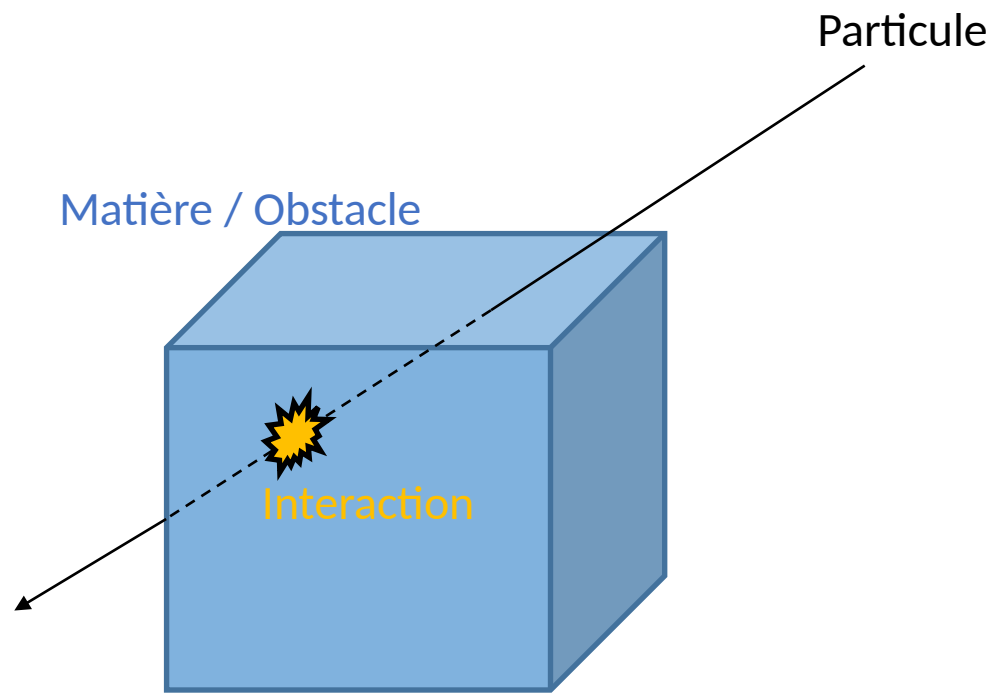


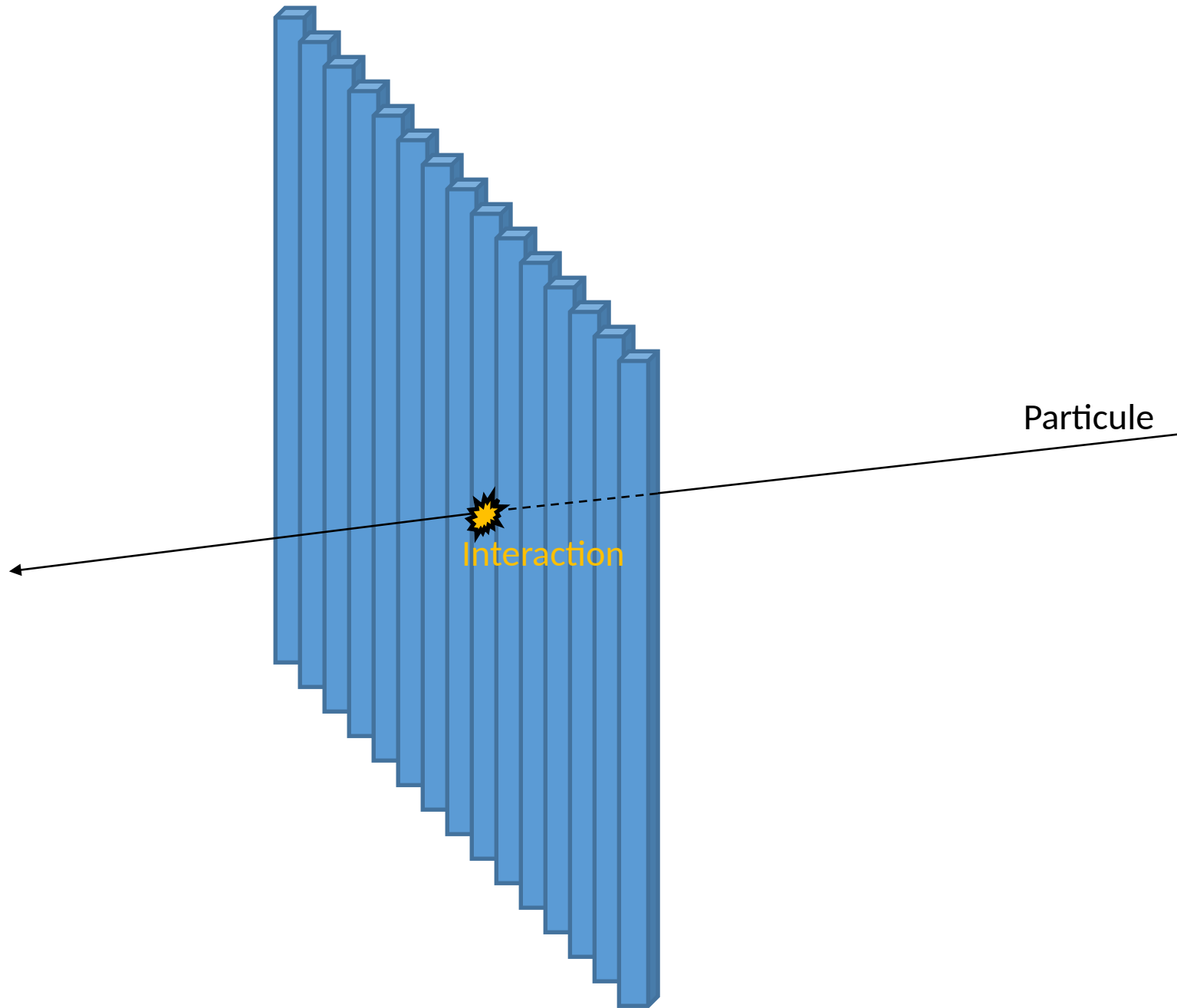
Techniques et Ingénierie des détecteurs

