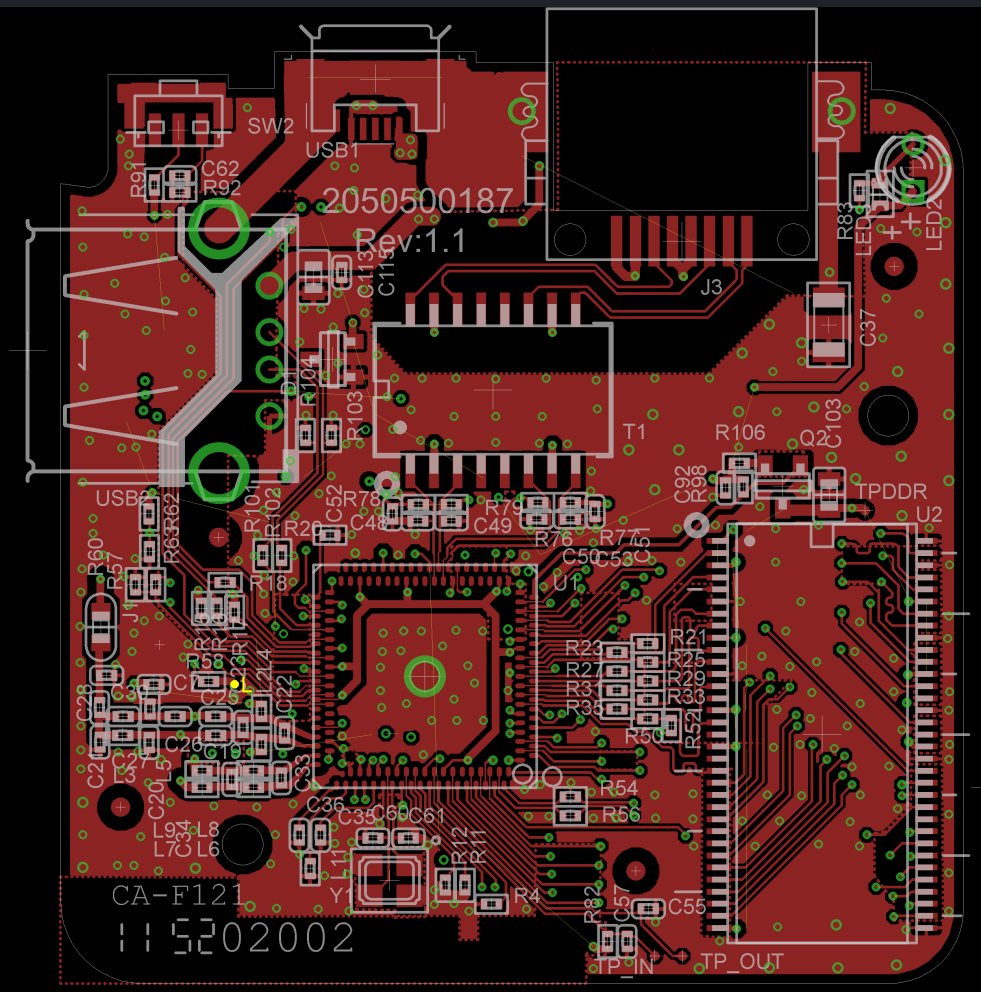
Un exemple concret: TL-WR703N

source: Wikimedia Commons

TL-WR703N

source: OpenWrt

TL-WR703N



TL-WR703N

150M 802.11n Wi-Fi Router

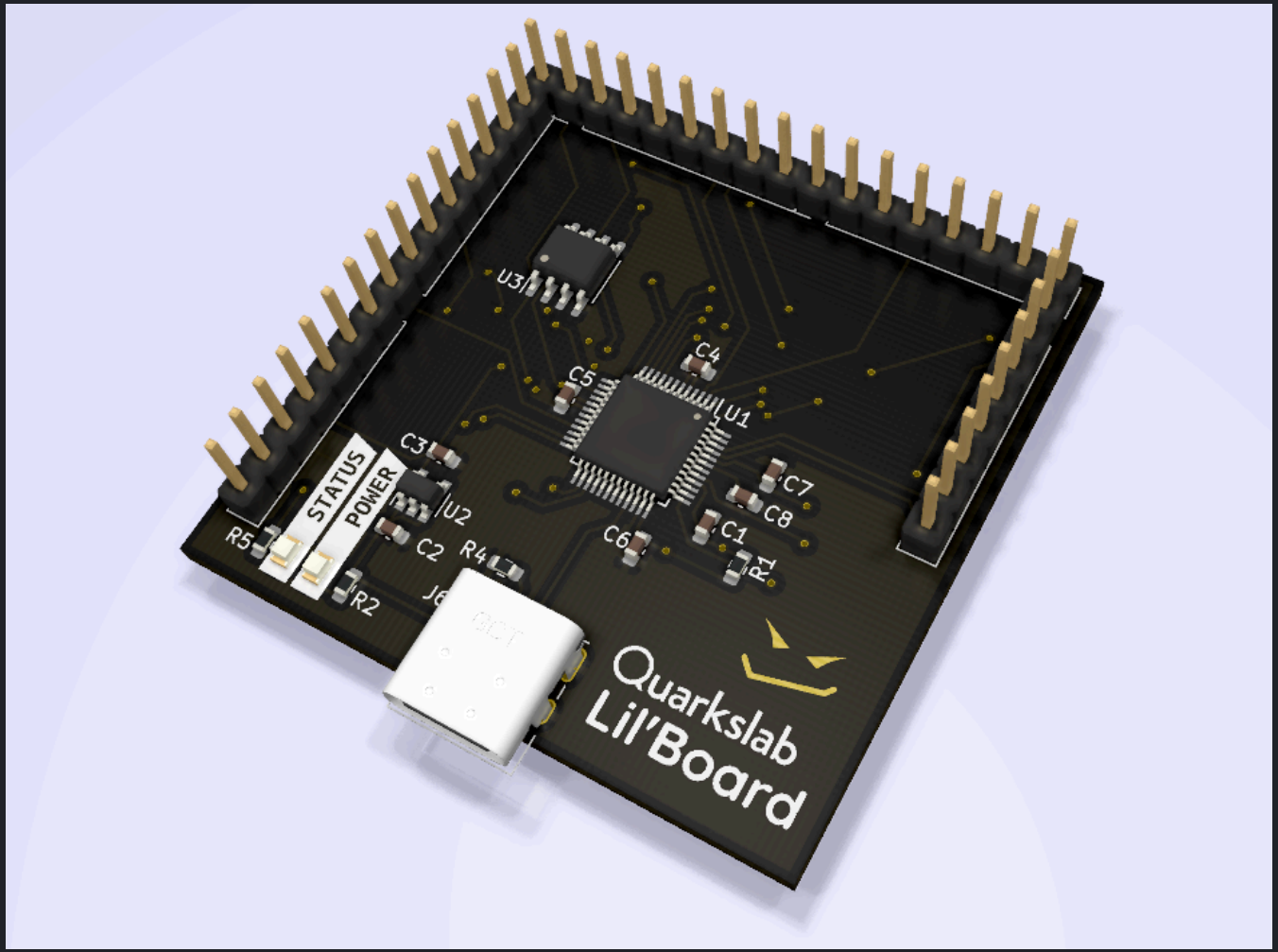
Reverse Engineering Work
http://squonk42.github.com/TL-WR703N/

Based on teardown performed by Kean
on the OpenWRT forums, see:
http://www.kean.com.au/oshw/WR703N/teardown/

CC-BY-SA

© 2012, Michel Stempin. All rights reserved.
Licensed under a Creative Commons
Attribution-Share-Alike Unported License.
See creativecommons.org/licenses/by-sa/3.0/

Cible d'étude



Monter son labo

Comment s'outiller sans se ruiner

S'équiper à pas cher, c'est possible:

- Notre ami **Ali Express** propose du matériel correct peu onéreux (mais avec des délais de livraison conséquents);
- **Amazon** propose un grand choix d'outils abordables et livrables rapidement (attention toutefois à la qualité);
- L'**impression 3D** peut aider dans certains cas.

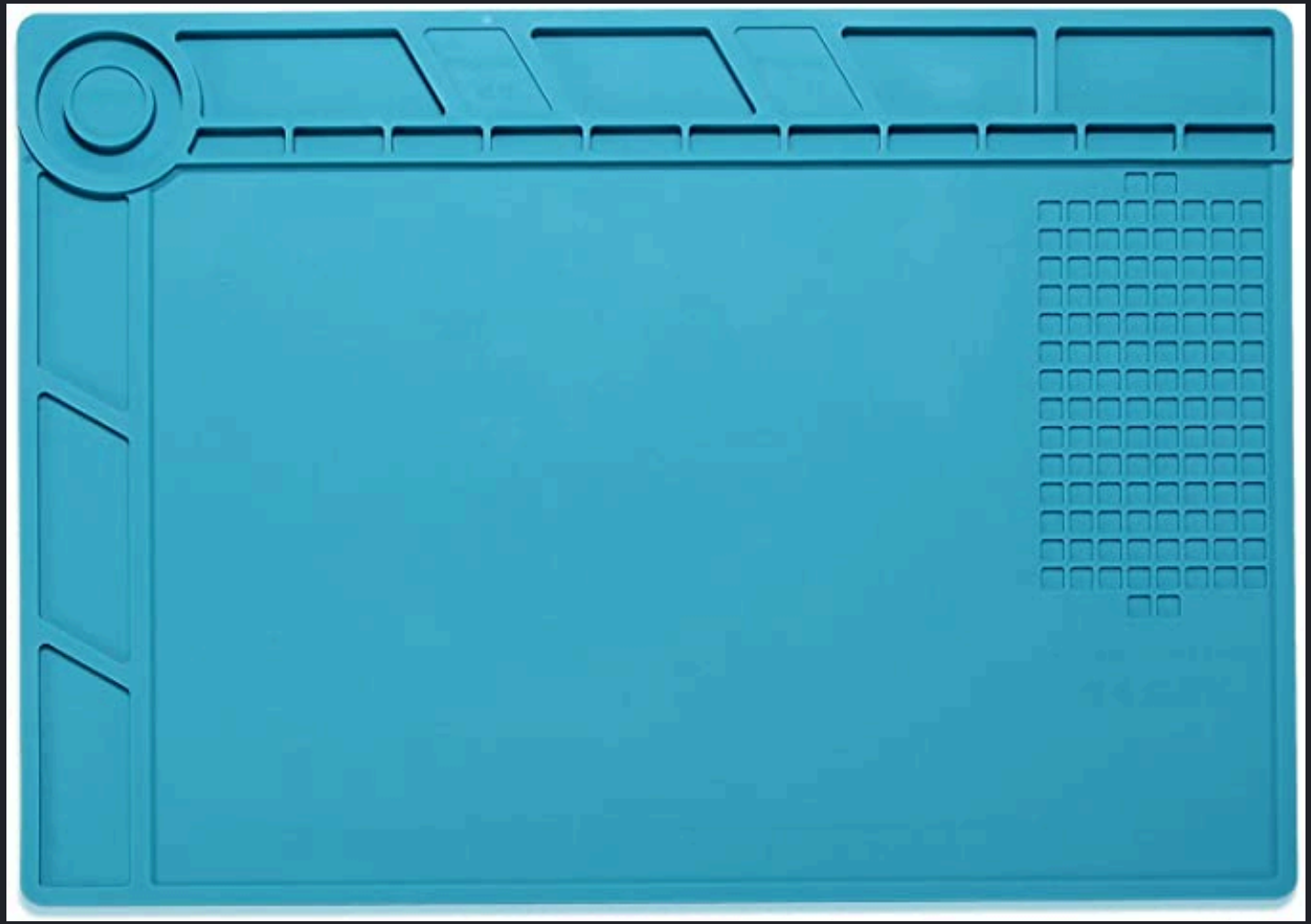
Comment s'outiller sans se ruiner

Quelues conseils pratiques:

- Les outils onéreux ont souvent des **alternatives moins chères**;
- Il est possible de se **débrouiller avec des outils abordables** pour réaliser des tâches requérant habituellement des outils chers;
- Certains outils sont tout de même **indispensables**, mais restent très abordables!

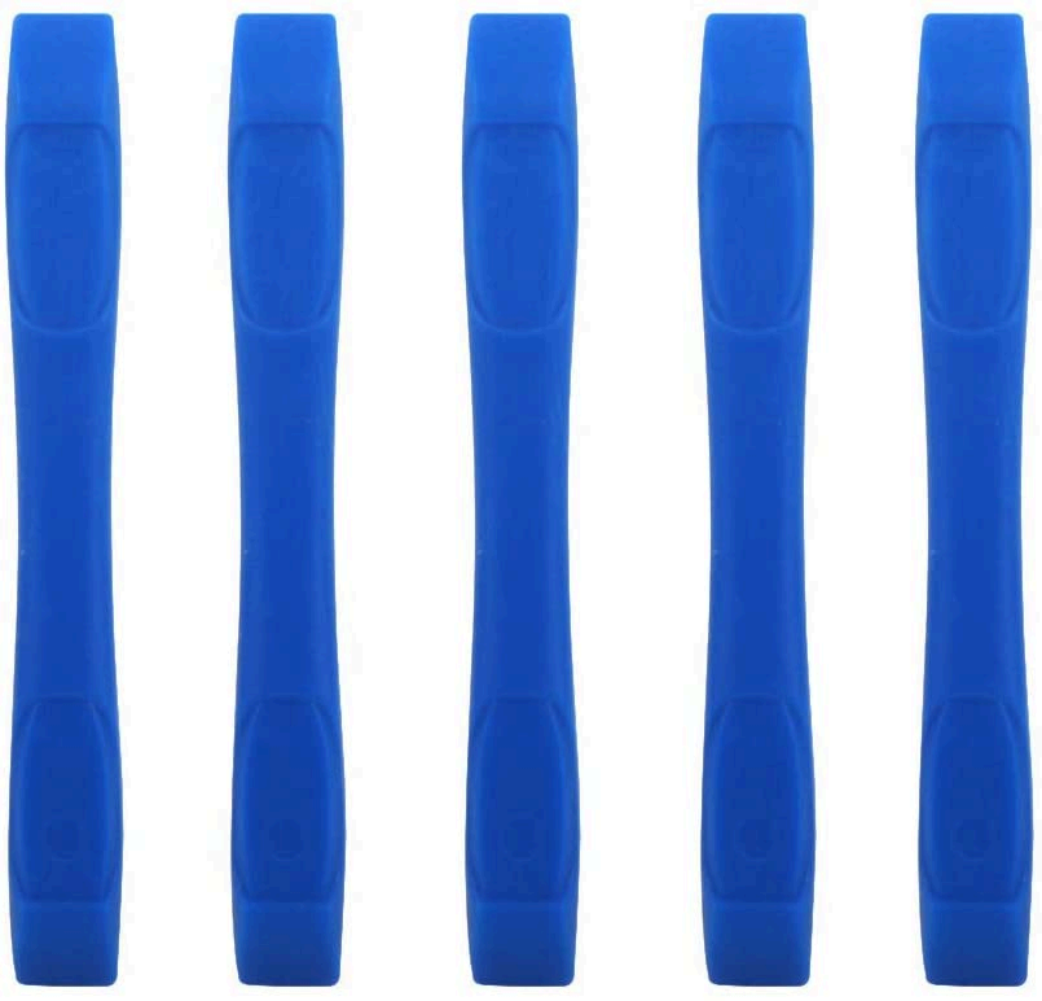
La quincaillerie indispensable

Le tapis silicone



Jeu de tournevis

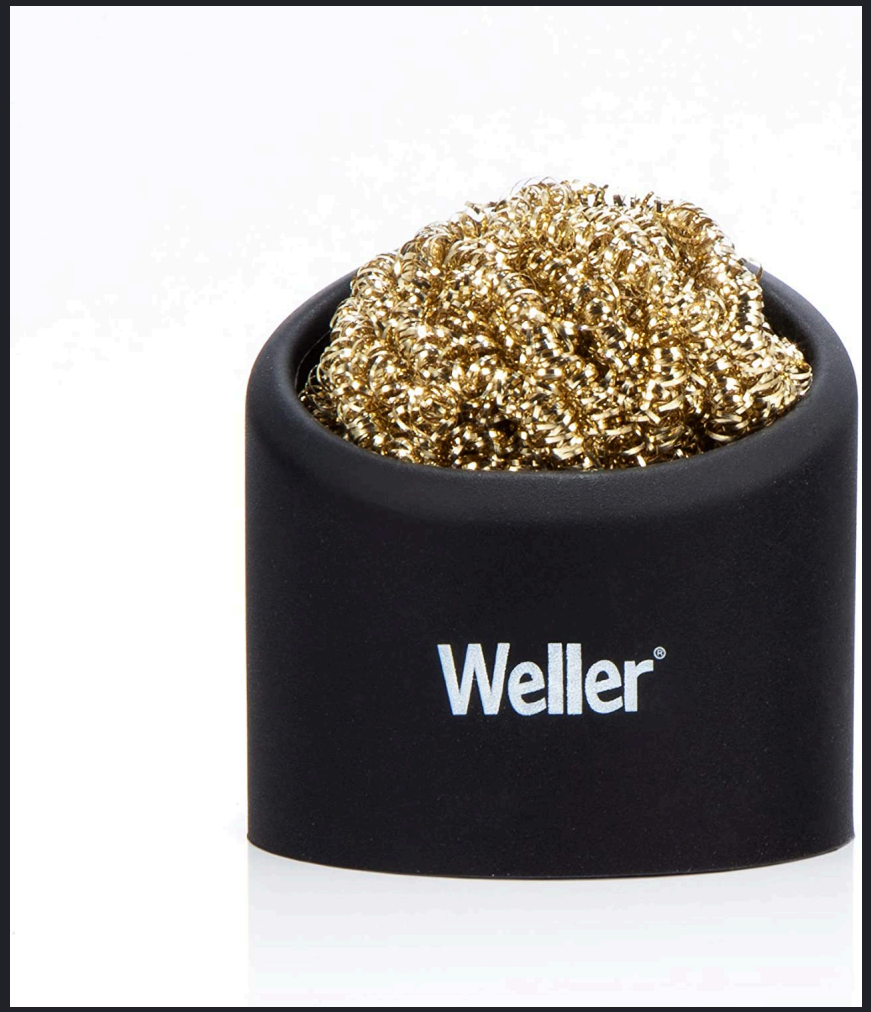
Spudgers



Fer à souder

Bobine d'étain SnPb

Eponge laiton



Pince coupante

Pinces antistatiques

Fils de prototypage

Pince à dénuder

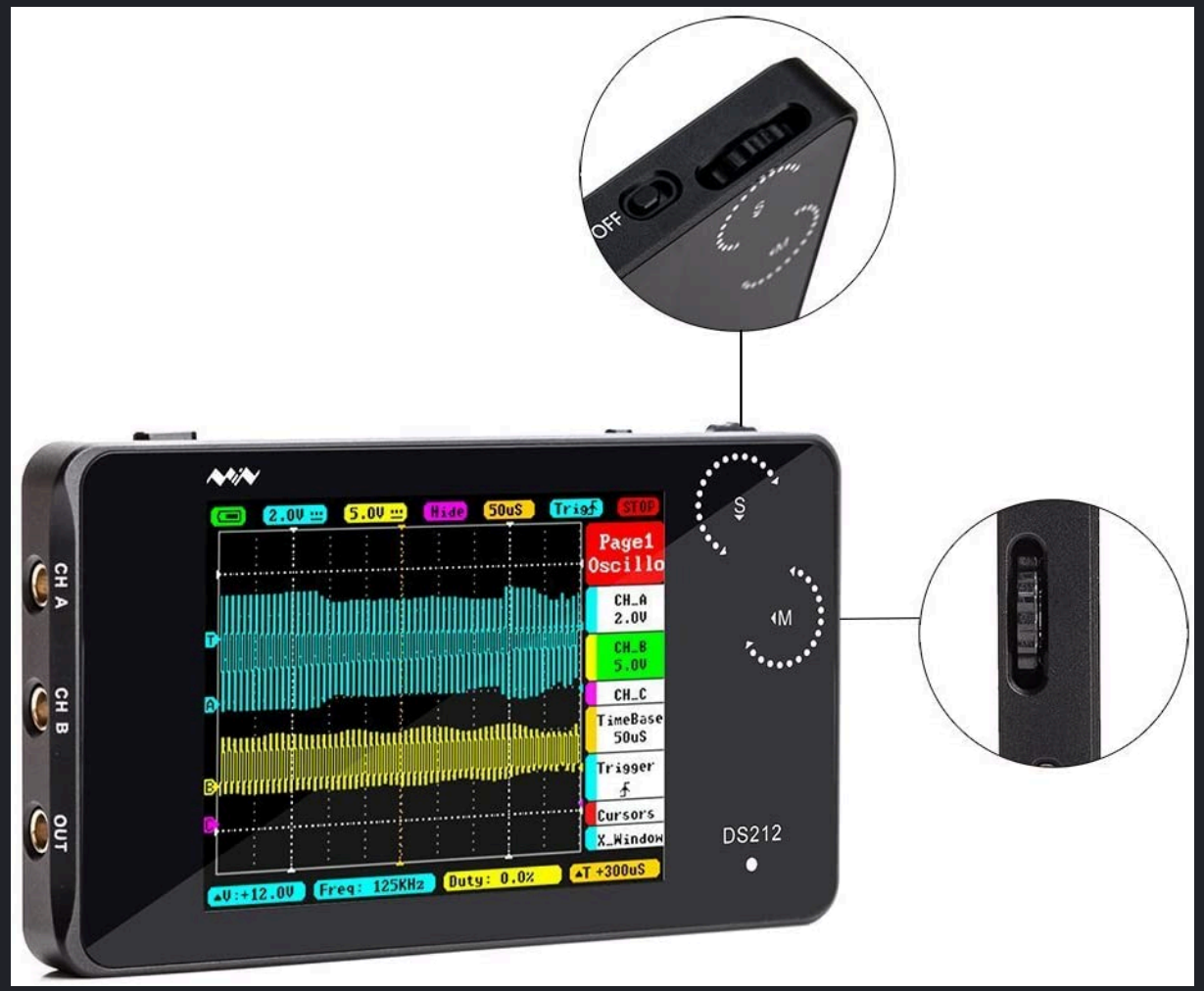
Loupe éclairante

Les outils de mesure

Multimètre

PROSTER®
VC97A
AUTO
APO
AC
230.7
V
MAX/MIN
Hz/DUTY
REL
HOLD
RANGE
SELECT
Hz
NCV
°C °F
hFE
OFF
AUTO OFF POWER
A
uAmA
COM
VΩHz
TRUE RMS

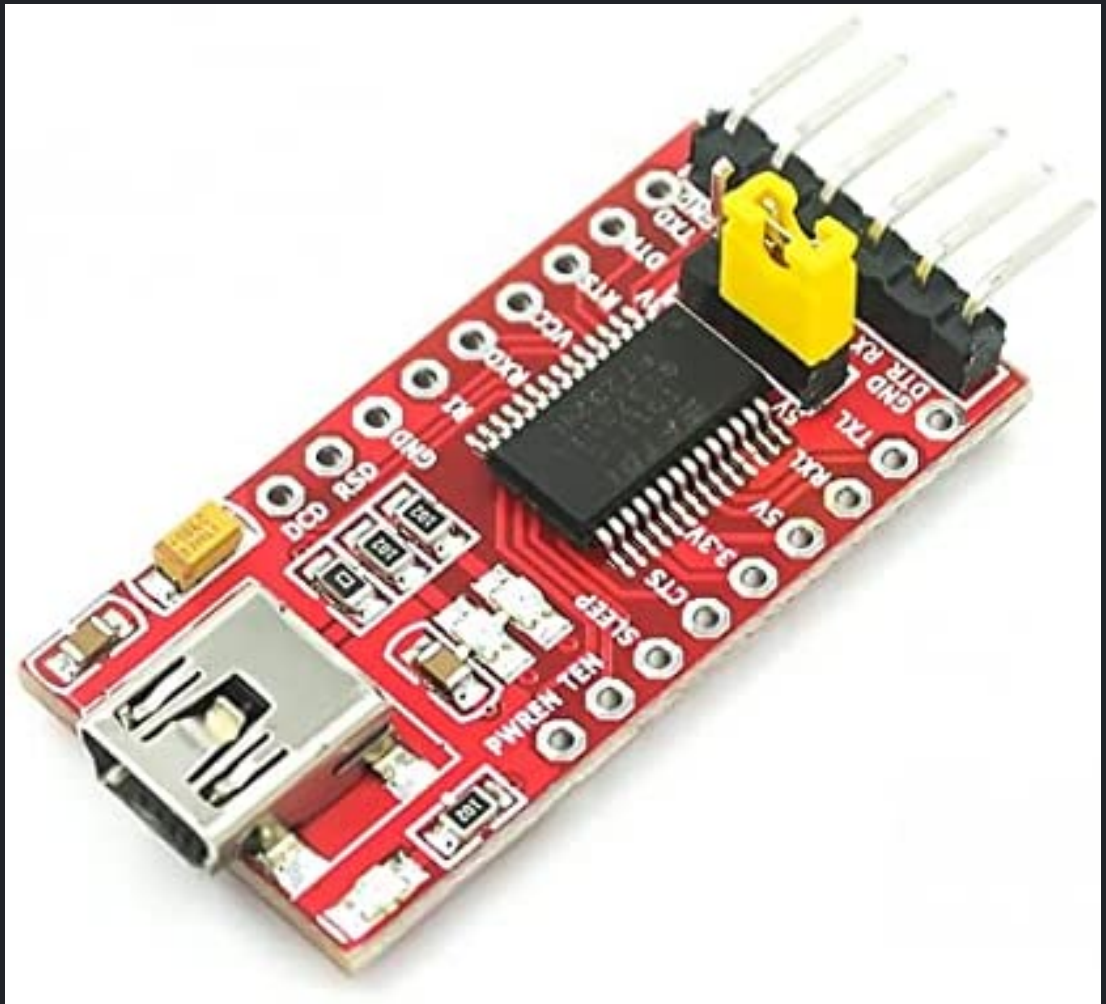
Oscilloscope de poche



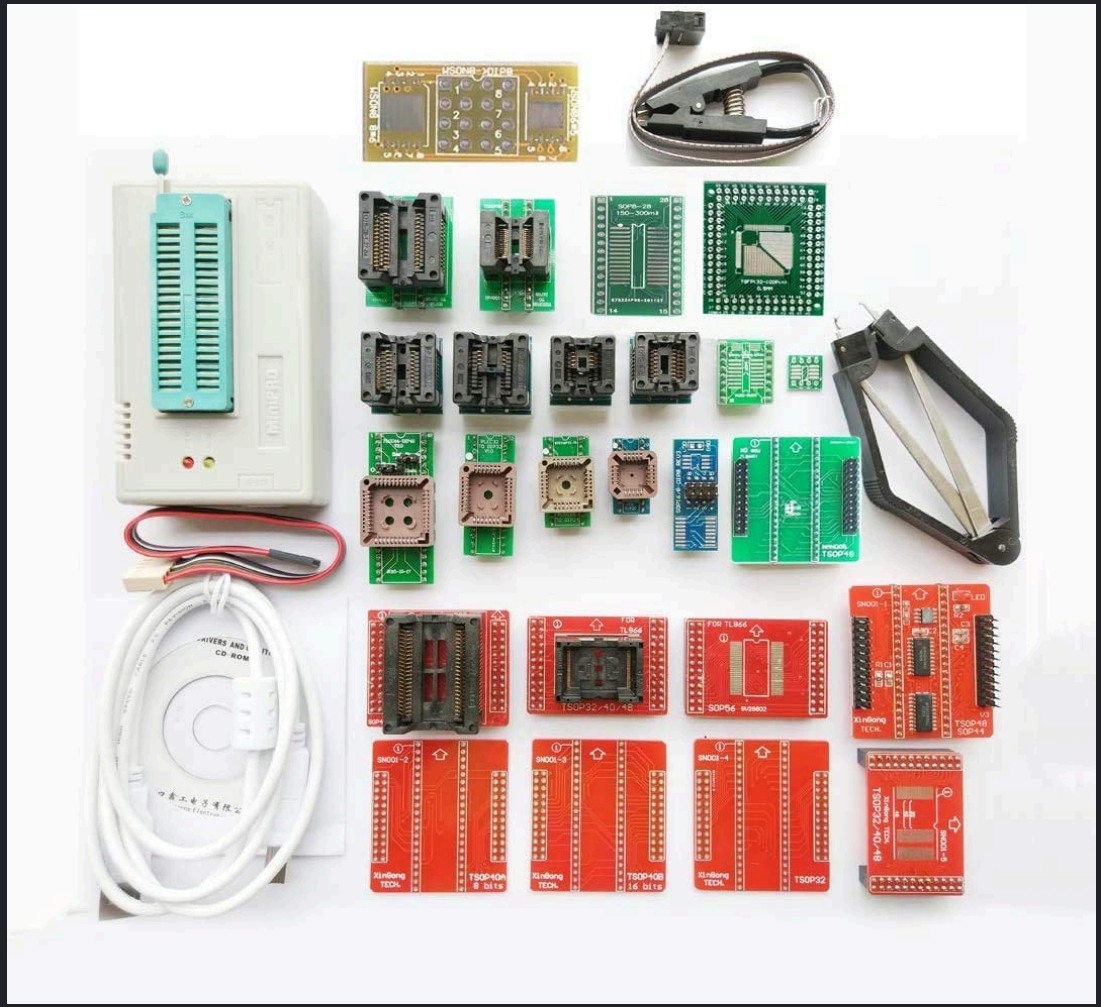
Les outils d'acquisition

Analyseur logique

Adaptateur FTDI



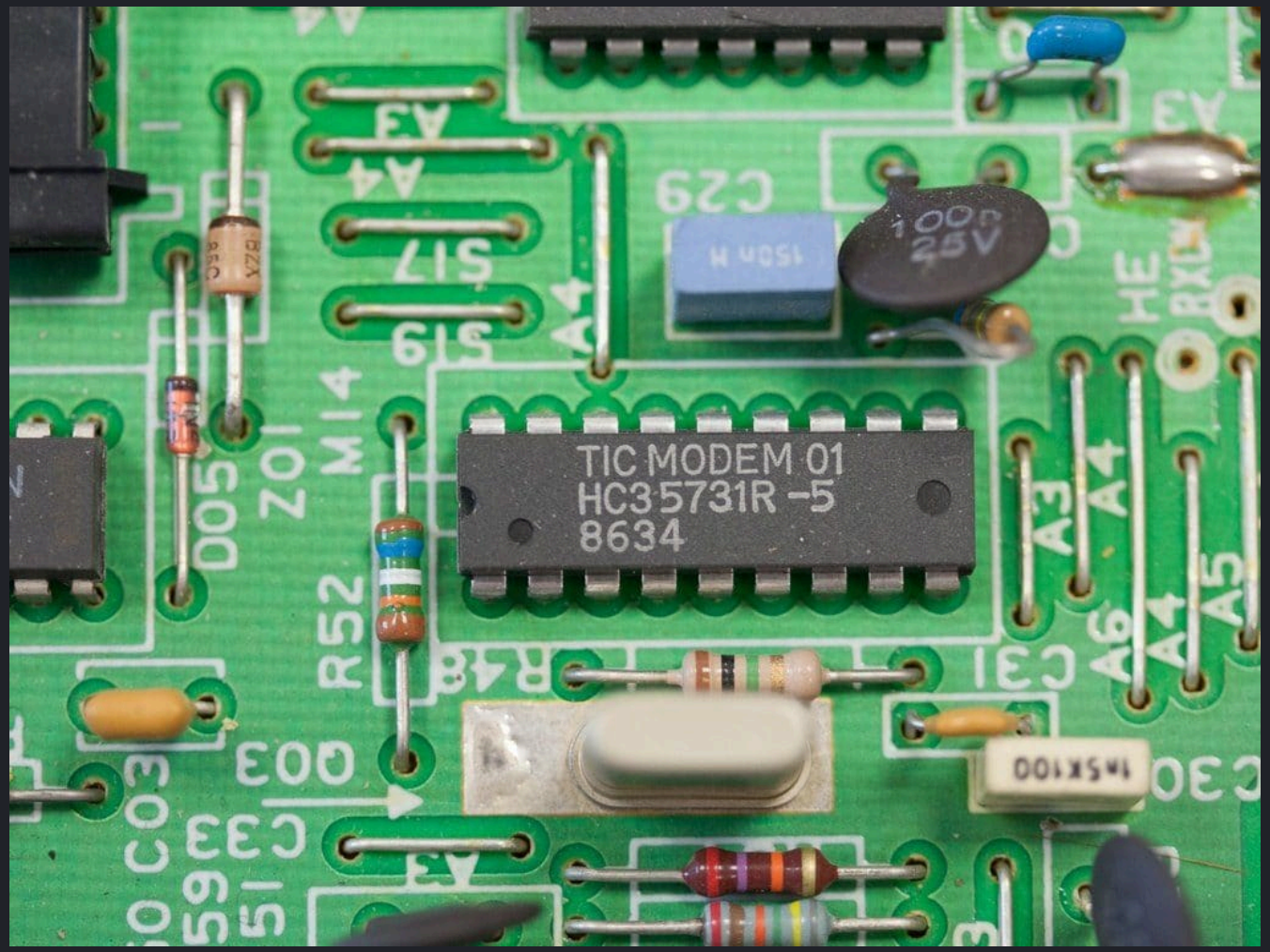
TL866 II Plus



Electronique: les composants

Technologie des composants

Historiquement, les composants étaient placés d'un côté et soudé de l'autre:



source: tomshardware.fr

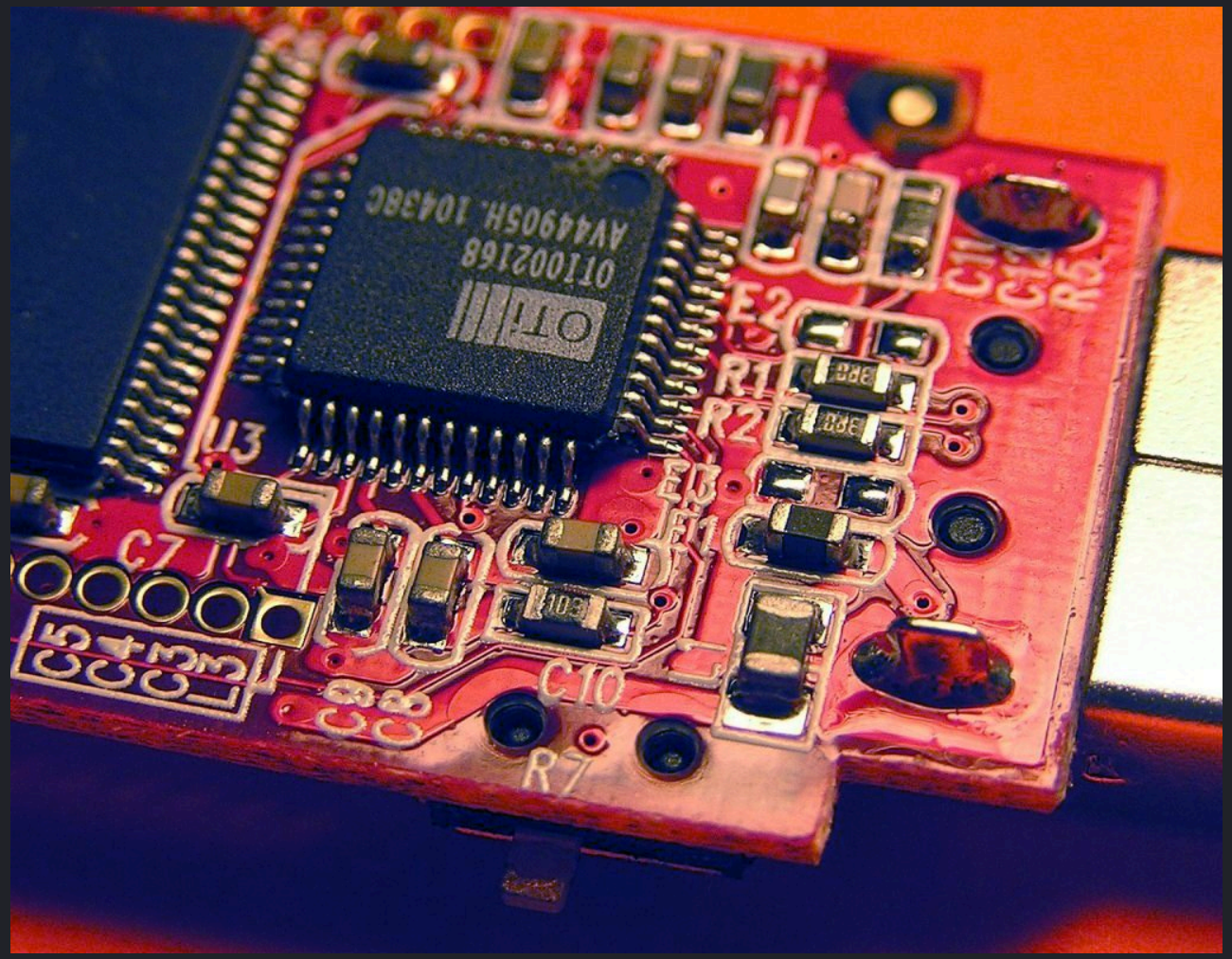
Technologie des composants

Cette méthode avait plusieurs inconvénients:

- l'assemblage automatisé était **complexe** (beaucoup de ces circuits étaient donc **assemblés à la main**);
- le **nombre de composants** au cm2 (densité) était **réduit**.

Technologie des composants

Le placement en surface est donc apparu, et s'est démocratisé:



source: wikipedia

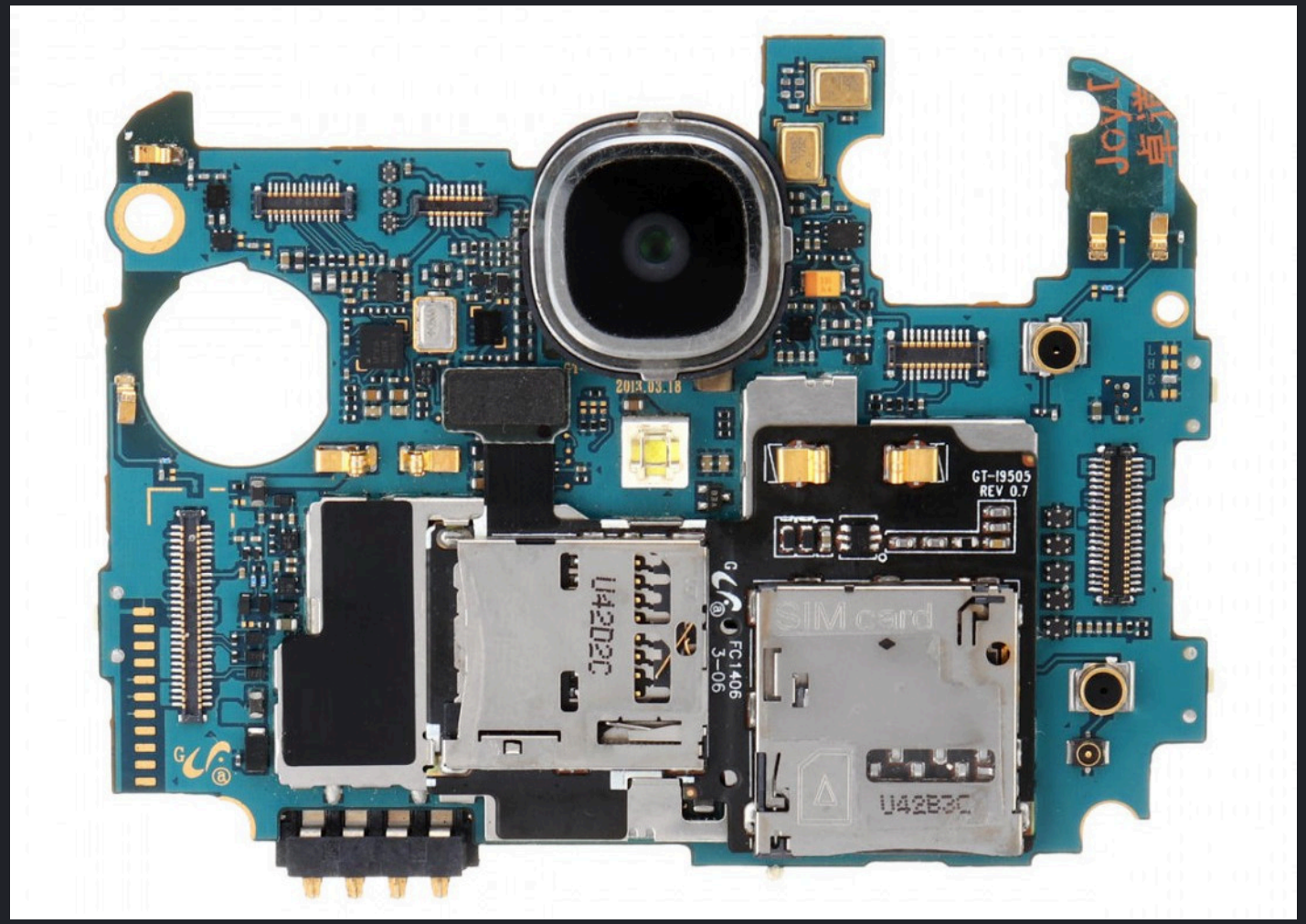
Technologie des composants

Le placement en surface offre de nombreux avantages:

- Il permet un assemblage **automatisé** (*pick and place*);
- La soudure des composants peut être réalisée avec un four spécial (*reflow oven*);
- Les composants peuvent être placés des deux côtés du PCB (**meilleure densité**).

Technologie des composants

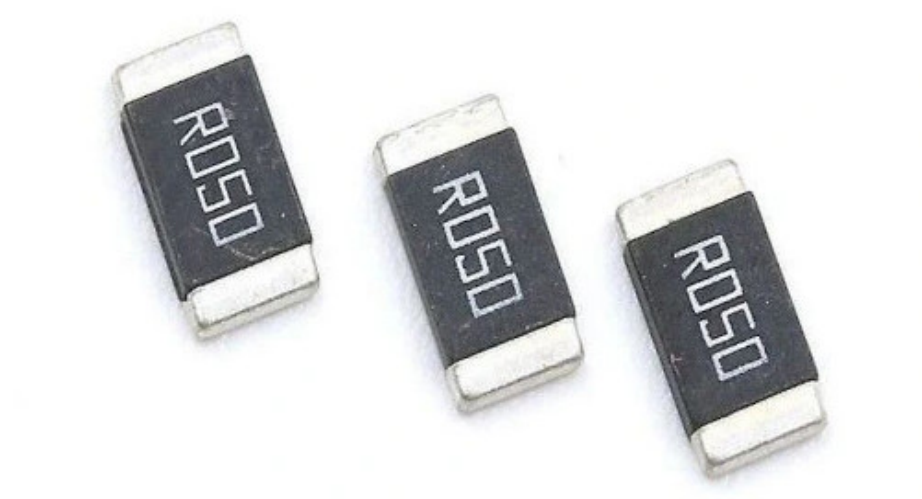
Les composants ont depuis été miniaturisés, et les PCBs ressemblent aujourd'hui plutôt à cela:



Les composants passifs

- **Composant passif**: ne permet pas d'augmenter la puissance d'un signal (en tension ou intensité)
- Résistance, condensateur, inductance

La résistance



La résistance

- Valeur exprimée en **Ohms** (Ω);
- Valeurs normalisées (**série E12** très répandue);
- **Code couleur** sur les résistances traversantes;
- **Code numérique** sur les résistances SMD (si lisible).

Le condensateur

Le condensateur

- Valeur exprimée en **Farads** (F);
- Valeurs normalisées;
- Marquage sur certains composants, mais **généralement aucun**.

Diode

Diode

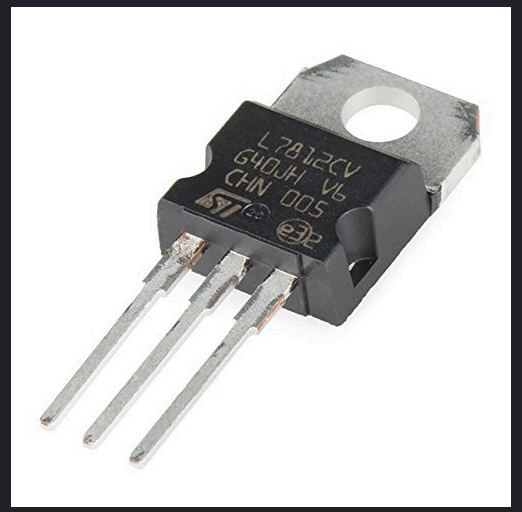
- La diode a **une tension fixe** à ses bornes, dénommée habituellement *Vf* et généralement autour de **2 volts**;
- Elle peut supporter une **tension maximale** indiquée dans la documentation;
- Elle ne laisse passer le courant **que dans un seul sens**.

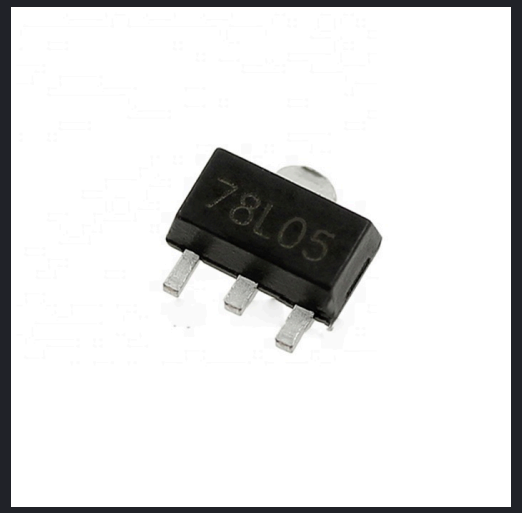
Transistor

Transistor

- **Semi-conducteur** qui laisse passer le courant entre son collecteur (C) et son émetteur (E), en fonction du courant appliqué à sa base (B);
- Lorsque le courant appliqué à la base dépasse une intensité précise, le transistor devient **saturé** (fonctionne comme un **interrupteur électronique**);
- Lorsque le courant appliqué à la base n'est pas suffisant pour le saturer, il fonctionne en **régime transitoire**.