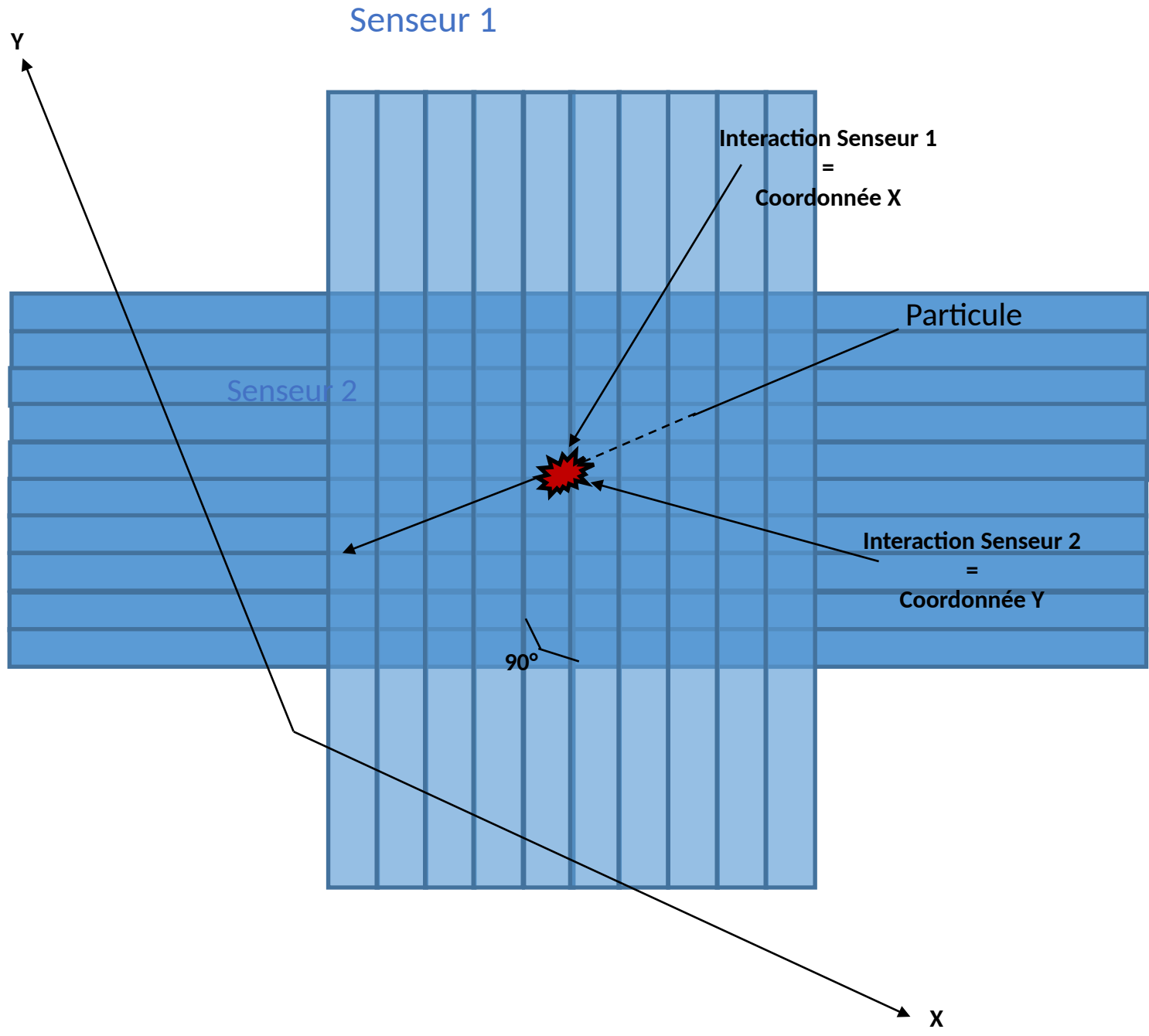
Christian Bonnin - Laurent Gross - Laurent Charles

MasterClasses

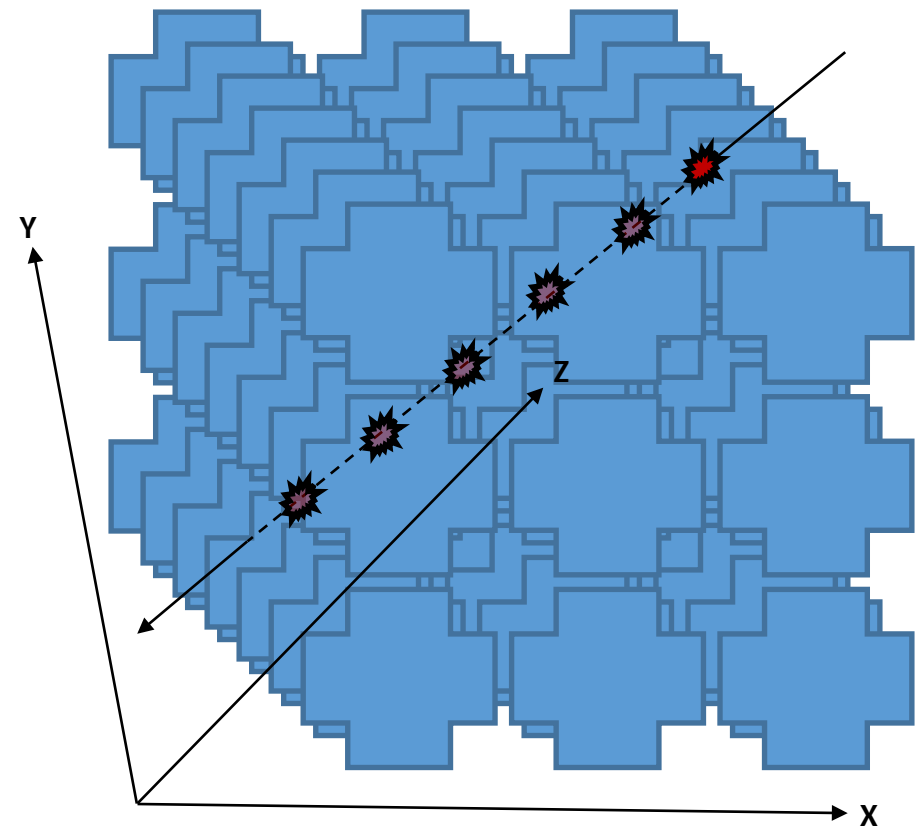
Bonjour à tous



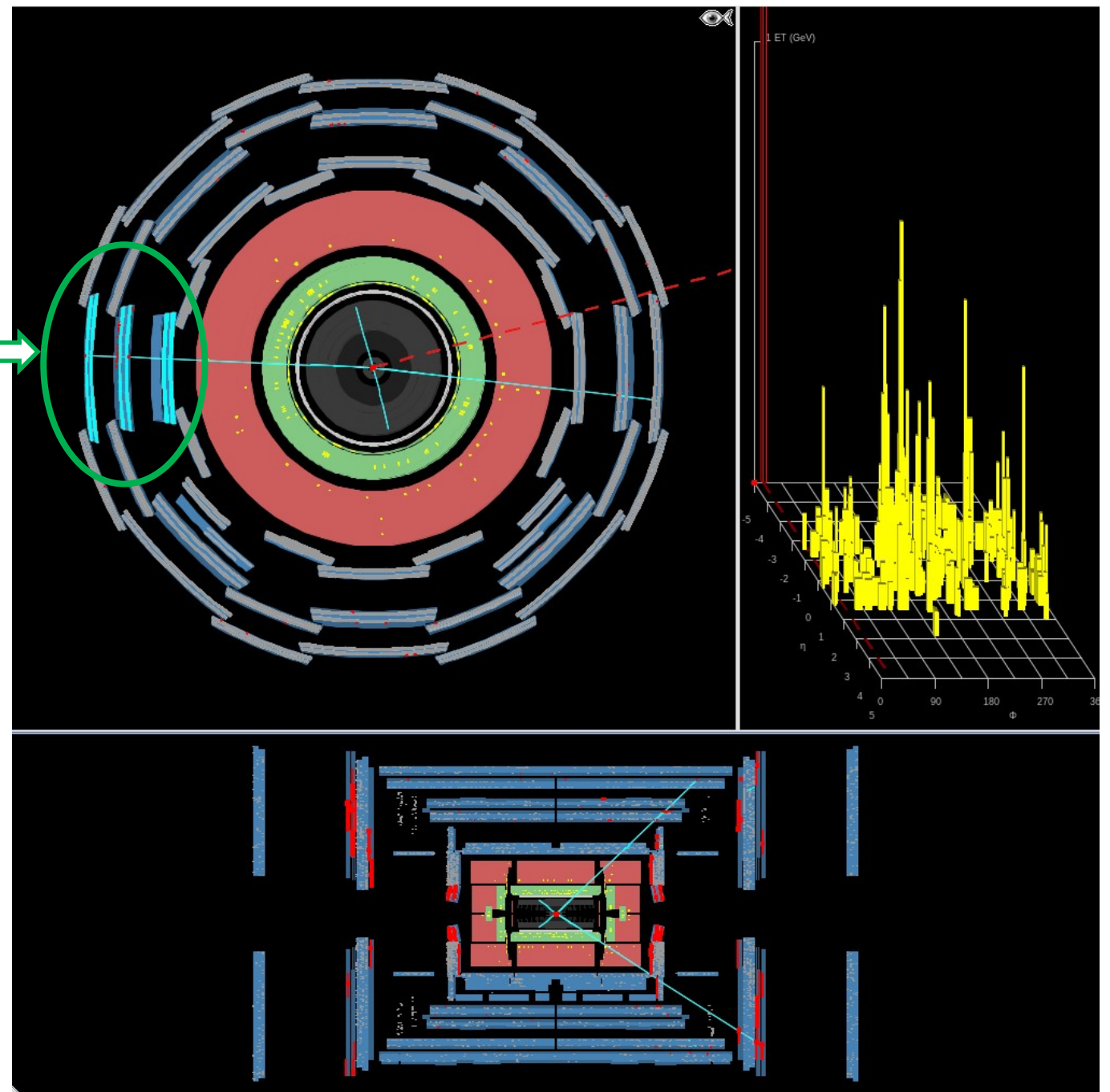
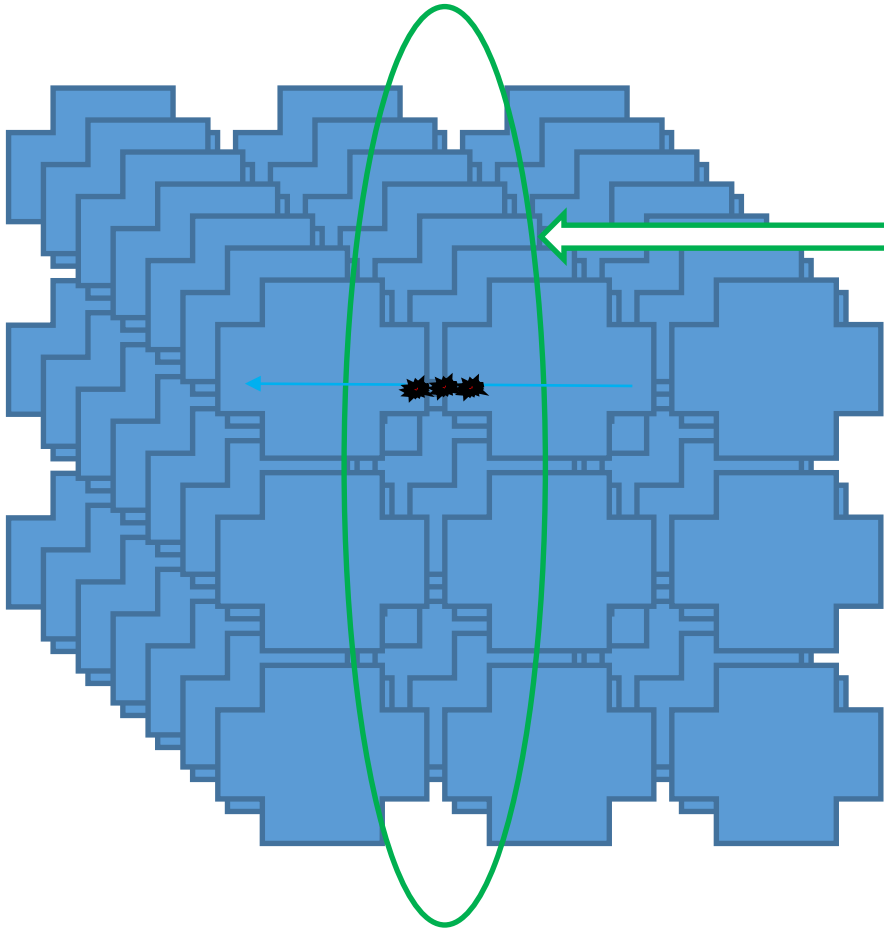