Le régulateur

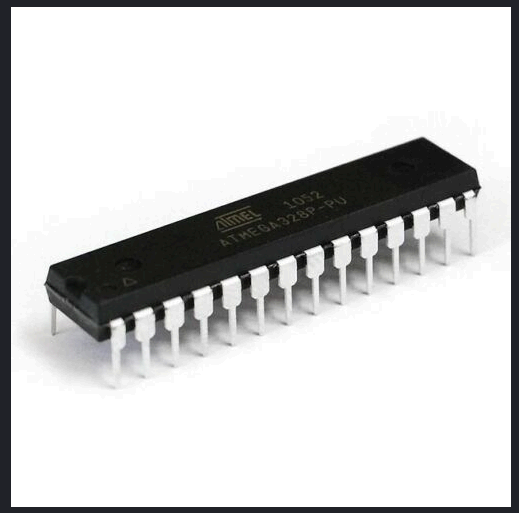




Le régulateur

- Le régulateur est un circuit intégré permettant de **fournir une tension fixe** (régulée);
- Il existe **différents types** de régulateurs: linéaires, Low Drop-Out (LDO);
- Il possède généralement **un dissipateur** pour dissiper l'excédent de puissance sous forme de chaleur (en particulier le régulateur linéaire)

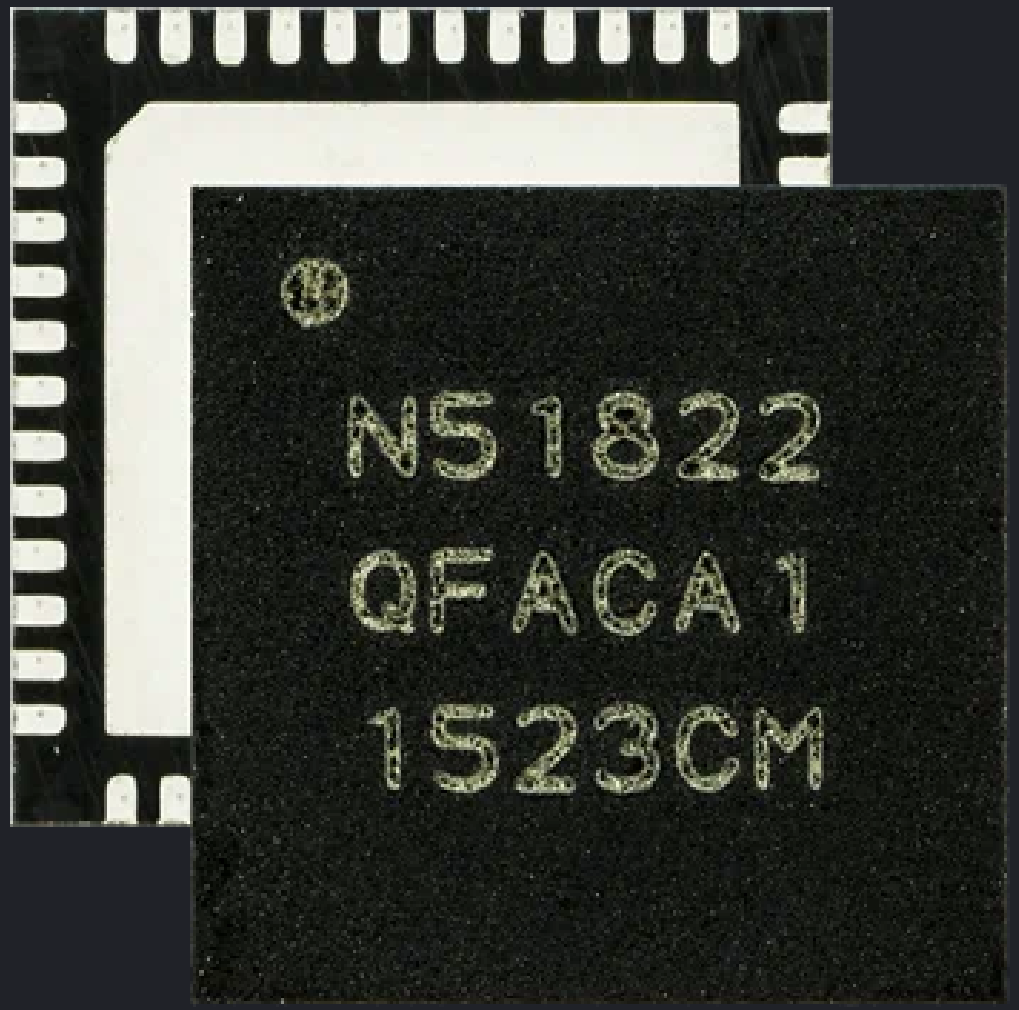
Le micro-contrôleur



Le micro-contrôleur

- Généralement constitué d'un **CPU**, de **RAM**, de mémoire **Flash** et de quelques **interfaces de communication**;
- De **nombreux formats** de boîtiers (DIP, TQFP, QFN, BGA, etc.);
- Contient un **programme** (**firmware** en anglais).

Le système-sur-puce (SoC)

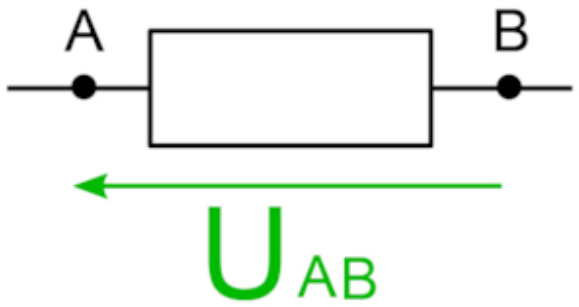


Le système-sur-puce (SoC)

- **Plus complexe** que le micro-contrôleur;
- Intègre des périphériques permettant de **réaliser des fonctions avancées** (traitement vidéo, communications RF, ethernet, etc.);
- Formats de boîtiers variés.

Electronique: les lois

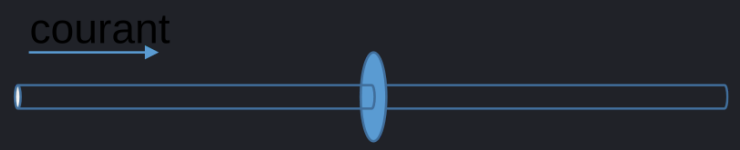
La tension

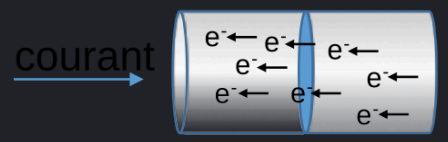


$$U_{AB} = V_A - V_B$$

- La tension est définie comme une **différence de potentiels électriques**;
- Elle est exprimée en **volts** (V);
- On la représente avec une **flèche pointant sur le potentiel le plus élevé**.

L'intensité



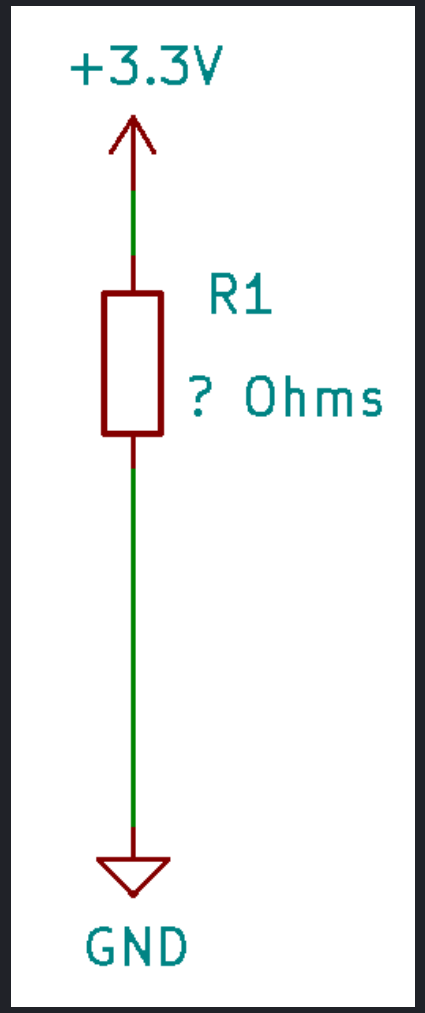


- L'intensité d'un courant est la quantité de charges électriques traversant un conducteur à un instant précis;
- Elle est exprimée en **ampères** (A);
- On la représente avec une **flèche allant dans le sens opposé de la circulation des électrons**, a.k.a "du plus vers le moins".

Loi d'Ohm

$$U = R \times I$$

- **U**: Tension en volts (V);
- **I**: Intensité en ampères (A);
- **R**: Résistance en ohms (Ω).



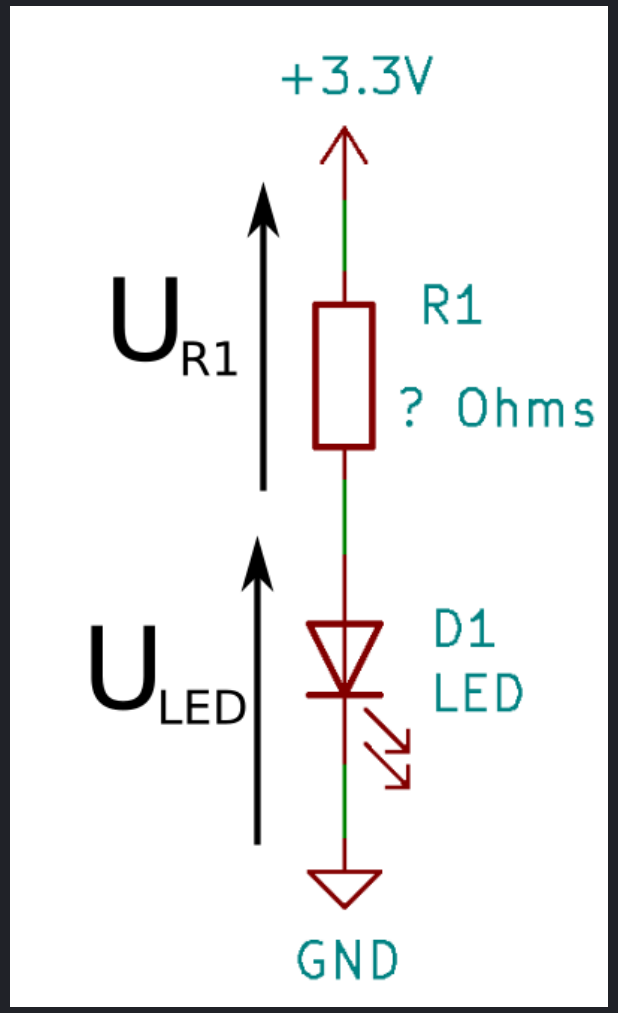
Application de la loi d'Ohm

- Intensité désirée dans le circuit: **20 mA**
- Tension d'alimentation: **3.3 V**
- $R = U/I$ soit $R = 3.3/0.02 = 165\Omega$
- La valeur **165** n'est pas standard, on choisit la plus proche: **180**.
- Soit une intensité de $I = 3.3/180 = 18,3$ mA

Loi d'additivité des tensions

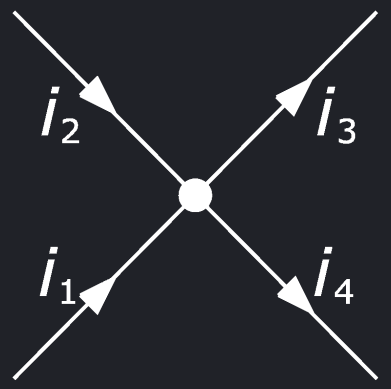
La tension aux bornes d’un ensemble de récepteurs branchés en série correspond à la somme des tensions de chacun d’entre eux.

Loi d'additivité des tensions



- $U_{R1} + U_{LED} = 3.3V$
- Intensité désirée dans le circuit: **20 mA**
- La tension au borne de la led V_f est de **2V**
- $R = U_{R1}/I$ soit $R = (3.3 - V_f)/I$, donc $R = (3.3 - 2.0)/0.02 = 65\Omega$
- La valeur **65** n'est pas standard, on choisit la plus proche: **68**.
- Soit une intensité de $I = (3.3 - 2.0)/68 = 19,1$ mA

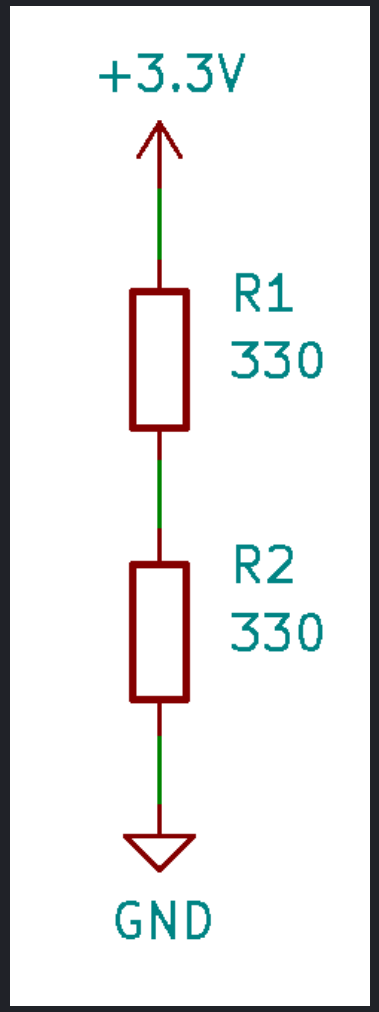
Loi des noeuds



$$i_1 + i_2 = i_3 + i_4$$

source: WikiMedia

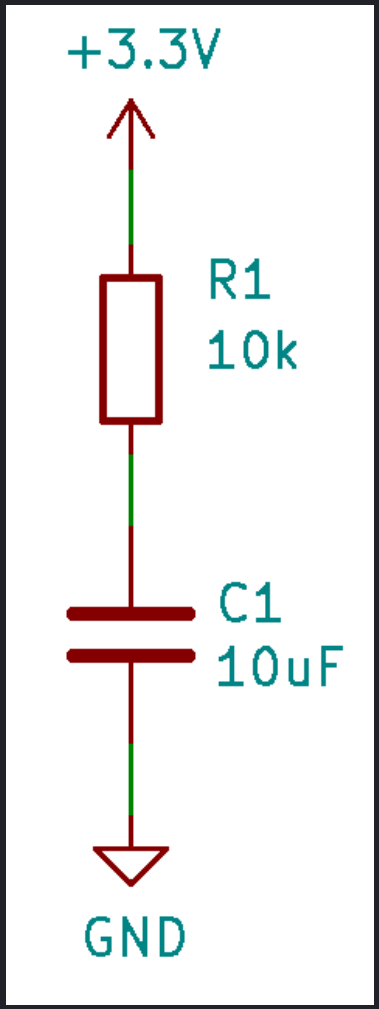
Dans un noeud, la somme des intensités des courants entrants est équivalente à la somme des intensités des courants sortants.



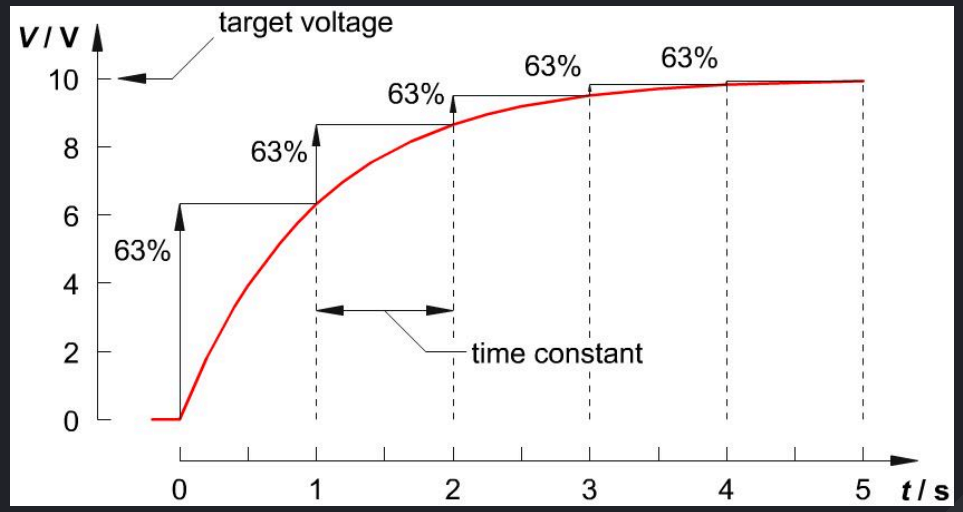
Pont diviseur de tension

- $U_{R1} + U_{R2} = 3.3$ V
- I = $\frac{U}{R1+R2} = 3.3/660 = 5$ mA
- $U_{R2} = R2 \times I = R2 \times \frac{U}{R1+R2} = U \times \frac{R2}{R1+R2}$
- $U_{R2} = 3.3 \times 0.5 = 1.65$ V
- Utilisé pour l'adaptation de petites tensions

Circuit RC



- Le condensateur **C1** se charge via le courant qui l'alimente, défini par **R1** (cas d'un condensateur parfait, ESR=0)
- La durée de charge est $T = 5 \times R \times C$



Circuit RC

- Utile pour du ***lissage*** *de tension* (éliminations des fluctuations parasites, *découplage*);
- Permet de réaliser des **circuits de délai** (combiné à un ou plusieurs transistors);
- Peut **emmagasiner et restituer de l'énergie** (faire office de *batterie* très sommaire);

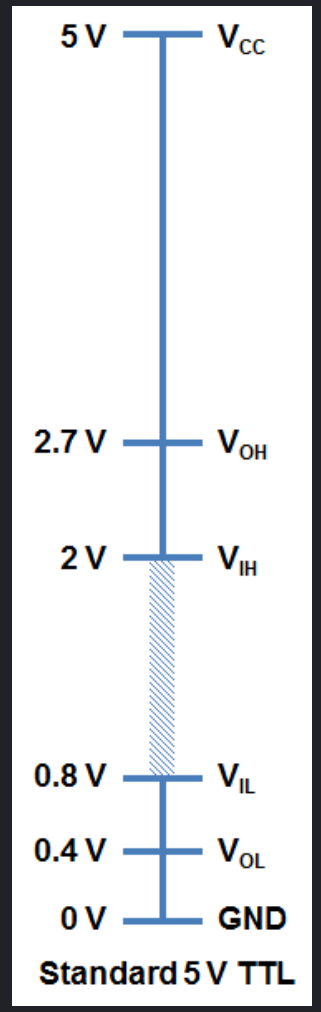
Electronique numérique

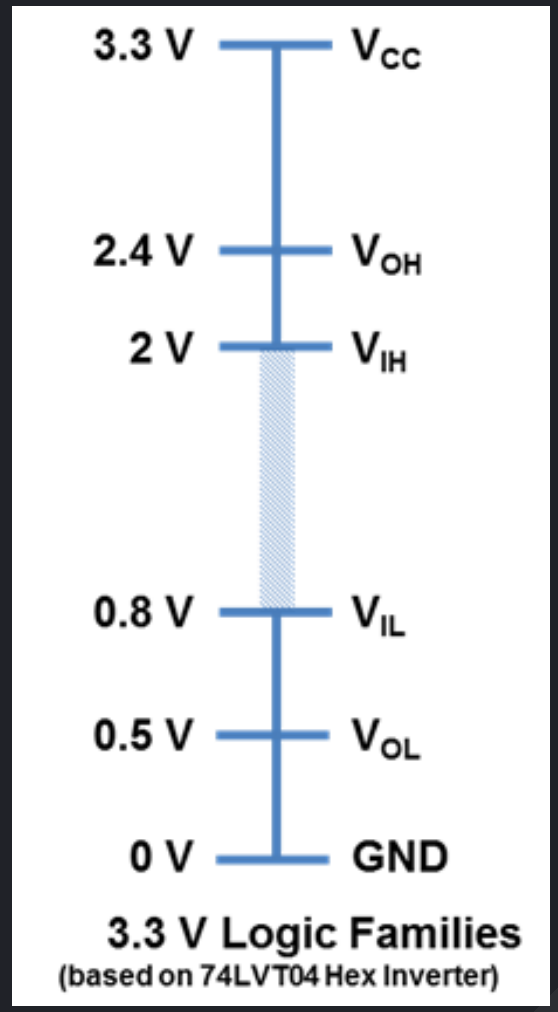
Electronique analogique vs. numérique

- On interprète des signaux analogiques comme des **niveaux logiques** (VRAI, FAUX);
- Des **composants intégrés** implémentent tout un ensemble de **circuits logiques**;
- L'électronique numérique fonctionne à base de **tensions standardisées** (5V, 3.3V principalement).