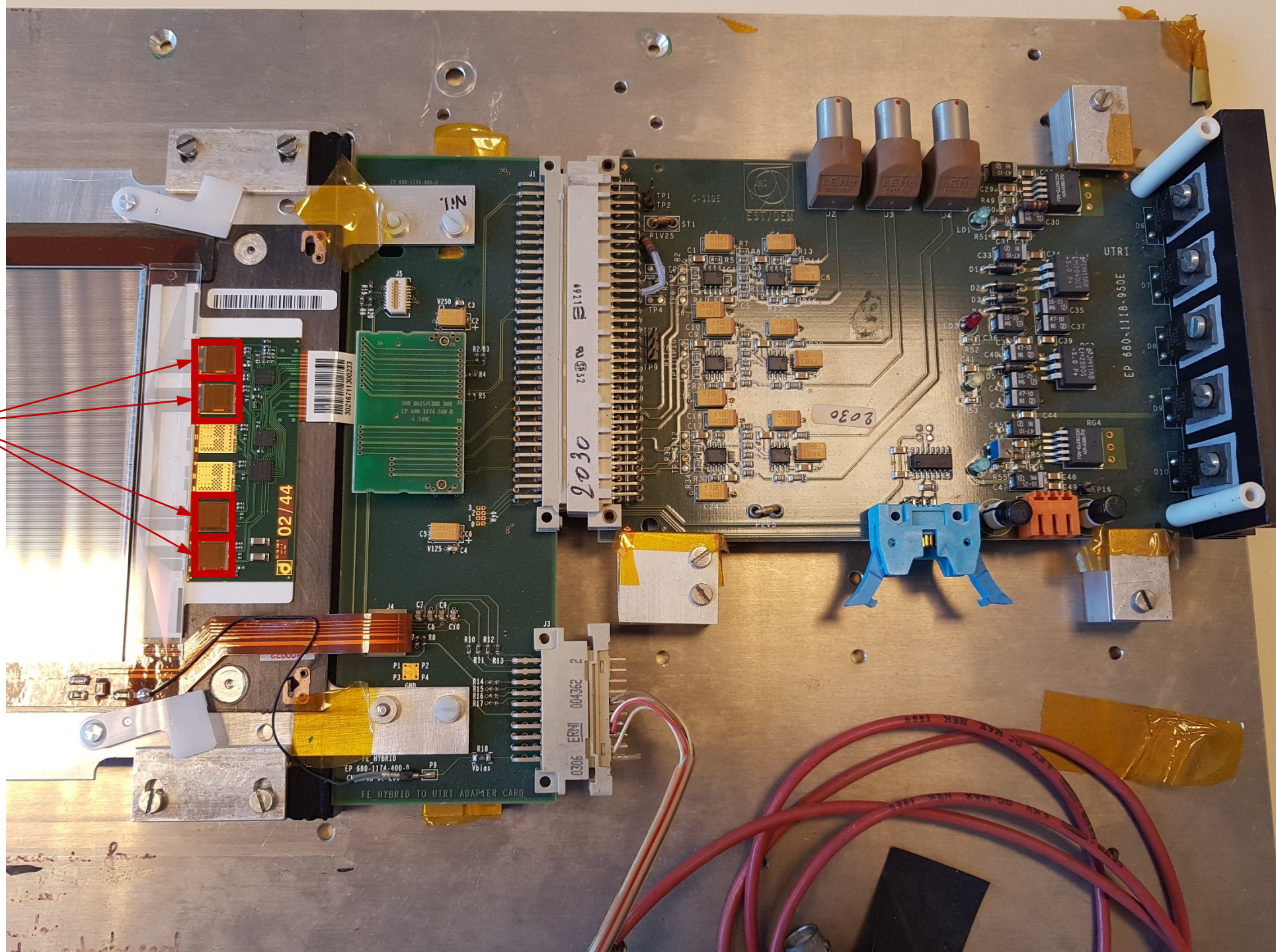
Volume de détection = Détecteur

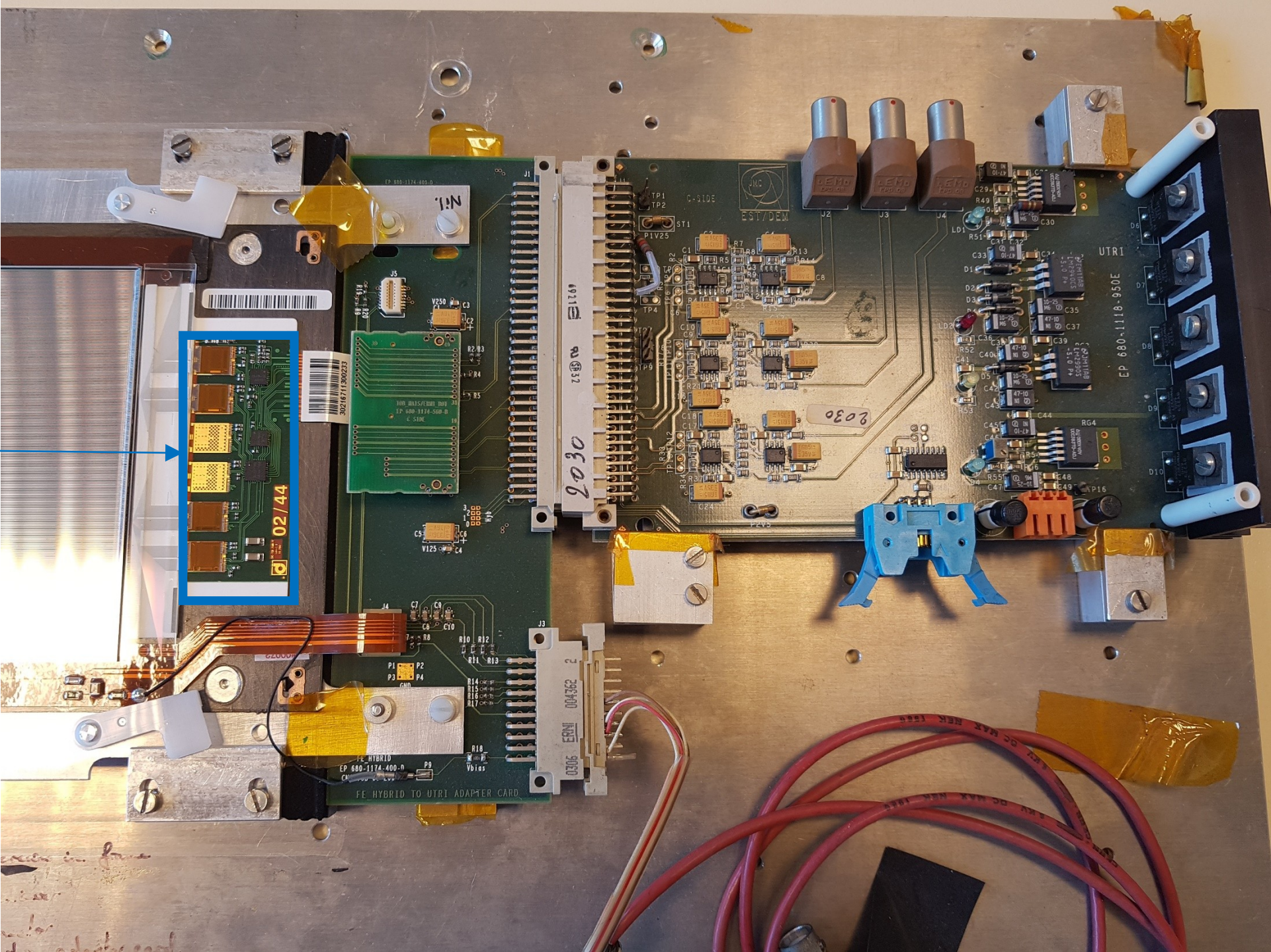


Volume de détection
- vue par la tranche -



Chips de Lecture