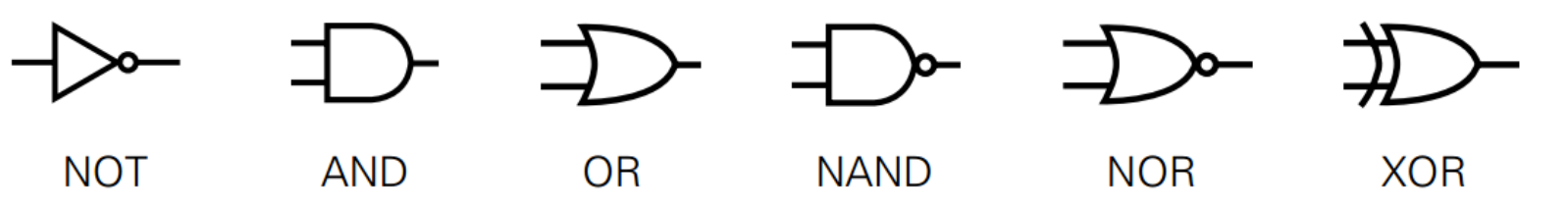
Electronique analogique vs. numérique

source: Sparkfun





Portes logiques



NOT

A	S
0	1
1	0

AND

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

NAND

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

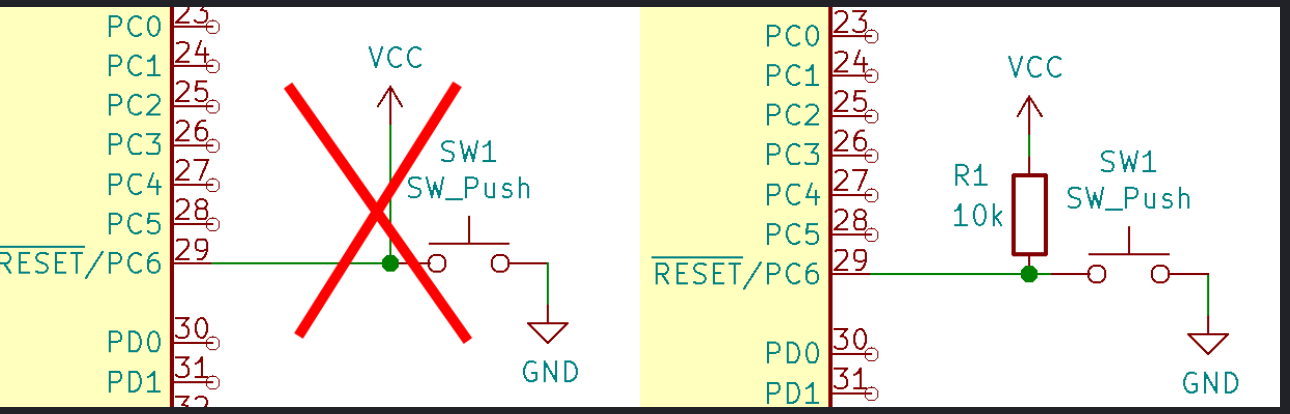
NOR

A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

XOR

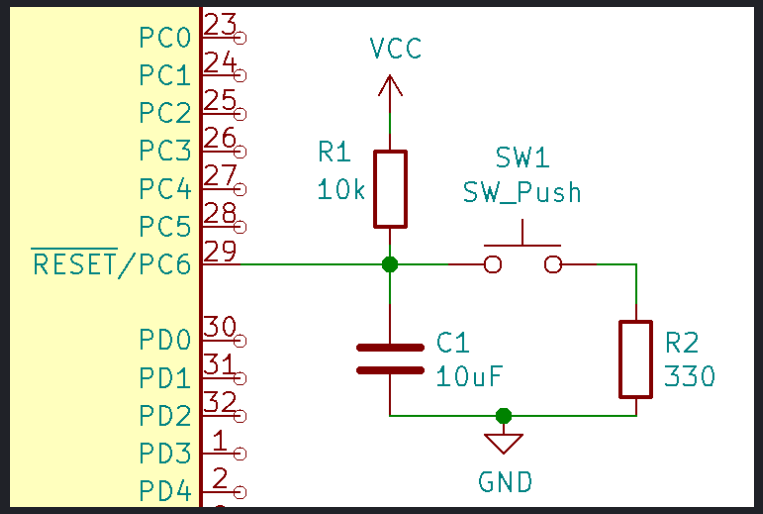
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

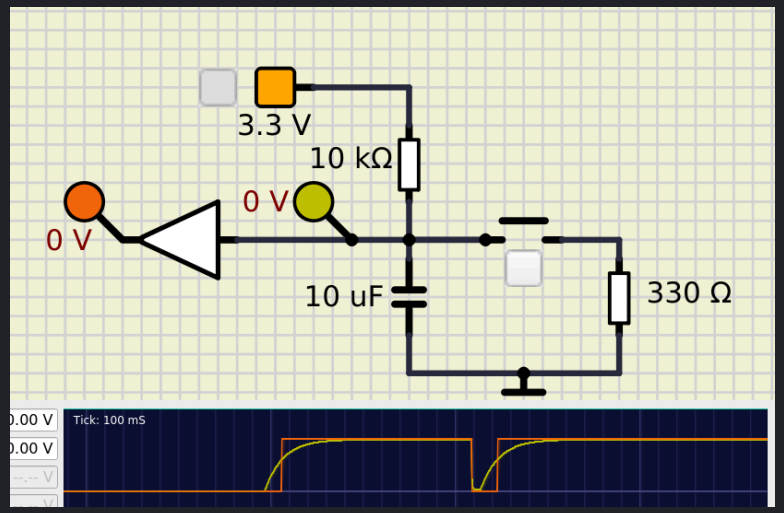
Pull-up et pull-down



- En électronique numérique, il faut pouvoir **fixer le niveau logique** d'une entrée;
- On **utilise des résistances** pour relier l'entrée soit à *Vcc* (état haut, *pull-up*) soit à la masse (à *GND*, état bas, *pull-down*).

Circuit de reset de MCU



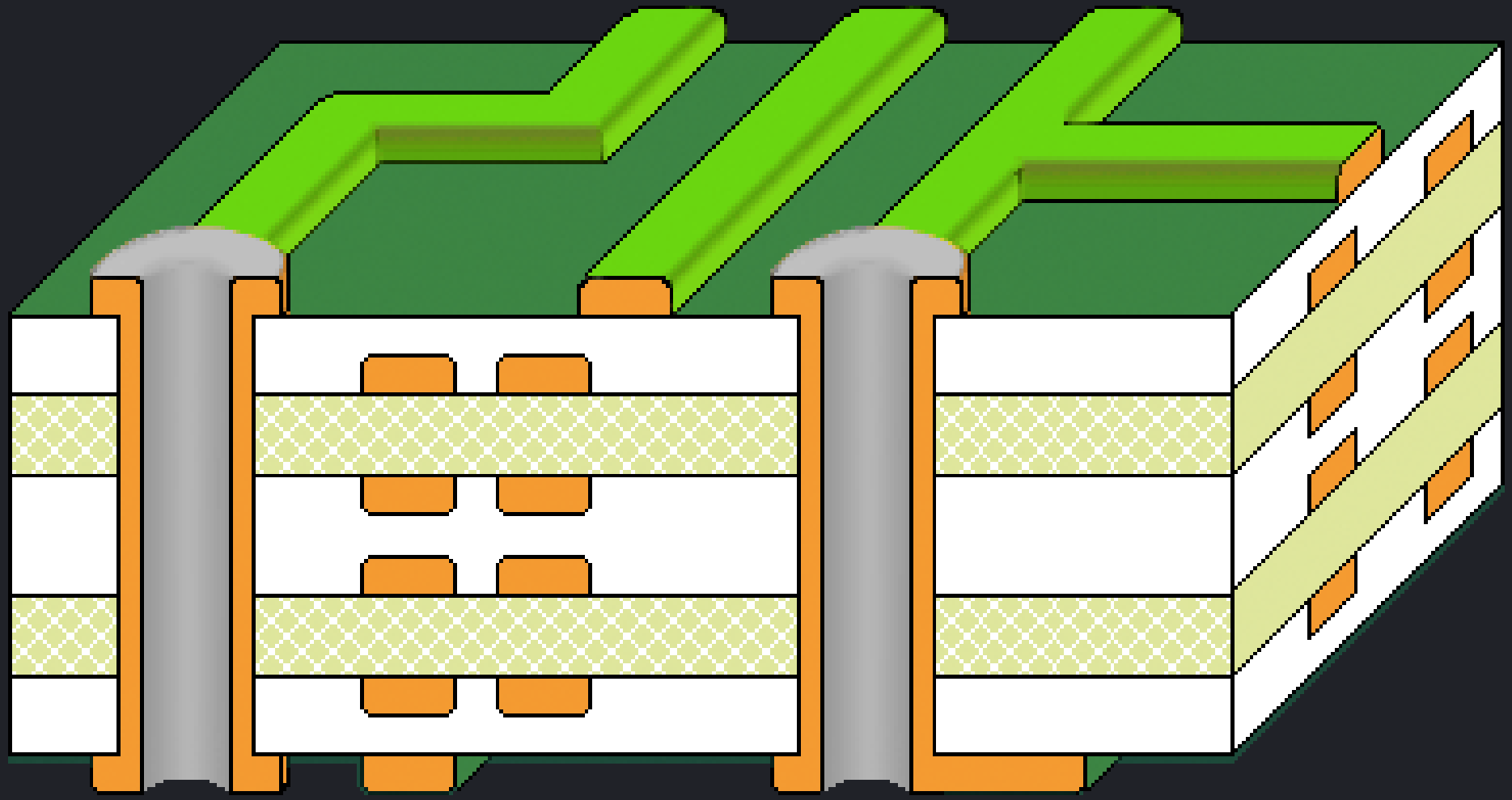


Rétro-ingénierie de PCB

Anatomie d'un PCB

- *PCB* signifie *Printed Circuit Board*, généralement traduit en *Circuit imprimé* en français;
- Il s'agit d'un **support** fait majoritairement **en fibre de verre** sur lequel des circuits de cuivre sont collés qui permettent de relier des composants électroniques entre eux;
- Leur **conception et production** sont **industrialisés**;
- La finesse des circuits imprimés s'est beaucoup améliorée, permettant l'emploi de composants miniaturisés et permettant la **réduction de la taille des circuits**.

Anatomie d'un PCB

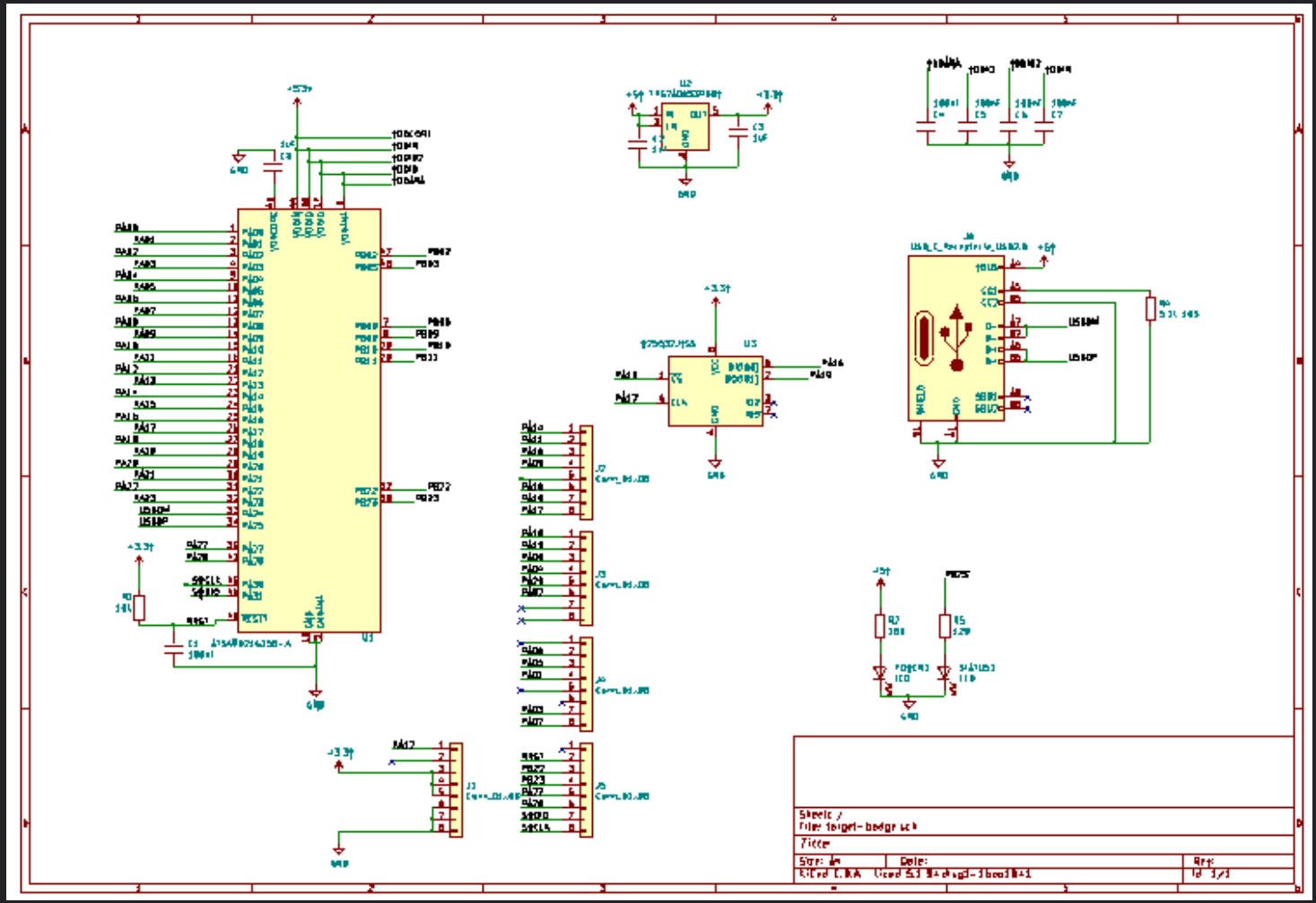


Caractéristiques d'un PCB

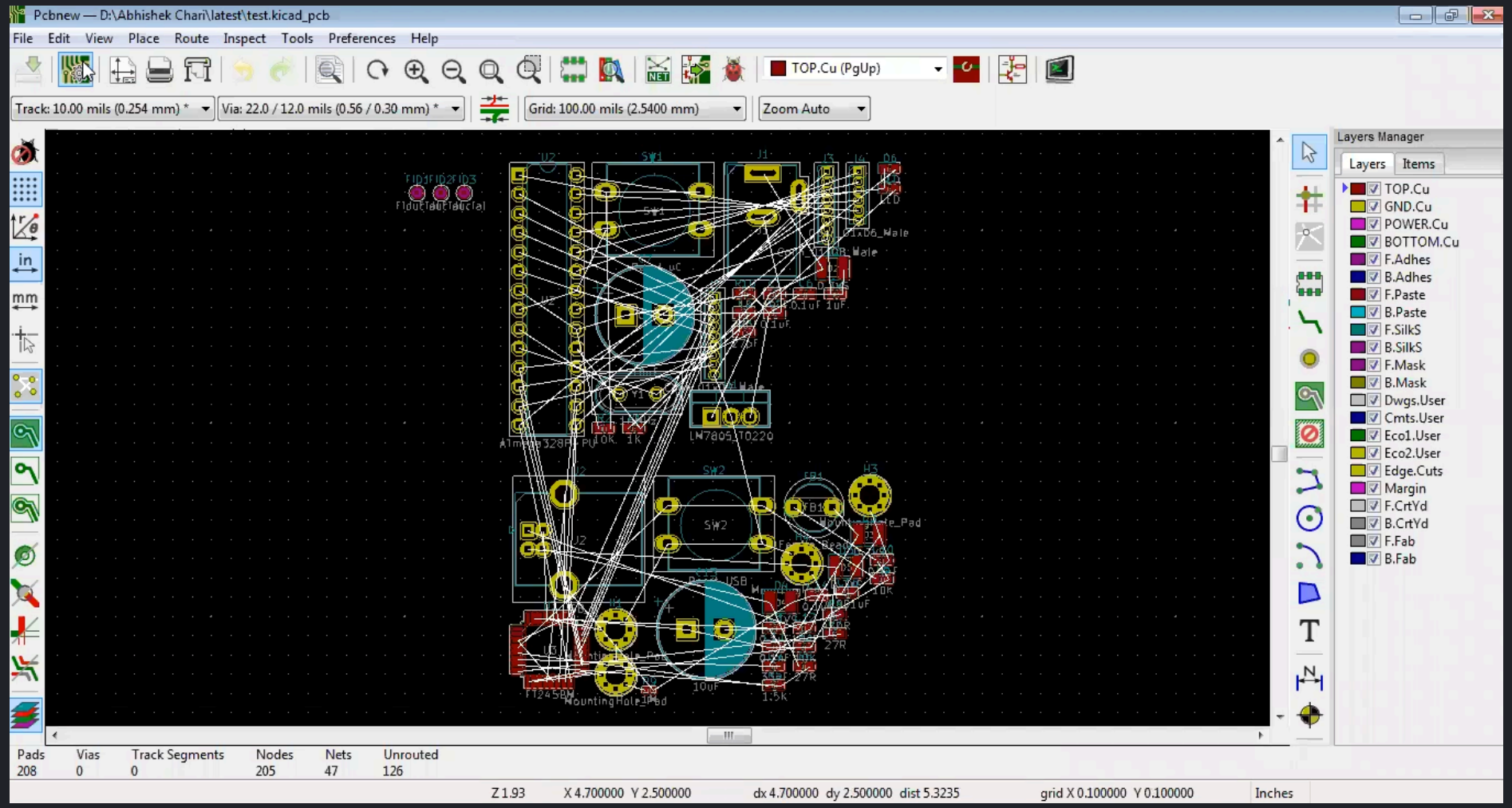
- Une ou plusieurs **couches de cuivre**;
- Des ***vias*** permettant d'interconnecter des portions de couches de cuivre;
- Du **vernis d'épargne (*soldermask*)** qui permet d'isoler les parties de cuivre qui n'ont pas besoin d'être soudées;
- Des **pastilles** destinées aux soudures;
- De la **sérigraphie**, pour indiquer l'emplacement et l'orientation des composants (optionnelle, certains circuits n'en possèdent pas).

Processus de conception

Etape 1: schématique



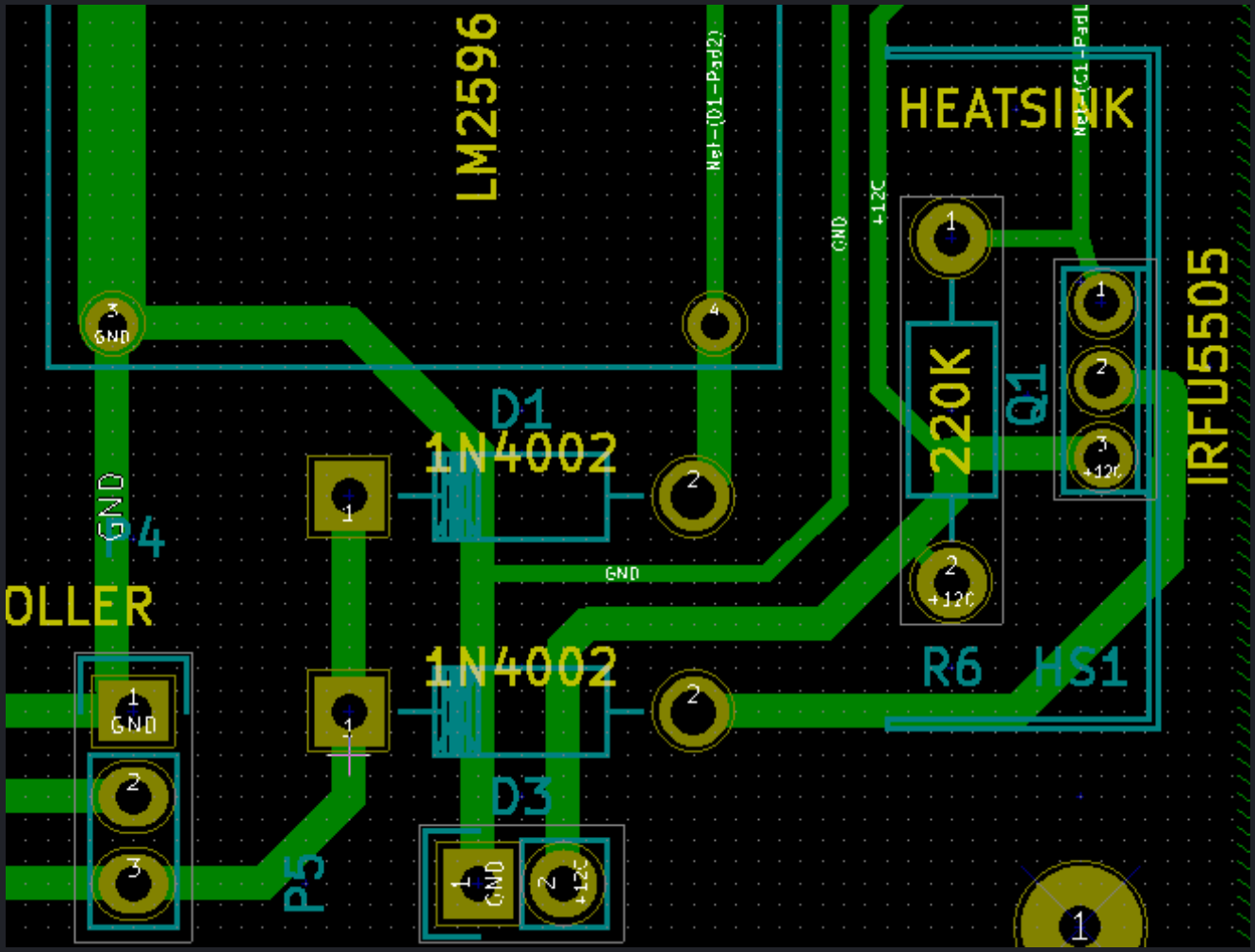
Etape 2: placement des composants



Bonnes pratiques de placement

- **Connecteurs** idéalement **en bordure** de PCB;
- Les composants d'une même fonction sont **groupés**;
- Les condensateurs de découplage sont **placés au plus près** de leurs sources;
- Attention à l'encombrement et penser à l'ordre lors de l'assemblage.

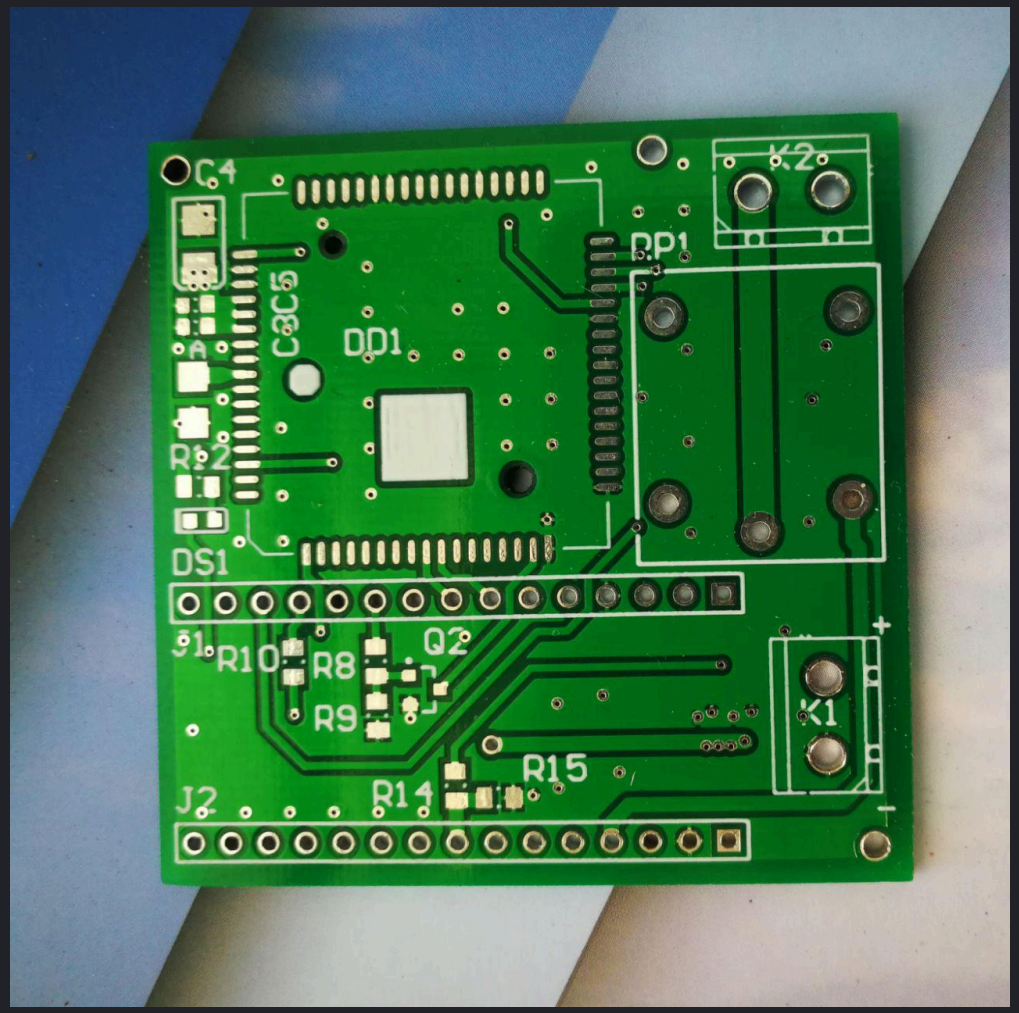
Etape 3: routage



Bonnes pratiques de routage

- **Pistes larges** pour tensions/courants élevés;
- Utilisation d'un **plan de masse** pour faciliter le routage, éliminer les parasites et dissiper la chaleur;
- **Isolation galvanique** en cas d'étage d'adaptation de tension;
- **Routage groupé** des pistes associées à une fonction précise.