Hybride

02/44

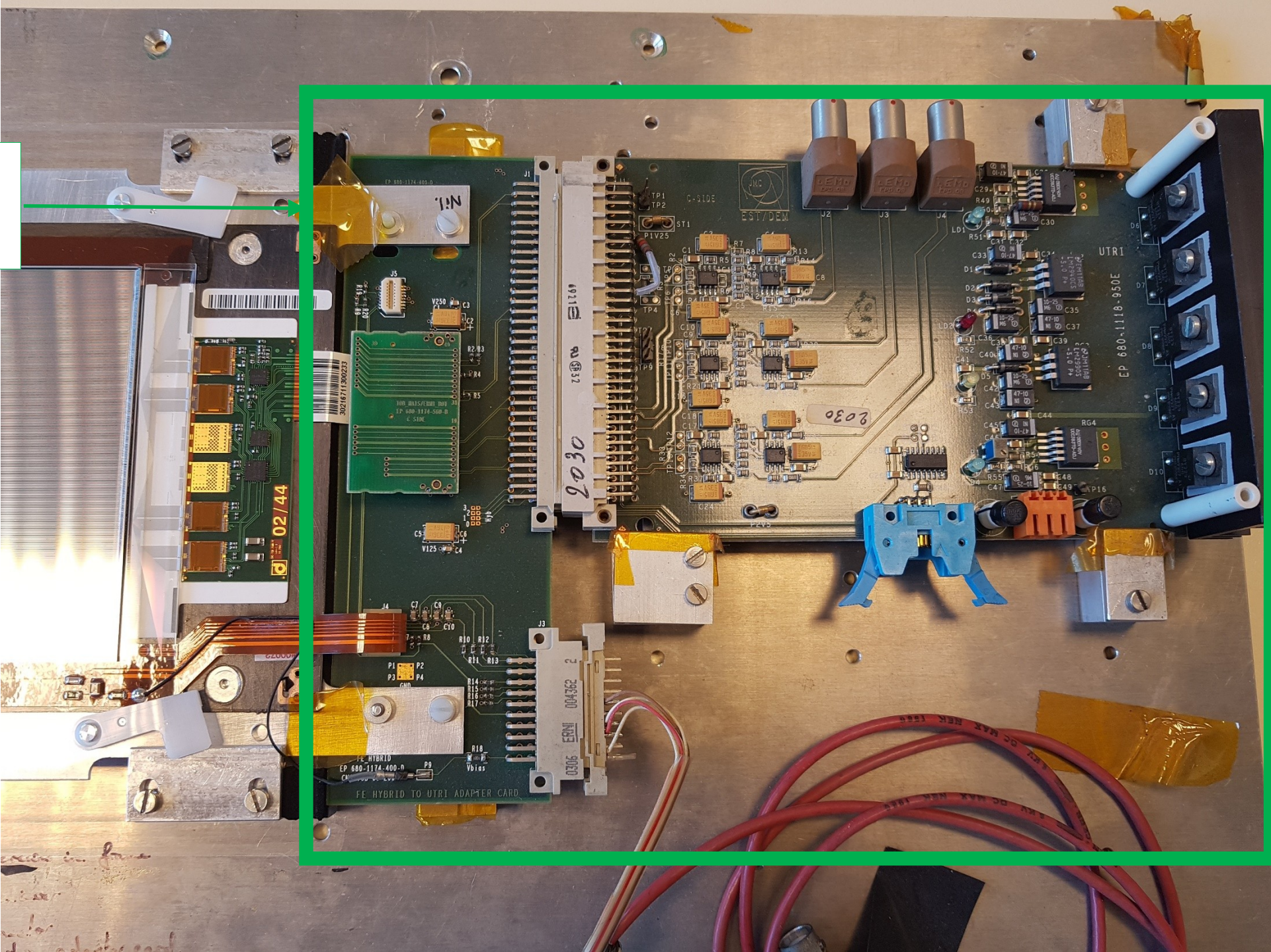
2030

UTRI

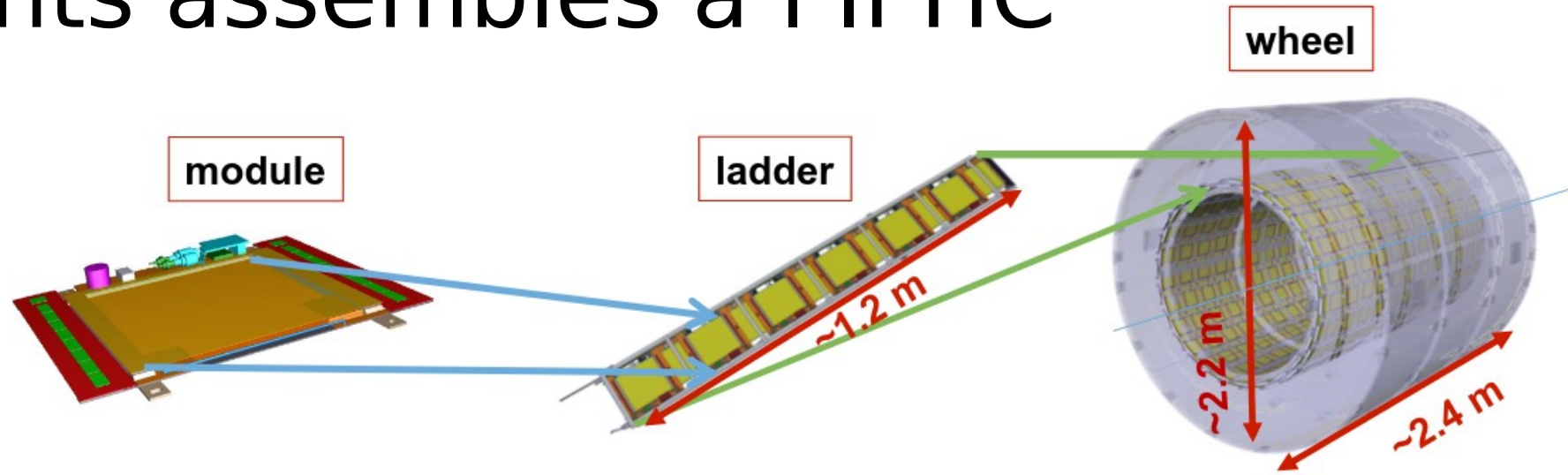
EP 680-1118-950E

FE HYBRID TO UTRI ADAPTER CARD

Entrées/Sorties
- Configuration
- Données



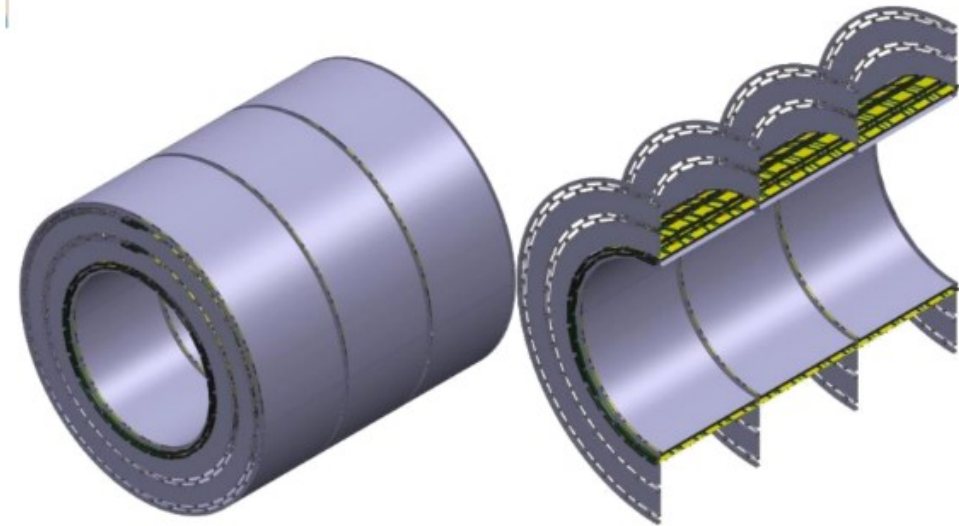
Éléments assemblés à l'IPHC



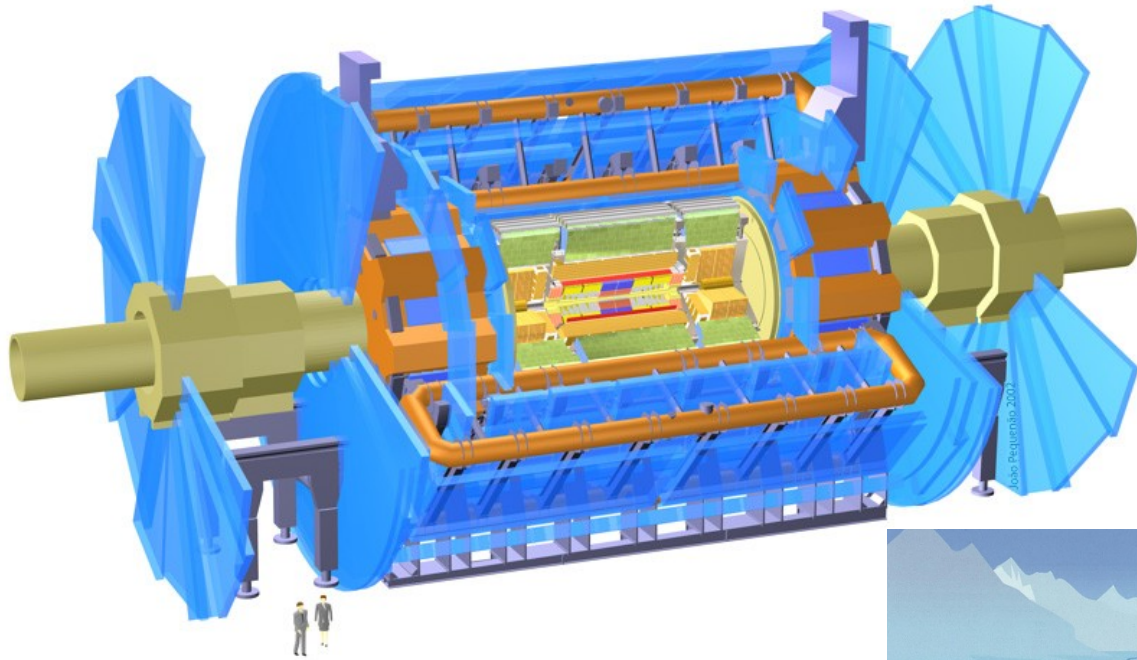
TOB



TB2S