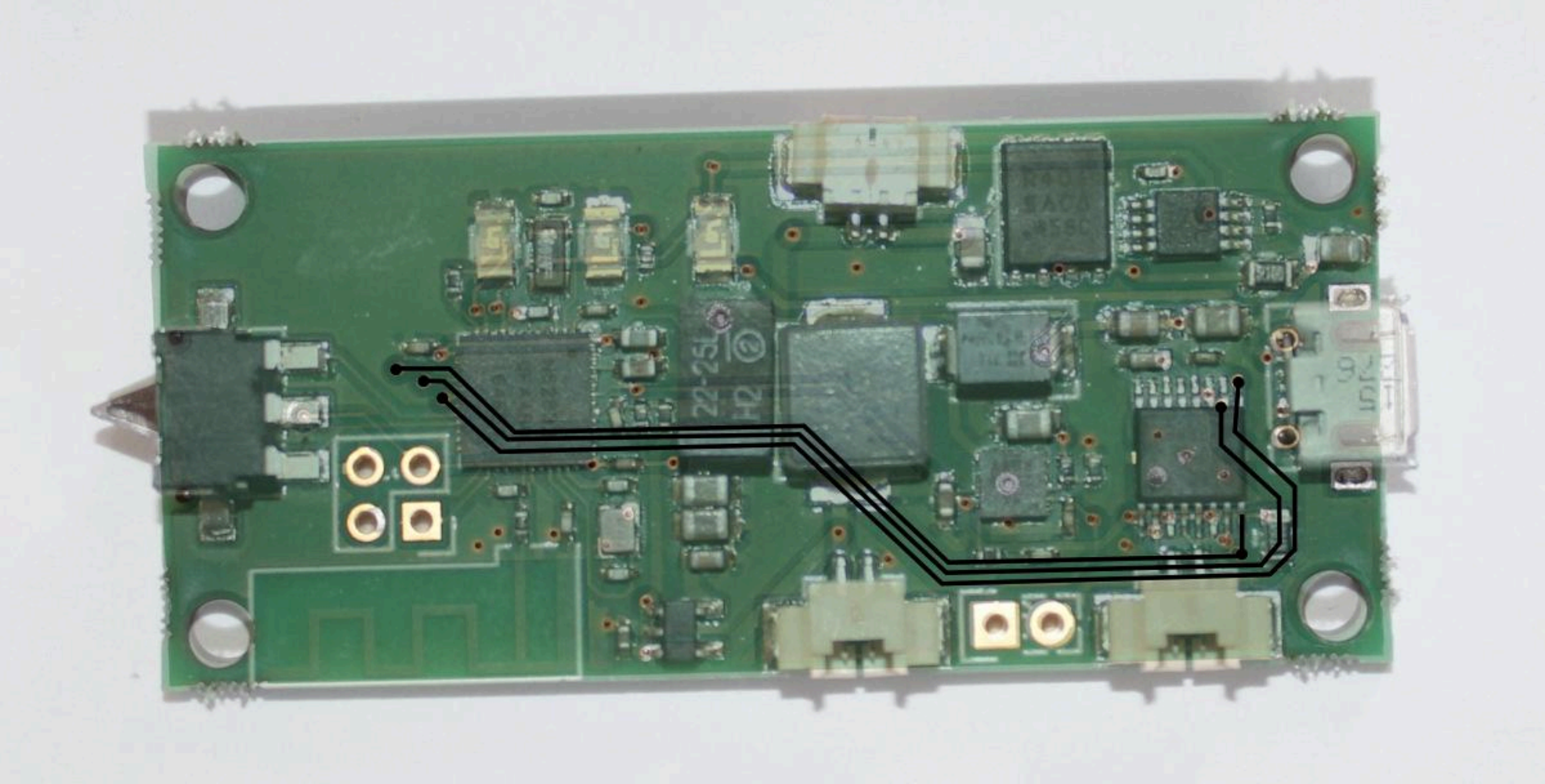
Etape 4: fabrication



Rétro-ingénierie par test de continuité

- Utilisation d'un **multimètre** avec mode *test de continuité* (avec alerte sonore) et **sondes fines**;
- Pratique pour rechercher les **connexions entre connecteurs** et circuits intégrés;
- Assez **laborieux**, sensible aux composants passifs.

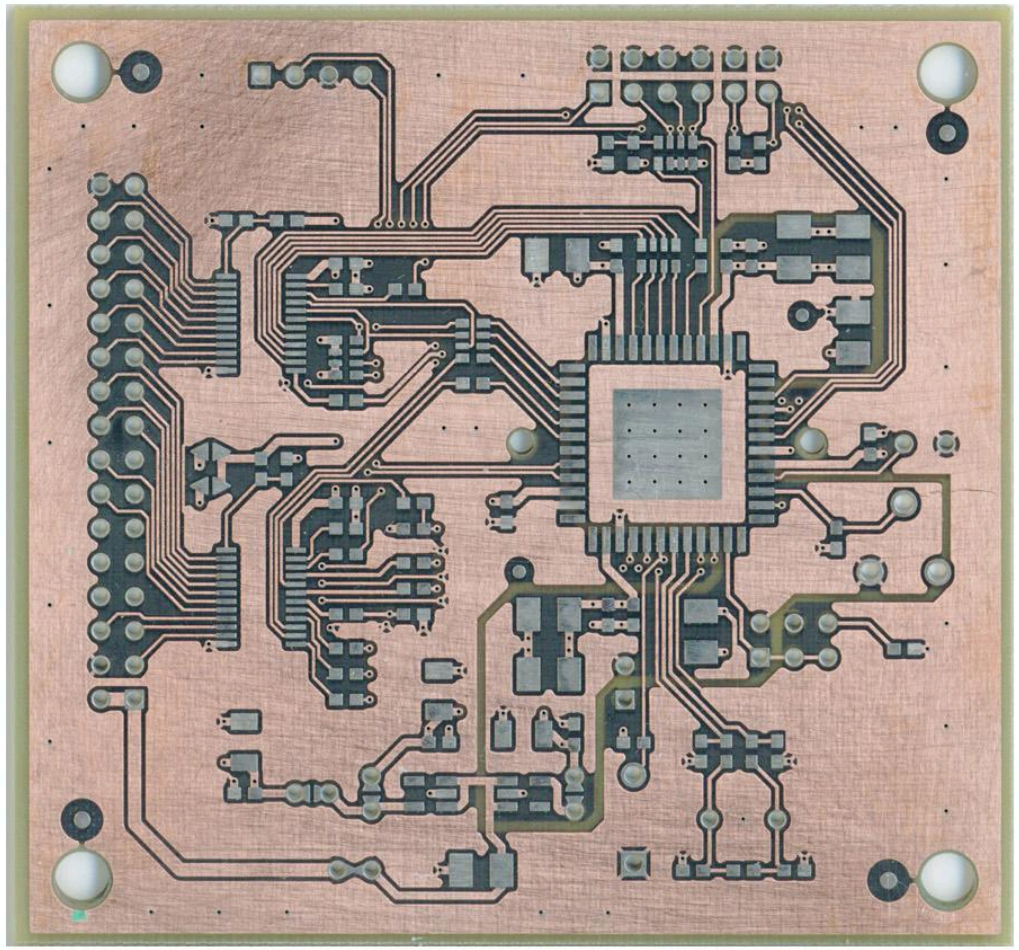
Rétro-ingénierie par photo

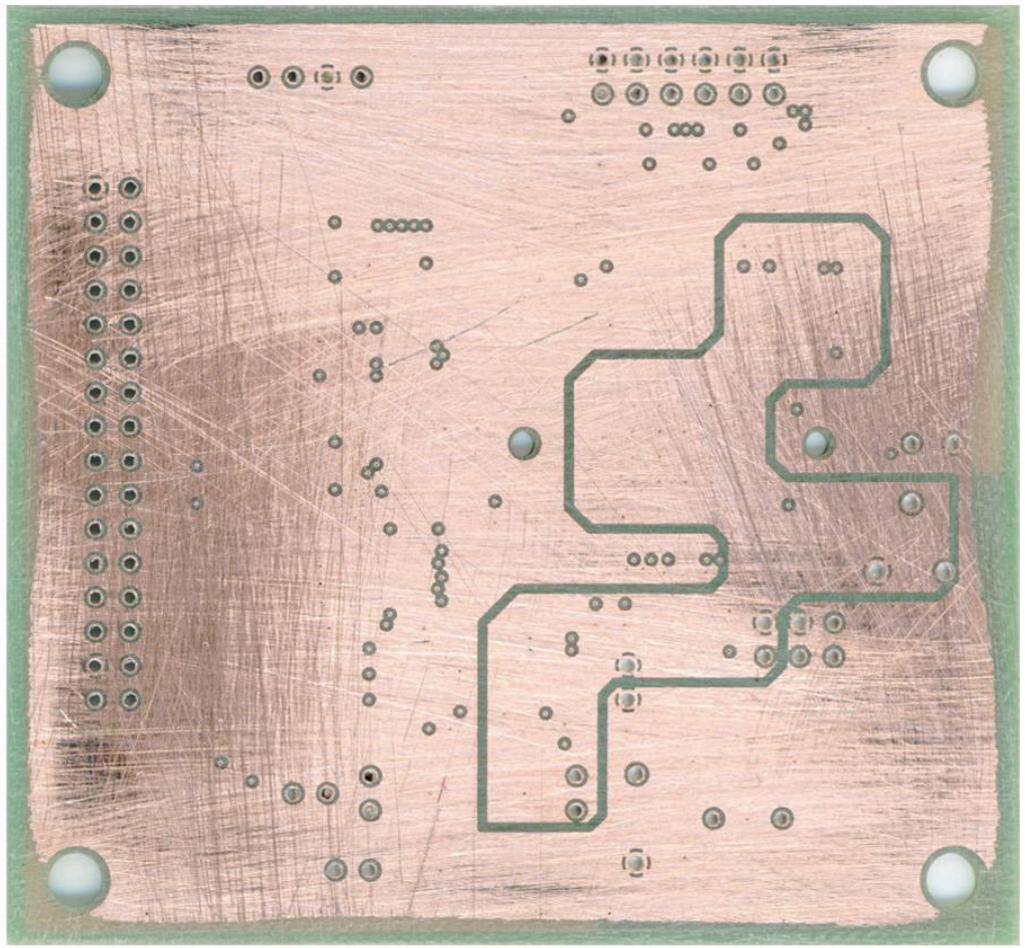


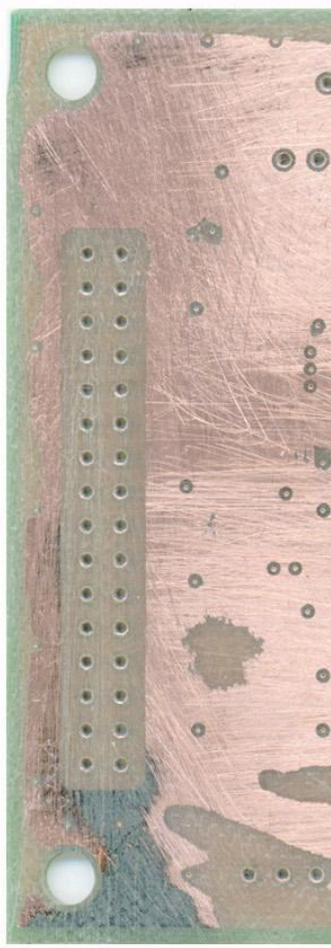
Rétro-ingénierie par delayering

source: PoC||GTFO 6

Rétro-ingénierie par *delayering*

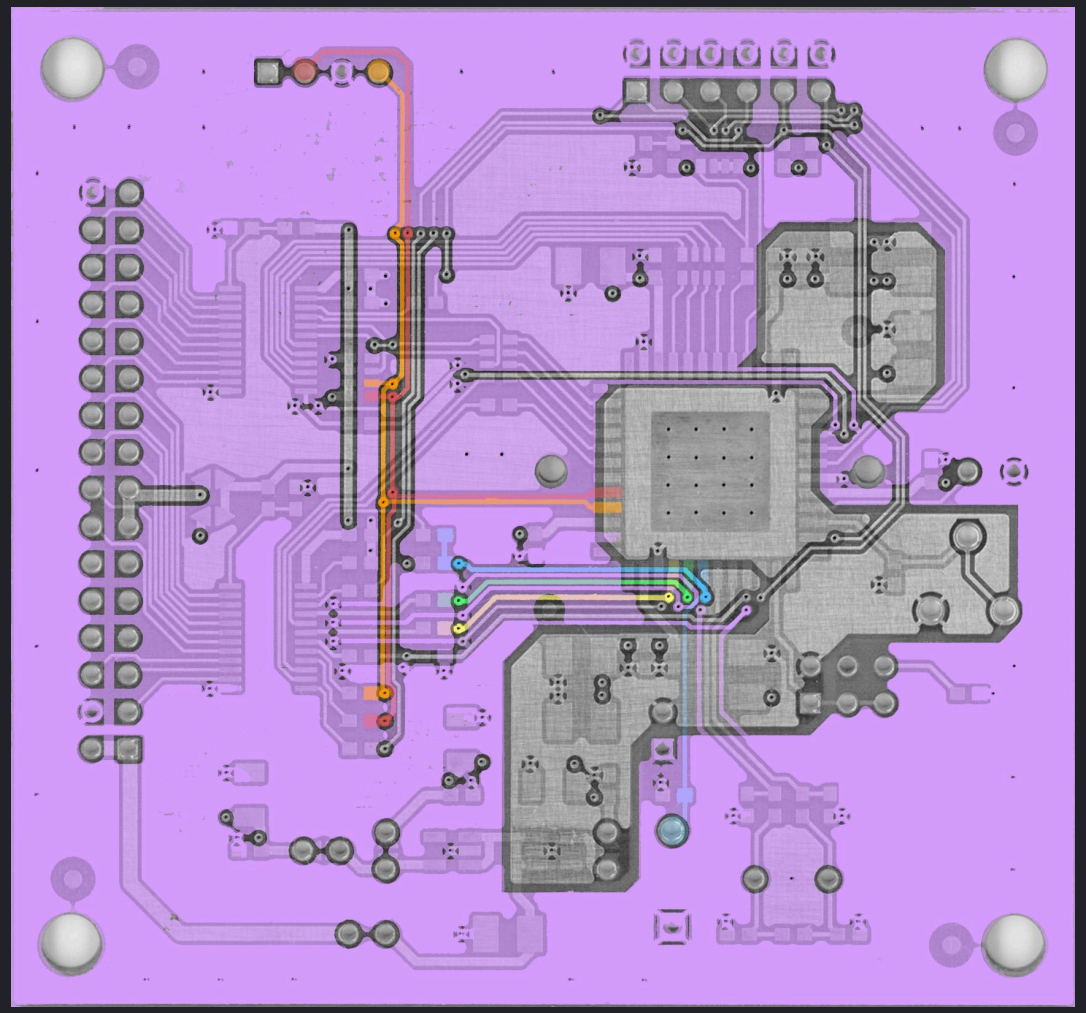






source: PoC||GTFO 6

Rétro-ingénierie par *delayering*

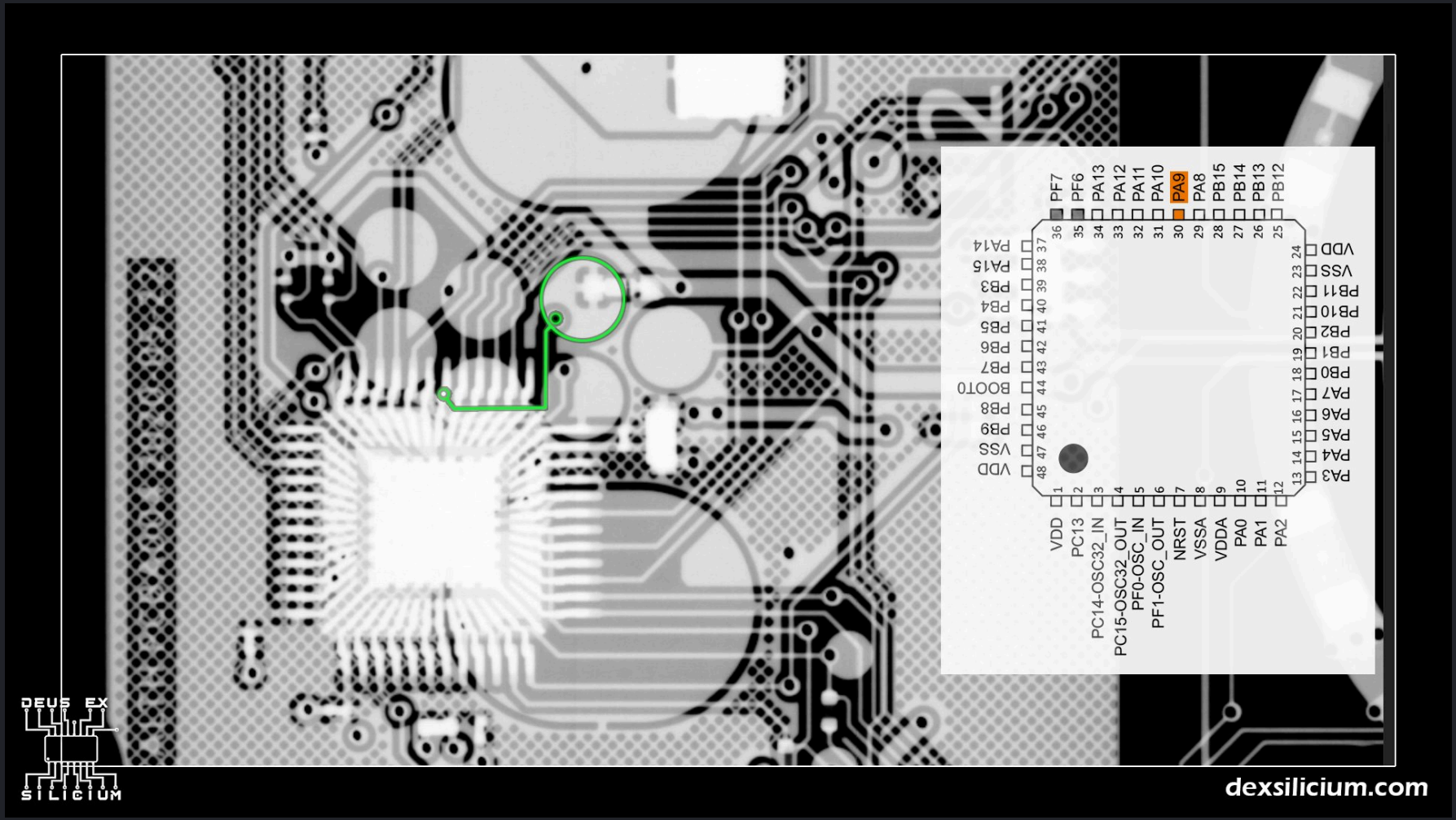


source: PoC||GTFO 6

Rétro-ingénierie par rayons X

source: Deus Ex Silicium

Rétro-ingénierie par rayons X



source: Deus Ex Silicium

Analyse du circuit

- **Analyse globale du circuit** (par fonction) dans un premier temps;
- Puis analyse des différents composants de chaque bloc identifié;
- **Identification des composants** électroniques non-passifs (circuits intégrés principalement);
- Identification des **interconnexions**;
- Création d'un **schéma de principe**.

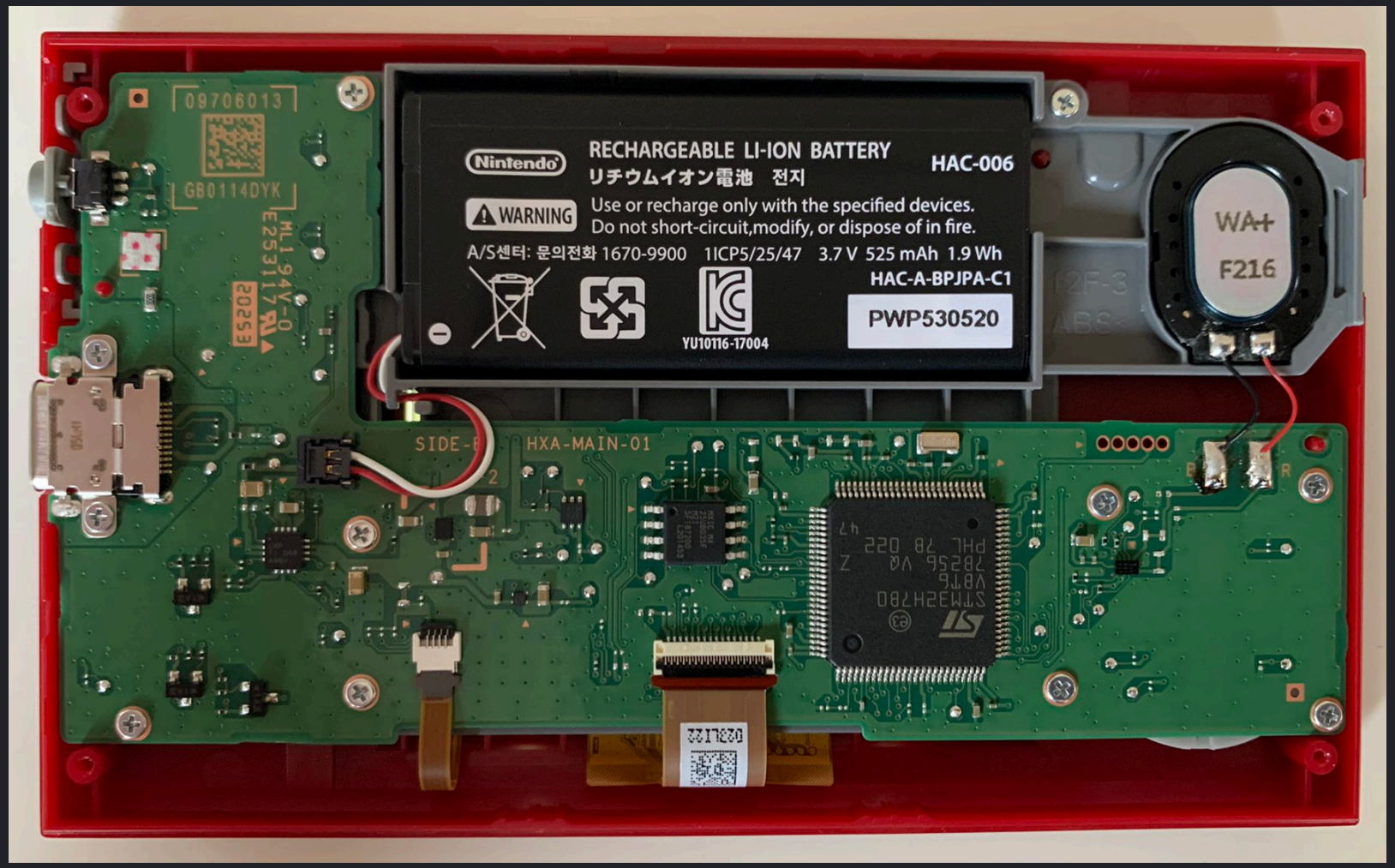
Analyse globale

- Identifier les **fonctions principales** du circuit;
- On se base principalement sur de l'**inspection visuelle**;
- Objectif: **se faire une idée du fonctionnement** du circuit (si possible).