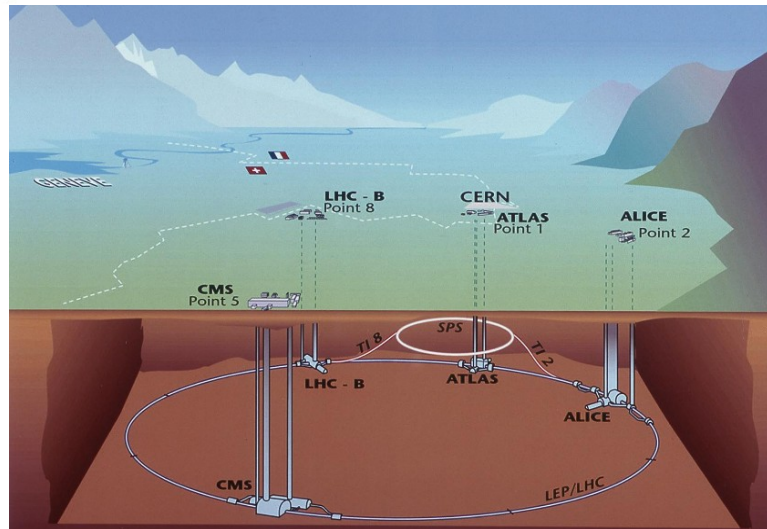
Grandes expériences du CERN



ATLAS

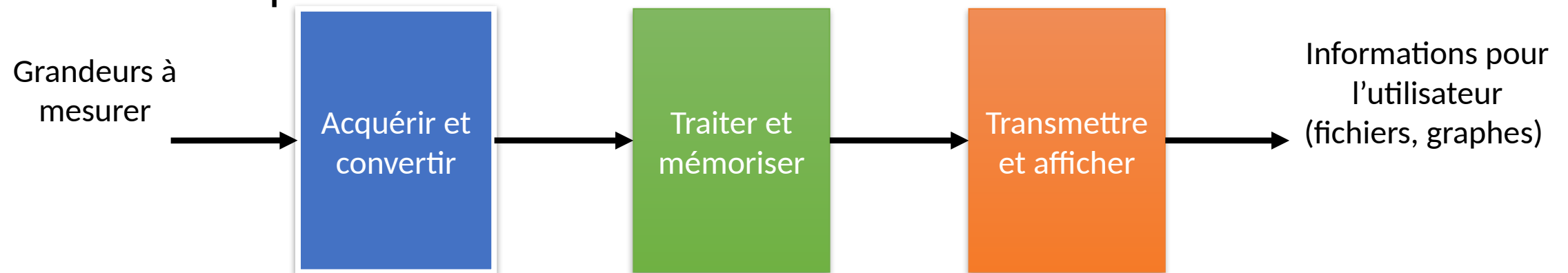


CMS

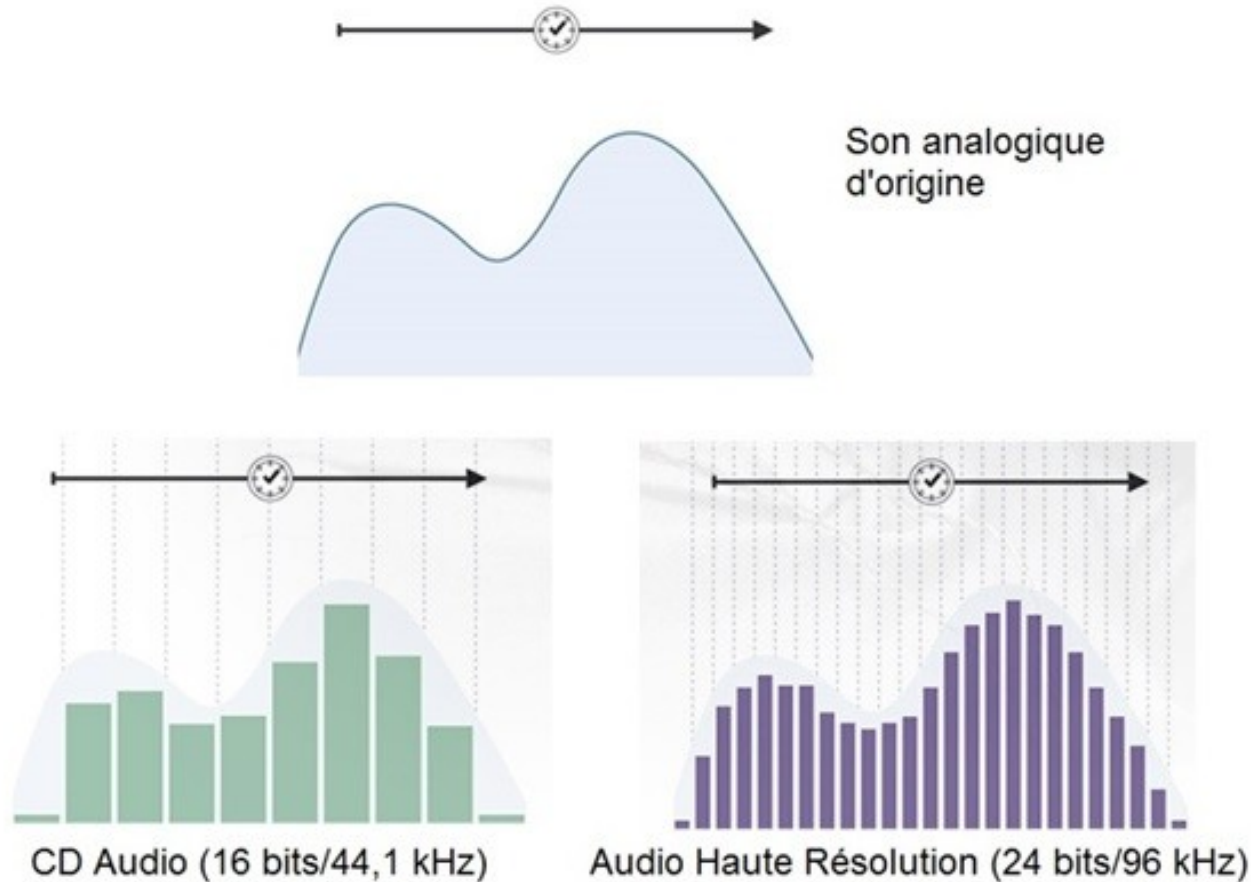


C'est quoi une DAQ ?