TP

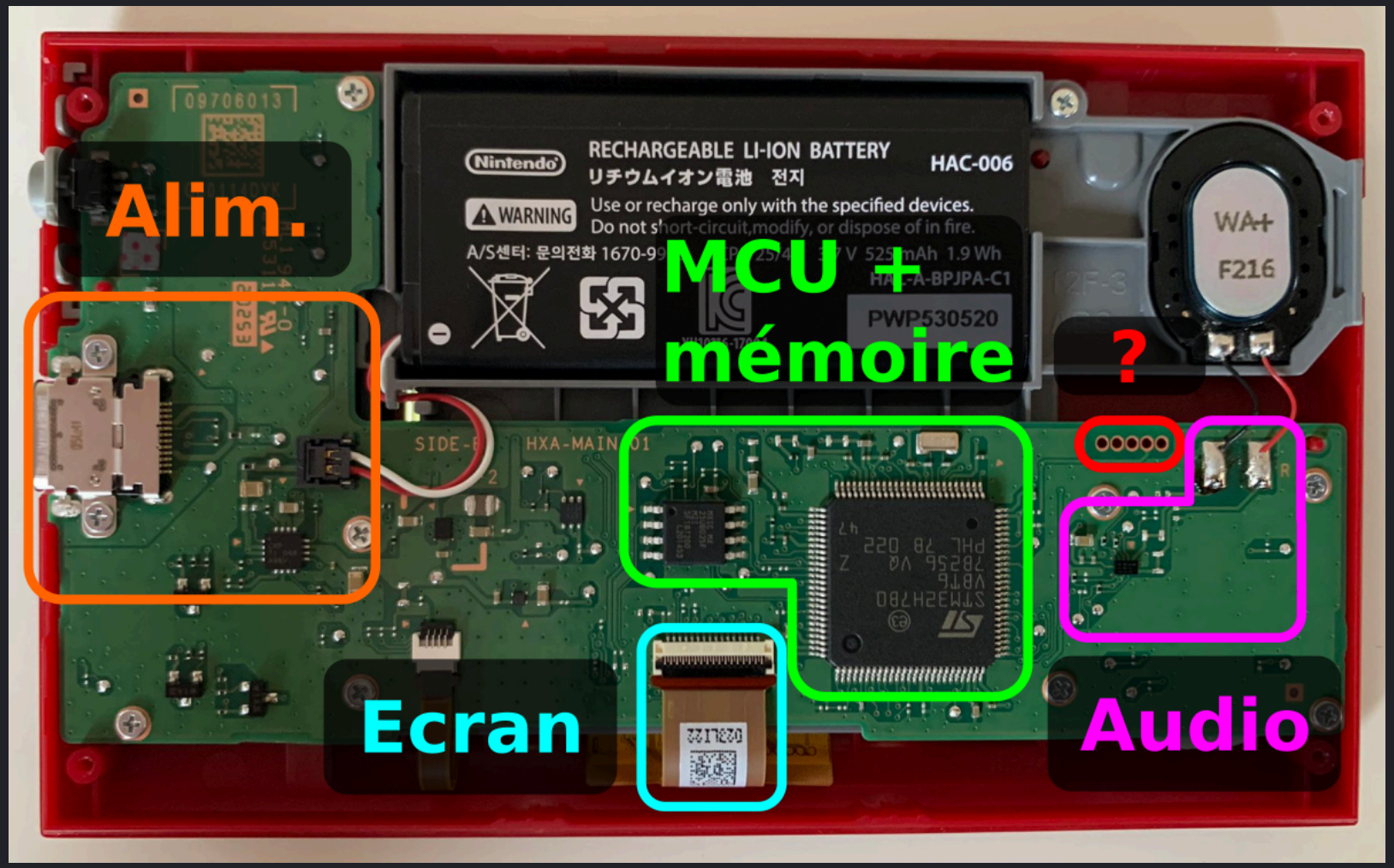
source: https://nintendo.fr

TP

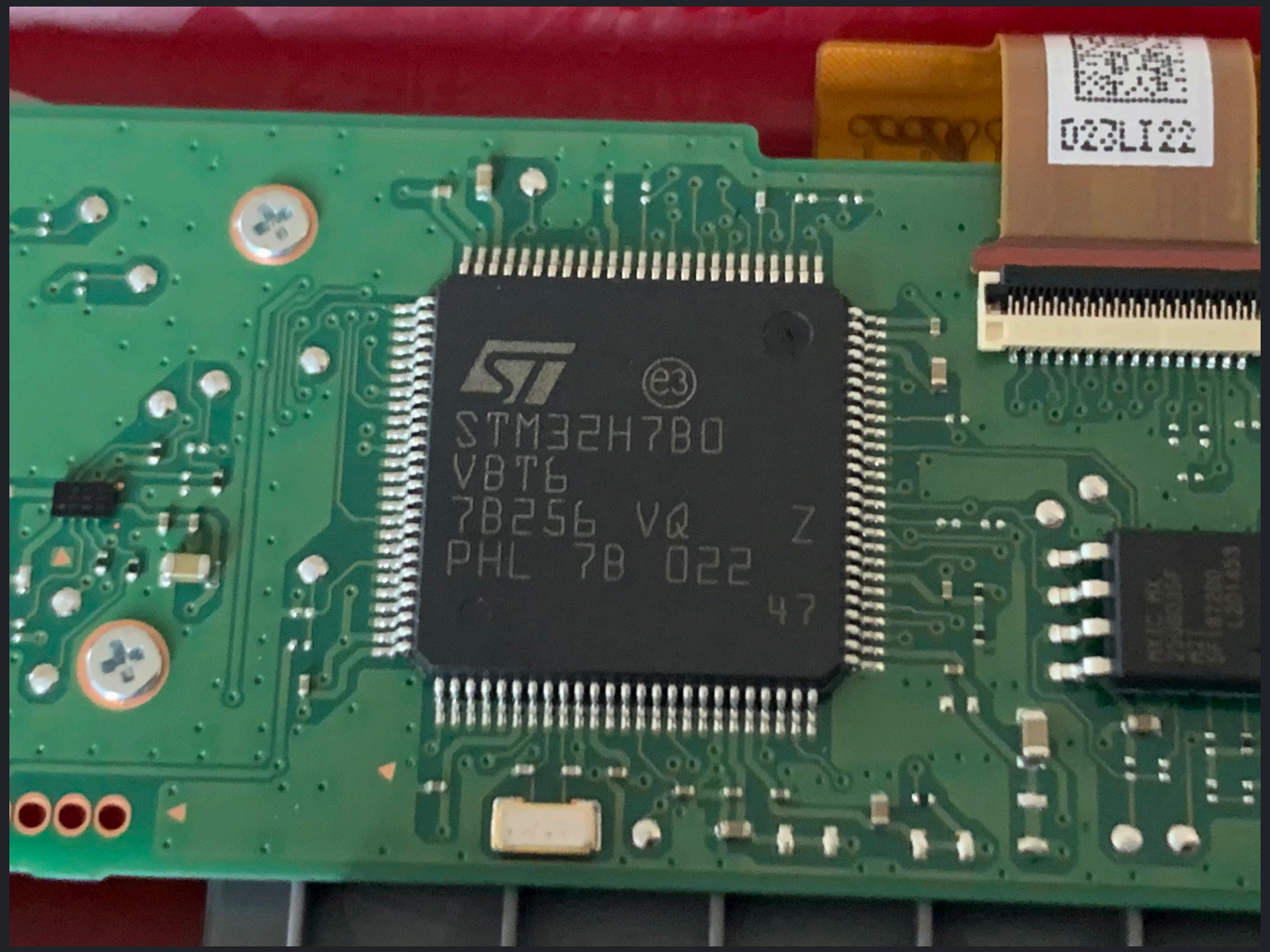


source: @ghidraninja on Twitter

TP


Alim.
MCU + mémoire
?
Ecran
Audio
RECHARGEABLE LI-ION BATTERY
HAC-006
Nintendo
WARNING
Use or recharge only with the specified devices.
PWP530520
WA+
F216
HXA-MAIN-01

Identification des composants



source: @ghidraninja on Twitter

Identification des composants

source: @ghidraninja on Twitter

Identification des interconnexions

- **Identification visuelle** des connexions entre composants (si possible);
- Validation au **multimètre** (si possible);
- Repérage des numéros de broches concernés (si possible).

Limitations: Le suivi des pistes est très difficile sur des PCBs à 4 couches et plus ou si les composants sont en boîtier *Ball Grid Array (BGA)*.

Schéma de principe

- Création d'un **schéma général du circuit**;
- Présente les différents **composants**, leurs **rôles** et les **interactions** supposées;
- Présence des **références** des composants et d'annotations.

Schéma de principe du Game & Watch

TP

- Analysez votre circuit;
- Identifiez les composants;
- Réalisez un schéma de principe.

Lecture de *datasheets*

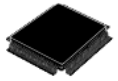
Fonctionnalités

life.augmented

STM32H7B0xB

Datasheet

32-bit Arm® Cortex®-M7 280 MHz MCUs, 128-Kbyte Flash memory, 1.4-Mbyte RAM, 46 com. and analog interfaces, SMPS, crypto



LQFP64
(10 x 10 mm)
LQFP100
(14 x 14 mm)
LQFP144
(20x20 mm)
LQFP176
(24 x 24 mm)

UFBGA169
(7 x 7 mm)
UFBGA176+25
(10x10 mm)

Features

Includes ST state-of-the-art patented technology

Core

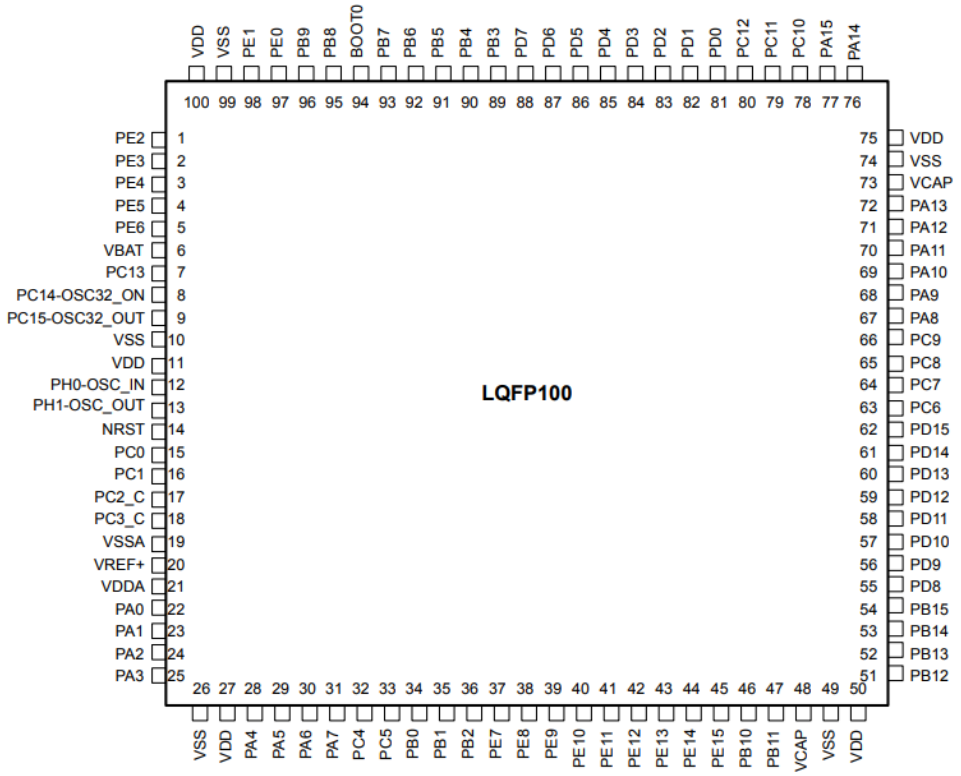
- 32-bit Arm® Cortex®-M7 core with double-precision FPU and L1 cache: 16 Kbytes of data and 16 Kbytes of instruction cache allowing to fill one cache line in a single access from the 128-bit embedded Flash memory; frequency up to 280 MHz, MPU, 599 DMIPS/ 2.14 DMIPS/MHz (Dhrystone 2.1), and DSP instructions

Memories

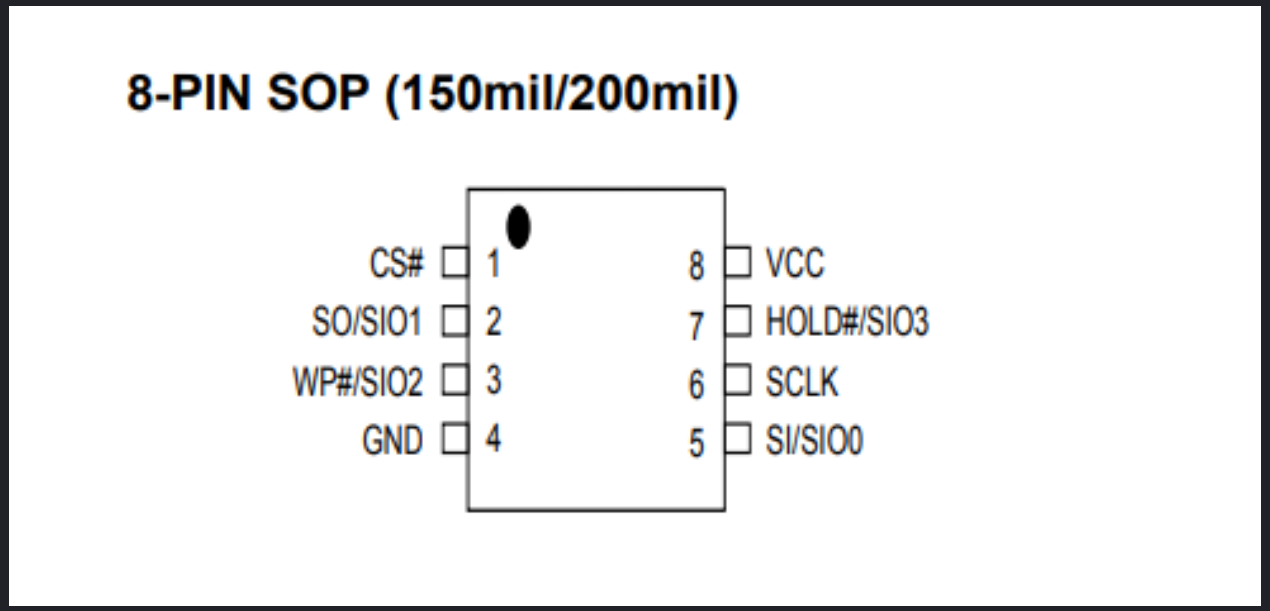
- 128 Kbytes of Flash memory plus 1 Kbyte of OTP memory
- ~1.4 Mbytes of RAM: 192 Kbytes of TCM RAM (inc. 64 Kbytes of ITCM RAM + 128 Kbytes of DTCM RAM for time critical routines), 1.18 Mbytes of user SRAM, and 4 Kbytes of SRAM in Backup domain
- 2x Octo-SPI memory interfaces with on-the-fly decryption, I/O multiplexing and support for serial PSRAM/NOR, Hyper RAM/Flash frame formats, running up to 140 MHz in SRD mode and up to 110 MHz in DTR mode
- Flexible external memory controller with up to 32-bit data bus:

Brochage

Figure 5. LQFP100 (STM32H7B0xB without SMPS) pinout



Brochage



- Nom de broche précédé de *N* ou suivi de # ou / = **état complémenté**;
- Un **détrompeur** présent sur les puces indique l'emplacement de la broche 1.

Rôle des broches

Pin/ball name[1] [2]						Pin name (function after reset)	Pin type	I/O structure	Alternate functions	Additional functions
UFBGA169 with SMPS	UFBGA176+25 with SMPS	LQFP64	LQFP100	LQFP144	LQFP176					
D2	A15	-	-	-	-	VSS	S	-	-	-
-	-	-	-	-	7	PI8	I/O	FT	EVENTOUT	TAMP_IN2/ TAMP_OUT3, RTC_OUT2, WKUP4
D3	C1	2	7	7	8	PC13	I/O	FT	EVENTOUT	TAMP_IN1/ TAMP_OUT2/ TAMP_OUT3, RTC_OUT1/RTC_TS, WKUP3
-	C2	-	-	-	-	VSS	S	-	-	-
C1	D2	3	8	8	9	PC14-OSC32_IN (OSC32_IN)	I/O	FT	EVENTOUT	OSC32_IN
B1	D1	4	9	9	10	PC15-OSC32_OUT (OSC32_OUT)	I/O	FT	EVENTOUT	OSC32_OUT
-	-	-	-	-	11	PI9	I/O	FT_h2	OCTOSPIM_P2_IO0, UART4_RX, FDCAN1_RX, FMC_D30, LCD_VSYNC, EVENTOUT	-
-	-	-	-	-	12	PI10	I/O	FT_h2	OCTOSPIM_P2_IO1, FMC_D31, PSSI_D14, LCD_HSYNC, EVENTOUT	-
-	-	-	-	-	13	PI11	I/O	FT	OCTOSPIM_P2_IO2, LCD_G6, OTG_HS_ULPI_DIR, PSSI_D15, EVENTOUT	WKUP5

Valeurs nominales

Table 20. General operating conditions

Symbol	Parameter	Operating conditions	Min	Typ	Max	Unit
V_{DD}	Standard operating voltage	-	1.62[1]	-	3.6	
V_{DDLDO}	Supply voltage for the internal regulator	VDDLDO ≤ VDD	1.62[1]	-	3.6	
			1.2[2]	-	3.6	
V_{DDSMPS}	Supply voltage for the internal SMPS Step-down converter	VDDSMPS = VDD	1.62[1]	-	3.6	
V_{DDMMC}	Standard operating voltage for independent MMC I/Os	Indenpent MMC I/Os used	1.62[1]	-	3.6	
		Independent MMC I/Os not used V_{DDMMC} = V_{DD}	1.62[1]	-	3.6	
$V_{DD33USB}$	Standard operating voltage, USB domain	USB used	3.0	-	3.6	
		USB not used	0	-	3.6	
V_{DDA}	Analog operating voltage	ADC or COMP used	1.62	-	3.6	
		DAC used	1.8	-		
		OPAMP used	2.0	-		
		VREFBUF used	1.8	-		
		ADC, DAC, OPAMP,				

Absolute maximum ratings

6.2 Absolute maximum ratings

Stresses above the absolute maximum ratings listed in Table 17. Voltage characteristics, Table 18. Current characteristics, and Table 19. Thermal characteristics may cause permanent damage to the device. These are stress ratings only and the functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability. Device mission profile (application conditions) is compliant with JEDEC JESD47 Qualification Standard, extended mission profiles are available on demand.

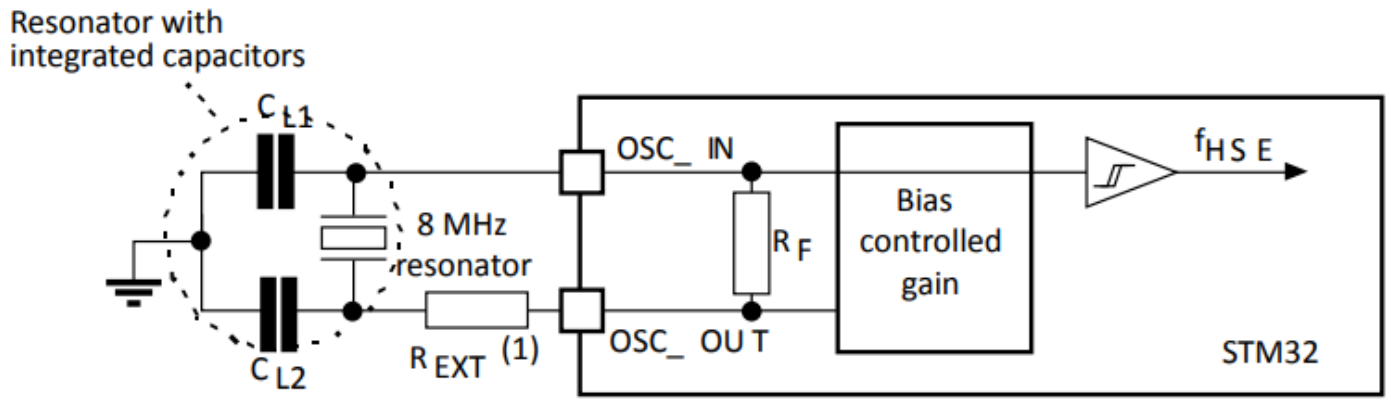
Table 17. Voltage characteristics

All main power (V_{DD}, V_{DDA}, $V_{DD33USB}$, V_{DDMMC}, V_{DDSMPS}, V_{BAT}) and ground (V_{SS}, V_{SSA}) pins must always be connected to the external power supply, in the permitted range.

Symbols	Ratings	Min	Max	Unit
$V_{DDX} - V_{SS}$	External main supply voltage (including V_{DD}, V_{DDLDO}, V_{DDSMPS}, V_{DDA}, $V_{DD33USB}$, V_{DDMMC}, V_{BAT}, V_{REF+})	−0.3	4.0	V
V_{IN}[1]	Input voltage on FT_xxx pins	V_{SS}−0.3	Min(V_{DD}, V_{DDA}, $V_{DD33USB}$, V_{DDMMC}, V_{BAT}) +4.0[2][3]	V
	Input voltage on TT_xx pins	V_{SS}−0.3	4.0	V
	Input voltage on BOOT0 pin	V_{SS}	9.0	V
	Input voltage on any other pins	V_{SS}−0.3	4.0	V
$\lvert\Delta V_{DDX}\rvert$	Variations between different V_{DDX} power pins of the same domain	-	50	mV

Schémas de référence

Figure 23. Typical application with an 8 MHz crystal



Schémas de référence

11.7.3 CEAA WLCSP schematic with DC/DC converter setup

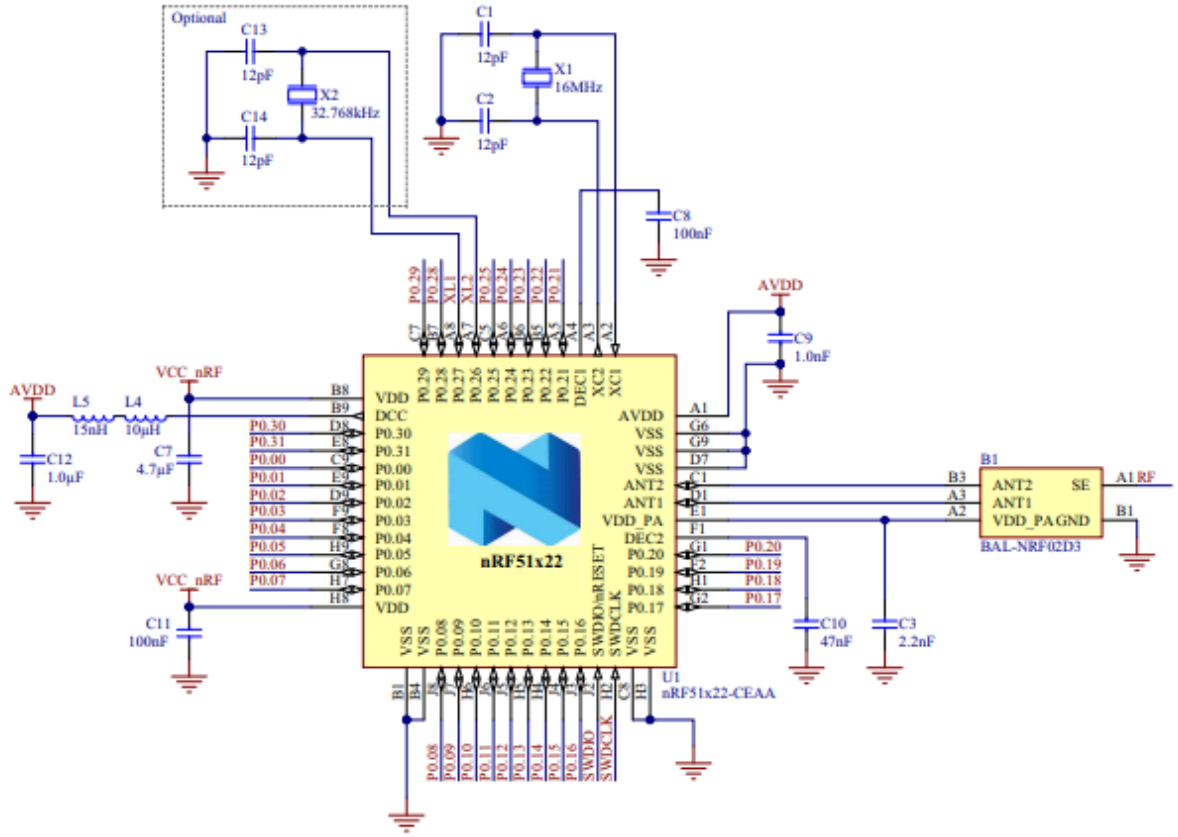
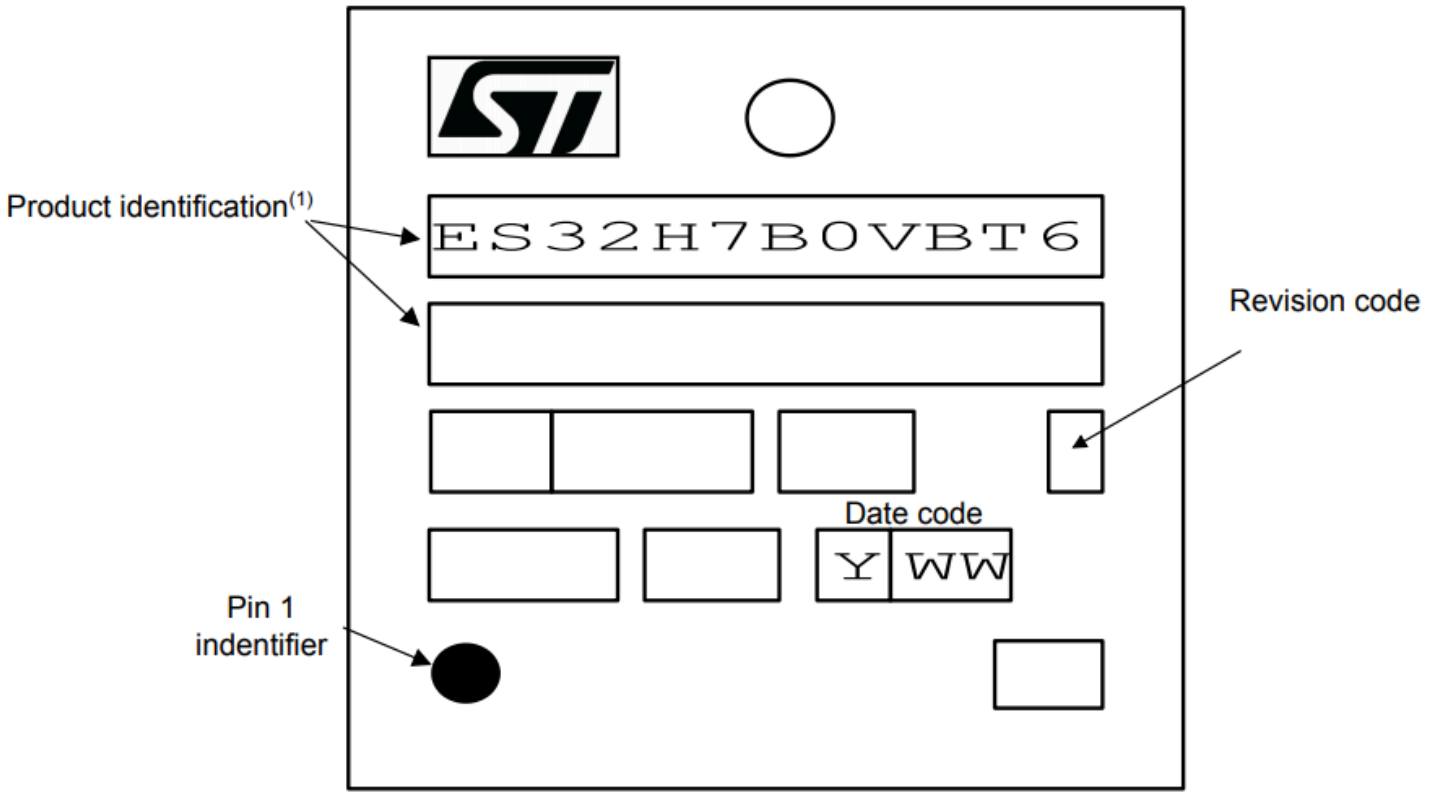


Figure 36 CEAA WLCSP with DC/DC converter setup

Marquages

Figure 78. LQFP100 marking example (package top view)



Datasheet vs. Reference Manual

- La *datasheet* contient les **spécifications matérielles** d'un composant (brochage, périphériques internes, mémoire);
- Elle n'est pas faite pour indiquer l'utilisation des différents périphériques;
- Certaines données sont volontairement manquantes, car présentes dans un manuel de référence (ou ***Reference Manual***)

Que trouve-t-on dans un *Reference Manual* ?

- Le détail des **éléments internes** d'un SoC (par exemple): mapping mémoire des registres de périphériques, machines à états, interconnexions, interruptions, etc.;
- Les **procédures standards** pour piloter les différents périphériques;
- Plus généralement, tout ce dont le développeur a besoin pour **créer du logiciel** pour la puce ciblée.
- Le *Reference Manual* est généralement **très dense** (plusieurs milliers de pages en moyenne) !

Exemple de manuel de référence

Table 6. Memory map and default device memory area attributes

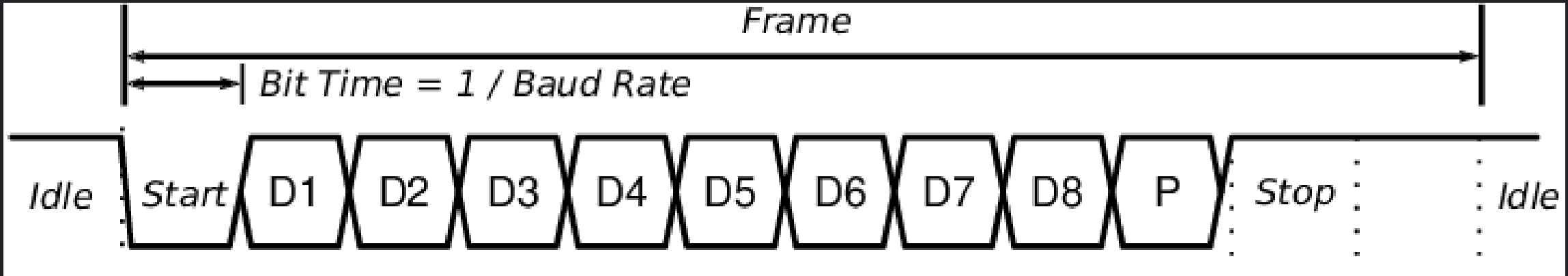
Region	Boundary address	Arm® Cortex®-M7	Type	Attributes	Execute never
External devices	0xD000 0000 - 0xDFFF FFFF	FMC SDRAM Bank2 (or reserved in case of FMC remap)	Device	-	Yes
	0xCC00 0000 - 0xCFFF FFFF	FMC SDRAM Bank1 (or remap of FMC NOR/PSRAM/SRAM 4 Bank1)			
	0xC800 0000 - 0xCBFF FFFF	FMC SDRAM Bank1 (or remap of FMC NOR/PSRAM/SRAM 3 Bank1)			
	0xC400 0000 - 0xC7FF FFFF	FMC SDRAM Bank1 (or remap of FMC NOR/PSRAM/SRAM 2 Bank1)			
	0xC000 0000 - 0xC3FF FFFF	FMC SDRAM Bank1 (or remap of FMC NOR/PSRAM/SRAM 1 Bank1)			
	0xA000 0000 - 0xBFFF FFFF	Reserved			
External memories	0x9000 0000 - 0x9FFF FFFF	OCTOSPI1	Normal	Write-through cache attribute	No
	0x8000 0000 - 0x8FFF FFFF	FMC NAND Flash memory			
	0x7000 0000 - 0x7FFF FFFF	OCTOSPI2			
	0x6C00 0000 - 0x6FFF FFFF	FMC NOR/PSRAM/SRAM 4 Bank1 (or remap of FMC SDRAM Bank1)		Write-back, write allocate cache attribute	
	0x6800 0000 - 0x6BFF FFFF	FMC NOR/PSRAM/SRAM 3 Bank1 (or remap of FMC SDRAM Bank1)			
	0x6400 0000 - 0x67FF FFFF	FMC NOR/PSRAM/SRAM 2 Bank1 (or remap of FMC SDRAM Bank1)			
	0x6000 0000 - 0x63FF FFFF	FMC NOR/PSRAM/SRAM 1 Bank1 (or remap of FMC SDRAM Bank1)			
Peripherals	0x4000 0000 - 0x5FFF FFFF	Peripherals (refer to *Table 7: Register boundary addresses*)	Device	-	Yes

Protocoles de communication

Protocole USART

- **U**niversal **S**ynchronous & **A**synchronous **R**eceiver/**T**ransmitter
- **Protocole série** électronique d'envoi et de réception de données
- **Caractéristiques**:
  - Vitesse de transmission en bauds;
  - Nombre de bits utilisés pour coder un symbole;
  - Parité;
- Aussi nommé **UART**, car peu utilisé en mode synchrone.

Chronogrammes UART



- L'envoi d'une série d'octets débute par un *start bit*;
- Les bits sont envoyés **du bit de poids faible** (bit 0) **au bit de poids fort** (bit 7 dans le cas d'un codage sur 8 bits);
- Le **bit de parité** est optionnel (cas de la parité *null*);
- Le **bit de stop** indique la fin de la transaction.
- Les caractéristiques sont notées de la façon suivante: **9600 8N1**

Détection de broches TX UART

Il est possible de détecter la fluctuation de niveau logique d'une broche d'un composant à l'aide d'un multimètre:

- Le multimètre en mode voltmètre affiche **la moyenne d'une série de mesures** effectuées;
- Si la tension mesurée varie entre 0 et Vcc (ce qui est le cas d'une broche TX UART), alors la tension affichée devrait être **aux alentours de Vcc/2** lorsqu'un envoi est effectué;
- Il est alors possible de confirmer cela avec un **analyseur logique**.

Capture de communications UART

A l'aide d'un analyseur logique, il est possible de capturer une communication UART:

- On connecte la masse de l'analyseur à la masse du circuit cible;
- On connecte la broche Tx sur le premier canal (*Channel 0*);
- On lance la capture ...

Capture de communications UART

Capture de communications UART

Il faut prendre quelques précautions:

Vitesse d'acquisition (*sampling rate*) **au moins 4 fois supérieure** à la vitesse du bus UART;

La **vitesse d'acquisition est réduite** si l'on échantillonne sur plusieurs canaux à la fois;

TP

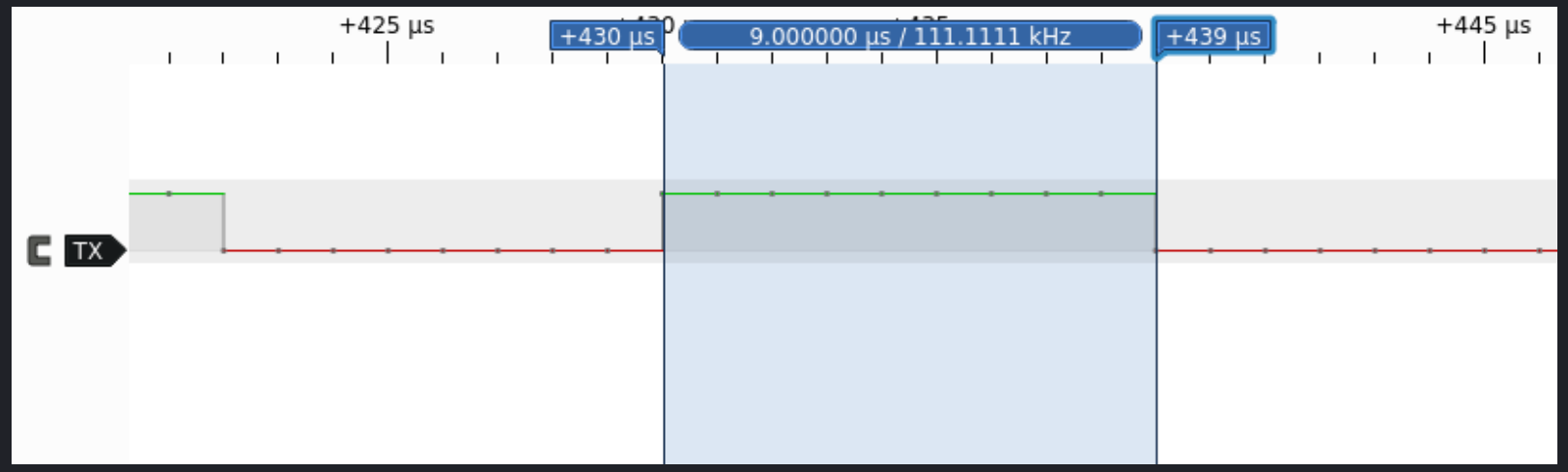
- A l'aide de *pulseview* et de l'analyseur logique, capturez la donnée transmise sur le port UART de la carte;
- Sauvegardez votre acquisition dans un fichier;
- Vérifiez que la donnée acquise correspond au format attendu.

Décodage de communication UART

Pour décoder une communication UART, il faut déterminer au préalable:

- La **vitesse** de transfert (en bauds ou bit/s);
- Le **nombre de bits** codant un symbole (généralement 8)
- La configuration du **bit de parité** (généralement, aucun bit de parité n'est utilisé)

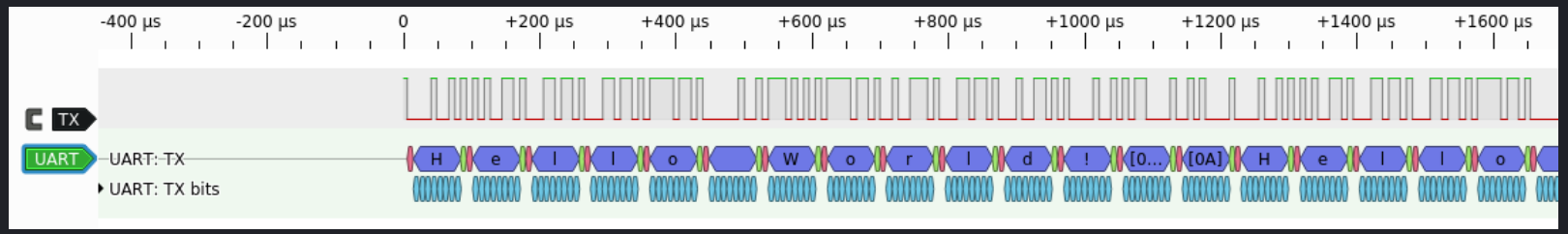
Déterminer la vitesse de transfert



Ici, la vitesse de transfert normalisée la plus proche est **115200**.

TP

Décodez la communication UART précédemment capturée avec **Pulseview**.

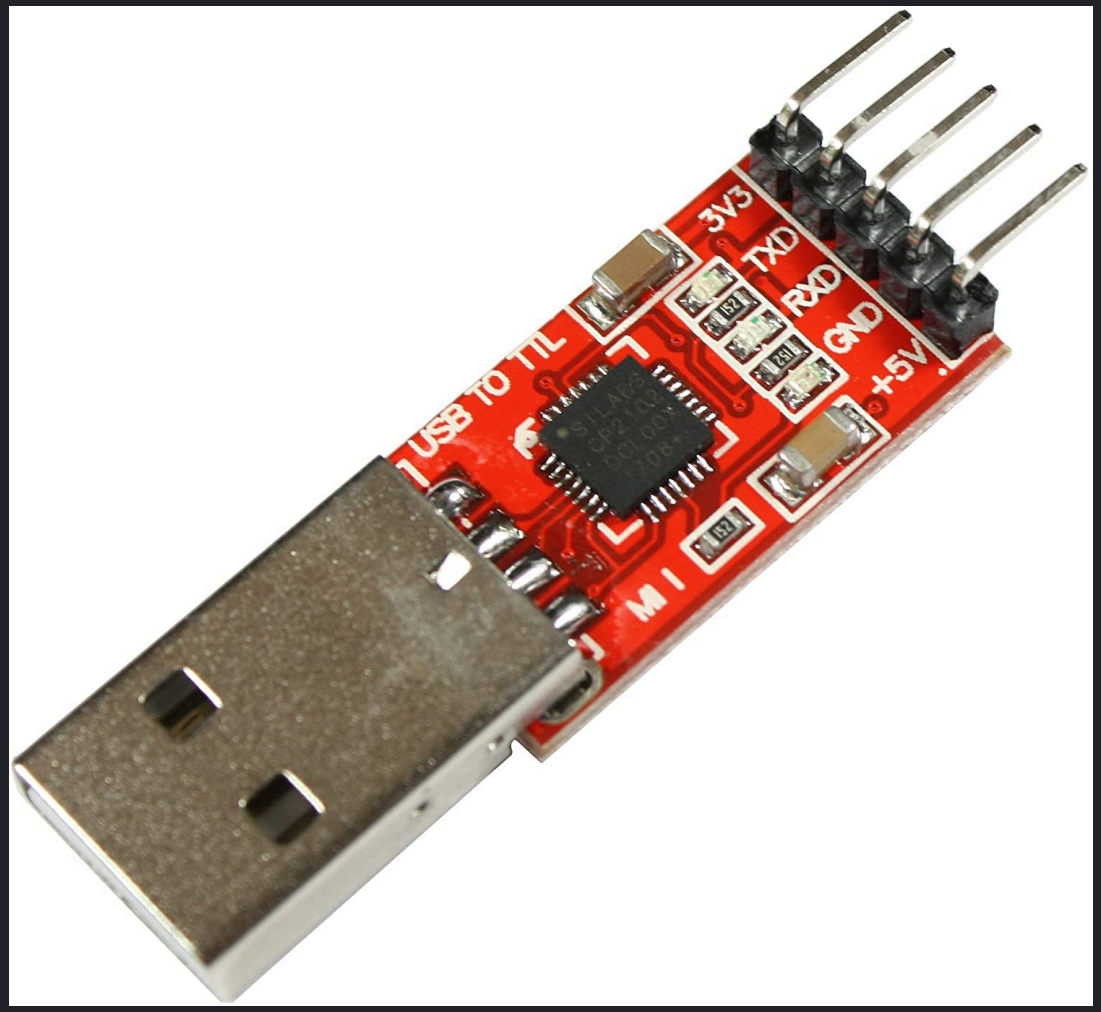


Communiquer avec une interface UART

Pour communiquer avec une interface UART, on utilise un adaptateur USB adéquat:

On utilise ensuite des logiciels dédiés:

- *screen* (Linux)
- *minicom* (Linux)
- *miniterm* (Linux)
- *picocom* (Linux)
- *HyperTerminal* (Windows)
- *Putty* (Linux, Windows)



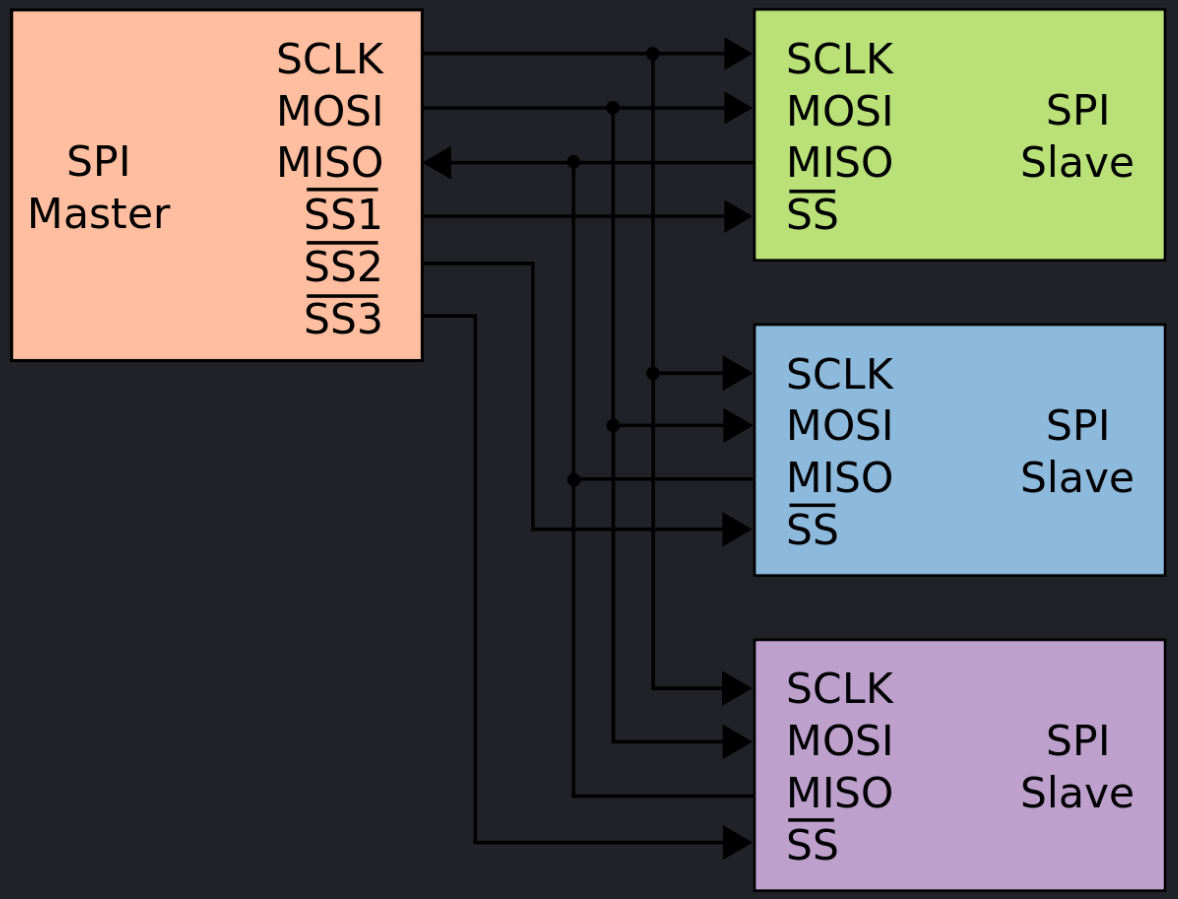
TP

Interfacez votre ordinateur avec le port UART de la carte.

Protocole SPI

- Inventé par **Motorola** dans les années 1980;
- **S**erial **P**eripheral **I**nterface;
- Transfert ***full-duplex***

Protocole SPI



source: Wikipedia

Protocole SPI

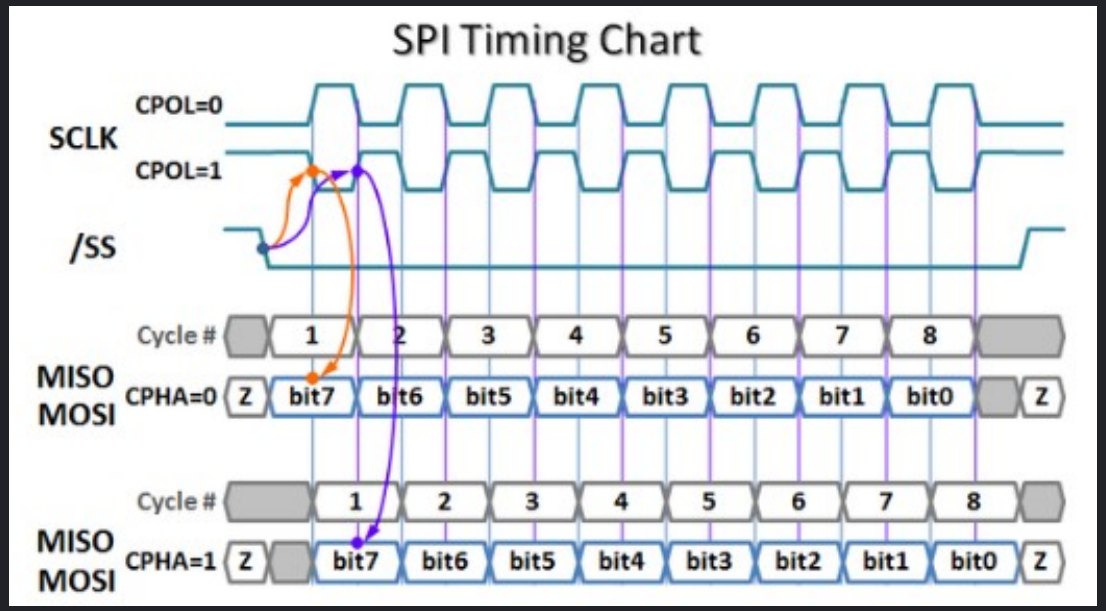
4 signaux principaux:

- **SCLK** (*serial clock*): signal d'horloge permettant d'indiquer au périphérique sélectionné qu'il y a un bit placé sur MOSI
- **MOSI** (*Master Out/Slave In*): donnée en provenance du contrôleur et à destination du périphérique;
- **MISO** (*Master In/Slave Out*): donnée en provenance du périphérique et à destination du contrôleur;
- **CS** (*Chip Select*): généralement complémentée, permet de sélectionner le périphérique avec lequel le contrôleur va communiquer.

Modes de fonctionnement

Le protocole SPI possède 4 modes de fonctionnement, dépendant de deux paramètres:

- La polarité du signal d'horloge (*CPOL*), correspondant à la valeur *par défaut* du signal;
- La phase du signal d'horloge (*CPHA*), correspondant à un type de front.



source: Wikipedia

TP

Capturez et décodez la communication SPI entre le MCU et la mémoire Flash.

Interpréter les transactions SPI

Le protocole *SPI* assure le **transport bidirectionnel** de données, mais la plupart des puces l'employant interprètent à leur façon les données reçues. Tout est généralement indiqué dans la **datasheet**:

10. COMMAND DESCRIPTION

Table 5. Command Set

Read/Write Array Commands

I/O	1	1	2	2	4	4
Command (byte)	READ (normal read)	FAST READ (fast read data)	2READ (2 x I/O read command)	DREAD (1I / 2O read command)	4READ (4 x I/O read)	QREAD (1I/4O read)
1st byte	03 (hex)	0B (hex)	BB (hex)	3B (hex)	EB (hex)	6B (hex)
2nd byte	ADD1	ADD1	ADD1	ADD1	ADD1	ADD1
3rd byte	ADD2	ADD2	ADD2	ADD2	ADD2	ADD2
4th byte	ADD3	ADD3	ADD3	ADD3	ADD3	ADD3
5th byte		Dummy	Dummy	Dummy	Dummy	Dummy
Action	n bytes read out until CS# goes high	n bytes read out until CS# goes high	n bytes read out by 2 x I/O until CS# goes high	n bytes read out by Dual Output until CS# goes high	n bytes read out by 4 x I/O until CS# goes high	n bytes read out by Quad output until CS# goes high

TP

Interprétez les transactions effectuées par le MCU.

Interfaces de Débogage

Débogage de MCU et SoC

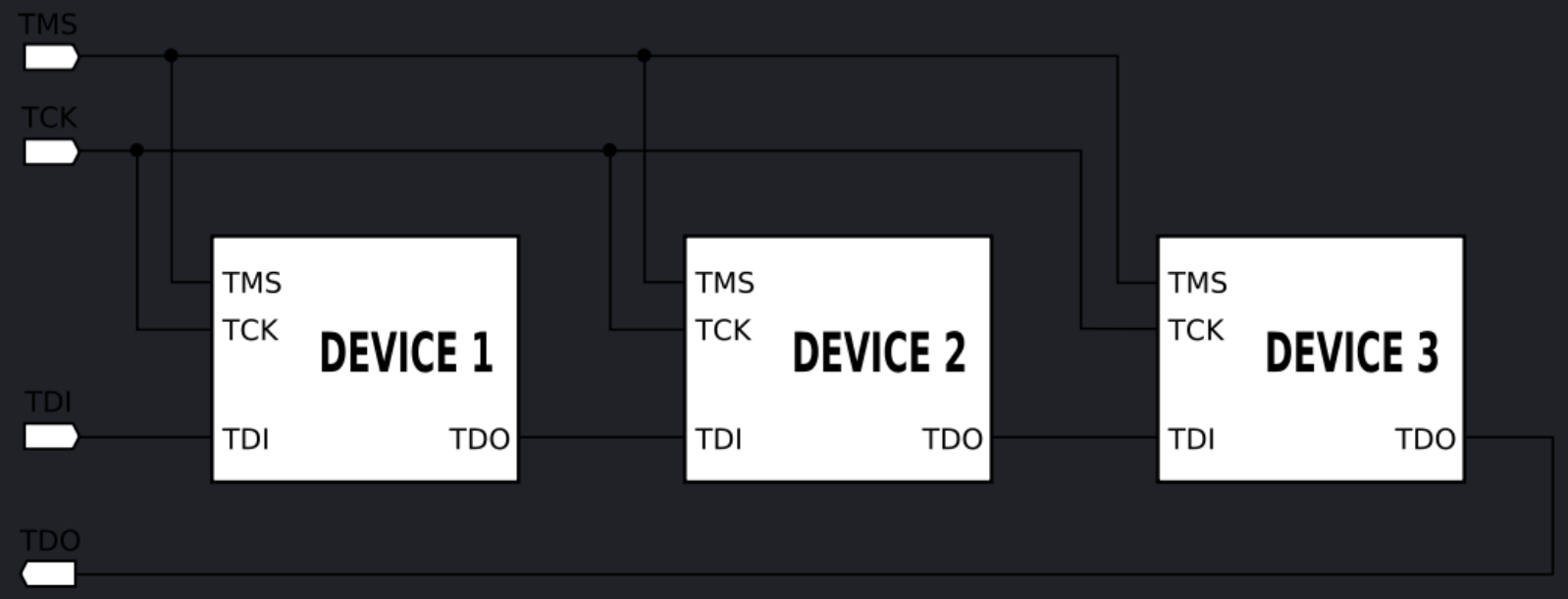
- Le débogage fait partie du **cycle de développement** d'un logiciel embarqué;
- Sur du matériel, il nous faut une **interface physique** sur laquelle se connecter afin de contrôler le ou les CPUs et les différents périphériques;
- Deux interfaces sont très employées: **JTAG** et **SWD**.

Interface JTAG

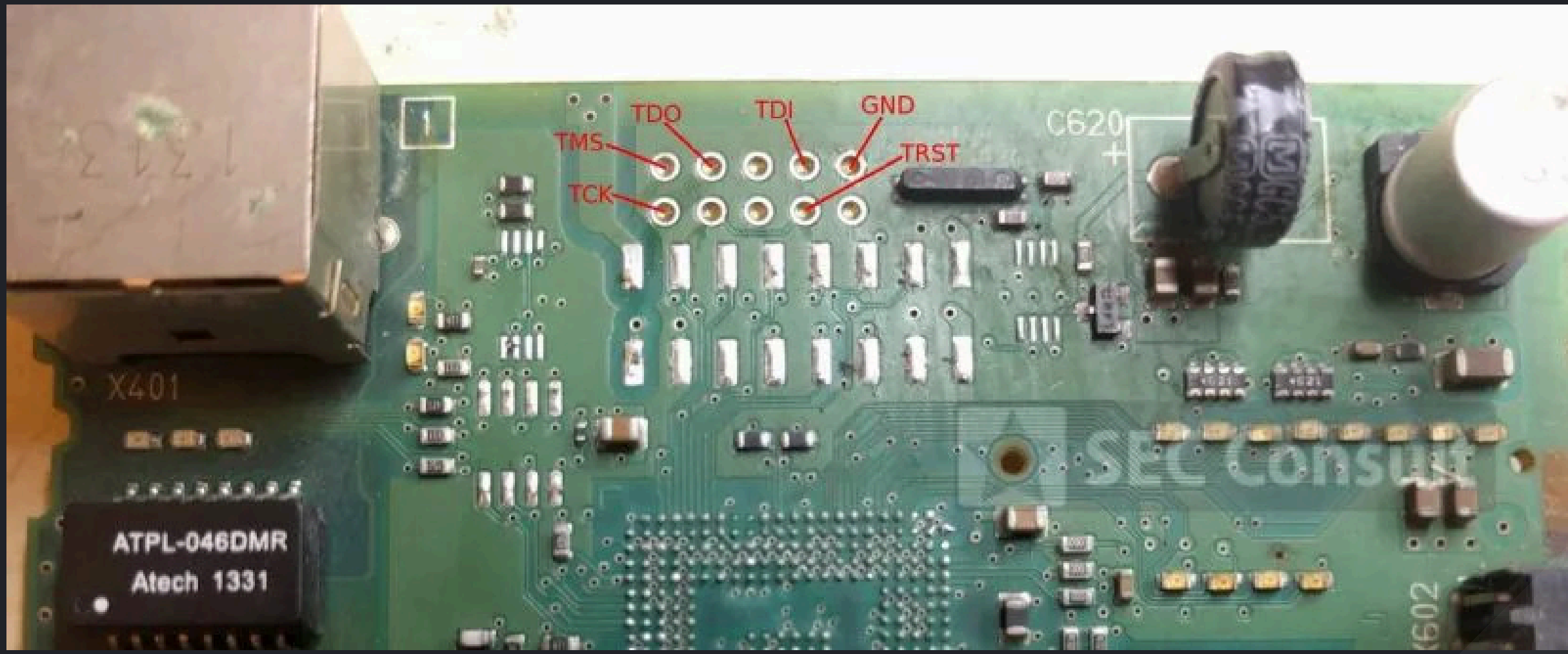
- **J**oint **T**est **A**ction **G**roup
- Interface de **débogage** et de **programmation**
- Plusieurs signaux requis:
  - **JTCK**: signal d'horloge;
  - **JTDI**: réception de données en provenance de l'équipement;
  - **JTDO**: envoi de données à destination de l'équipement;
  - **JTMS**: sélection de mode;
  - **JTRST**: remise à zéro de l'équipement.

Interface JTAG

Les périphériques JTAG peuvent être chaînés, c'est d'ailleurs ce principe qui permet à un MCU ou à un SoC d'exposer son CPU, sa mémoire, etc:



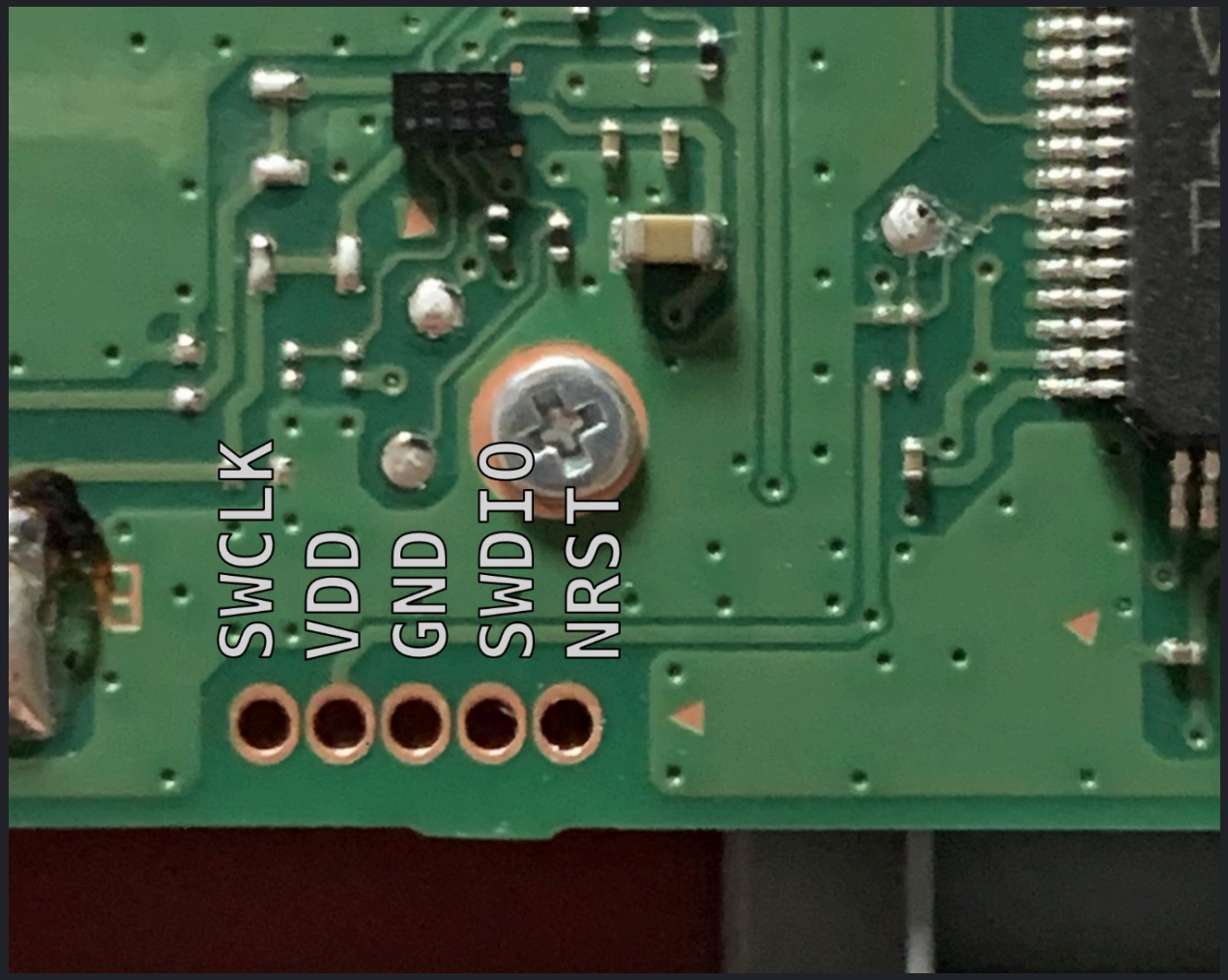
Interface JTAG



Interface SWD

- **S**ingle **W**ire **D**ebug
- Introduit par **ARM**;
- **Remplacement de *JTAG*** qui utilise moins de signaux (2 au lieu de 5);
- Fournit les **mêmes fonctionnalités** que JTAG ~~sans la possibilité de chaînage~~.

Interface SWD



source: @ghidraninja / Twitter

Interfaces propriétaires

- **CC Debug** pour des puces spécifiques *Texas Instruments*
- **PIC ICSP** (utilisée par les micro-contrôleurs Microchip notamment)
- ATMEL **AVR ISP** (In-Situ Programming)
- etc.

SEGGER JLink / JLink EDU

Supporte JTAG & SWD, ainsi qu'un grand nombre de puces, mais coûte cher (€€€).

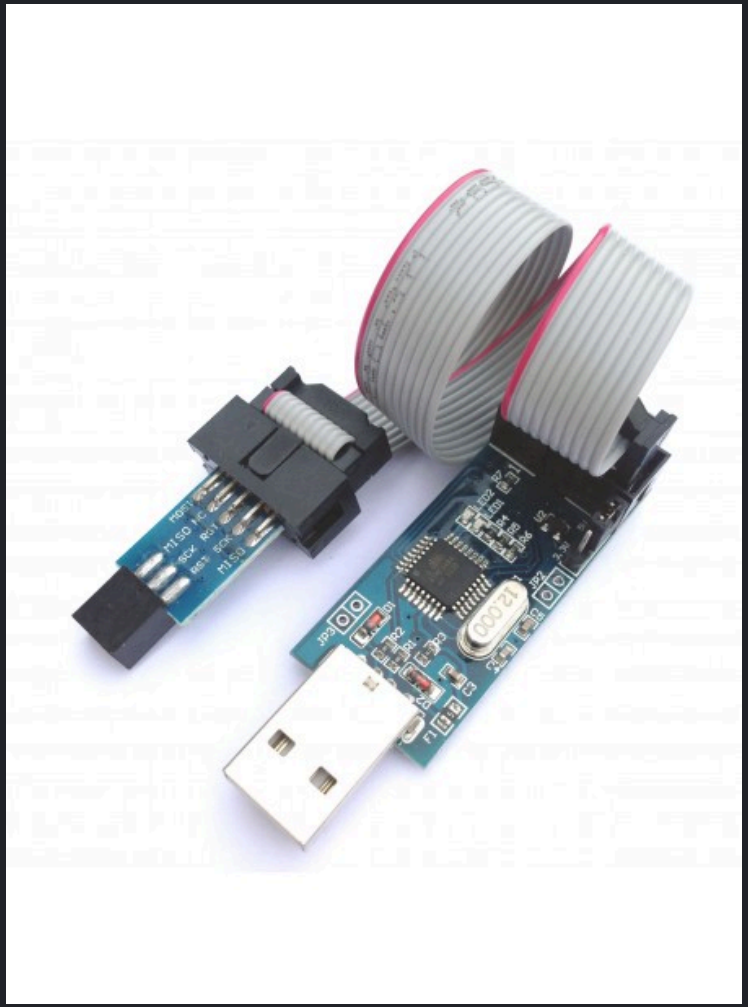
ST-Link v2 / v2.1

Supporte seulement SWD, peu onéreux (gauche) à un peu cher (droite).

Texas Instruments CC Debugger

PICKit 4

AVR USBASP



OpenOCD

- Open On-Chip Debugging
- Supporte la plupart des adaptateurs standards;
- Implémente la majorité des fonctionnalités de débogage et d'accès mémoire;
- S'interface avec GDB.

OpenOCD

On développe un script de configuration (*atsamd21g15.cfg*) pour l'interfaçage avec l'ATSAMD21G15:

```
source [find interface/stlink-v2.cfg]
set CHIPNAME at91samd21g15
set CPUTAPID 0x0bc11477
source [find target/at91samdXX.cfg]
```

OpenOCD

```
$ openocd -f atsamd21g15.cfg
$ nc localhost 4444
```

```
✘ virtualabs@virtubox  ~/qb/training  sudo openocd -f atsamd21.cfg
Open On-Chip Debugger 0.11.0-dirty (2022-01-06-14:38)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
WARNING: interface/stlink-v2.cfg is deprecated, please switch to interface/stlin
k.cfg
Info : auto-selecting first available session transport "hla_swd". To override u
se 'transport select <transport>'.
Info : The selected transport took over low-level target control. The results mi
ght differ compared to plain JTAG/SWD
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Info : clock speed 400 kHz
Info : STLINK V2J39S7 (API v2) VID:PID 0483:3748
Info : Target voltage: 2.835602
Info : at91samd21g15.cpu: hardware has 4 breakpoints, 2 watchpoints
Info : starting gdb server for at91samd21g15.cpu on 3333
Info : Listening on port 3333 for gdb connections
```

```
virtualabs@virtubox  ~  nc 127.0.0.1 4444
ÿûÿûÿýÿþOpen On-Chip Debugger
> reset halt
reset halt
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x000019b0 msp: 0x20000878
>
```

Extraction de firmware

15.4 Image loading commands

Command: **dump_image** *filename address size*

Dump *size* bytes of target memory starting at *address* to the binary file named *filename*.

TP

Extraire le contenu de la Flash du micro-contrôleur avec OpenOCD.

Thank you!

Contact information:

- Email: sbabigeon@quarkslab.com
- Website: https://blog.quarkslab.com