- Système ou chaîne d'acquisition de données
- Ensemble des éléments qui permet de mesurer, exploiter et analyser une grandeur ou un phénomène physique
 - Capture des signaux, conversion en valeur numérique, traitement, transmission, stockage (pour les utiliser et les traiter ultérieurement) affichage
- Système hétérogène qui utilise les technologies électroniques et informatiques



Conversion en signal numérique - Analogique vs Numérique

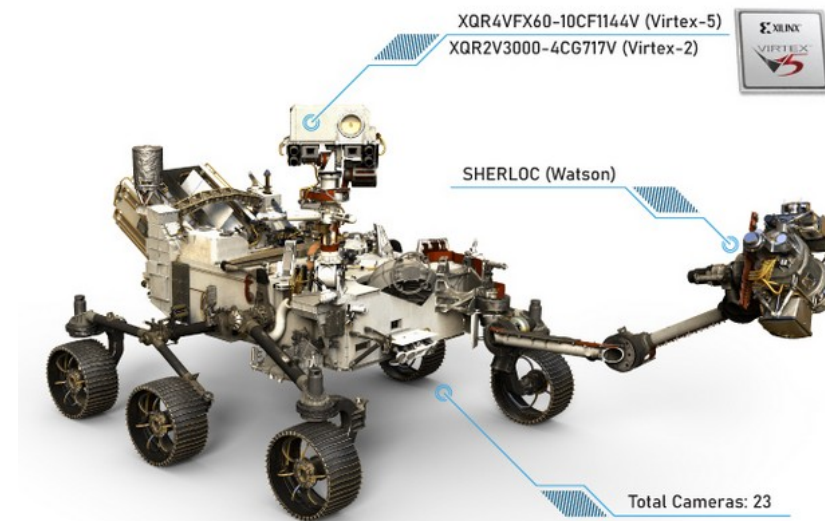


Numérique

- ❑ Signal analogique converti en numérique grâce à un convertisseur (échantillonneur et codeur)
- ❑ Amplitude des échantillons est codée en valeur binaire (suite de « 0 » et de « 1 ») compatible avec l'informatique moderne

Où trouve t-on des DAQ ? – Les applications

- Partout où l'on a des capteurs mesurant des grandeurs physiques (T°C, pression, luminosité, son, énergie d'une particule de physique, etc.)
 - Transports
 - Avionique / aéronautique (calculateurs)
 - Automobile (aides à la conduite, caméra de recul, systèmes de sécurité)
 - Etc.
 - Robotique (caméra, Lidar, etc.)
 - Spatial (rover sur Mars)
 - Expériences de physique
 - Etc.

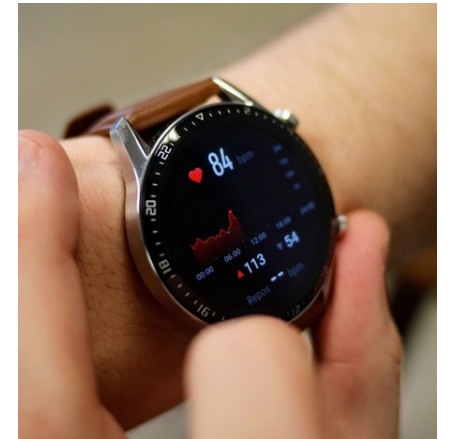


D'une DAQ à une autre

- Varie selon
 - Complexité du système (nombre de voies de mesures, cadence de mesure, bande passante de transfert des données, etc.)
 - Contraintes (taille, espace, conditions environnementales, consommation en énergie)
 - Technologies
 - Expériences LHC utilisent les dernières technologies et standards industriels
 - Coût
- Ex montre connectée mesurant plein de données (rythme cardiaque, température, phases sommeil, etc.)
 - Si seule au poignet
 - système 'simple' et autonome ('simple enregistreur')
 - Si couplée aux applications en ligne via son smartphone
 - DAQ étendue utilisant des équipements informatiques distants (envoi des données dans une base de données, logiciels d'analyse, affichage de graphes de monitoring)

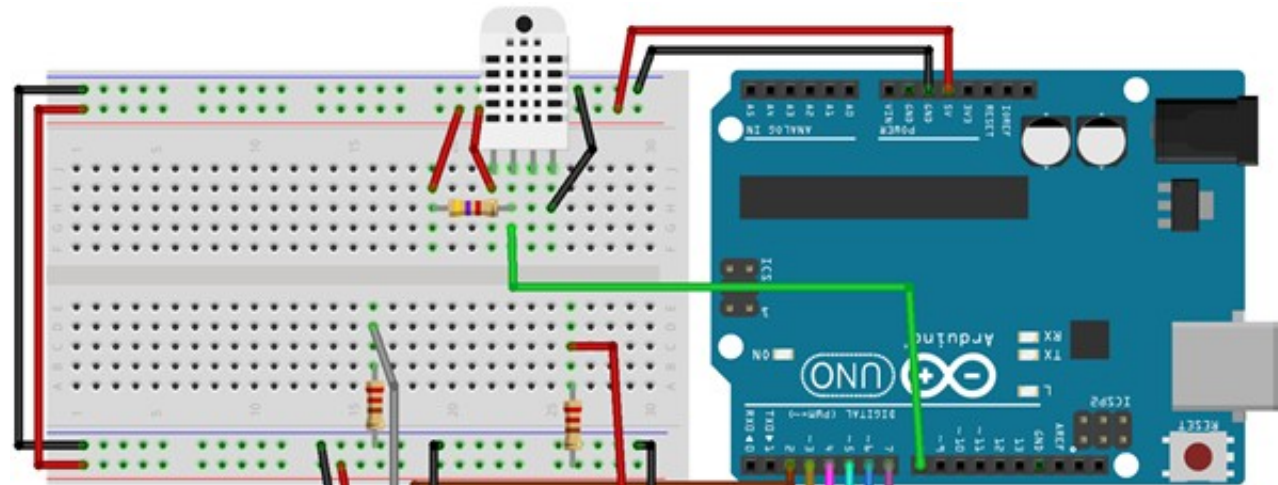
Couche Pixels Atlas

- ❑ 92 millions de voies
- ❑ Cadence : 40 MHz



Exemple de DAQ simple : mesure de T°C

Sonde de T°C
(analogique)
montée sur
plaquette d'essai



Carte
Electronique
Arduino avec
microprocesseur
« partie
intelligente »

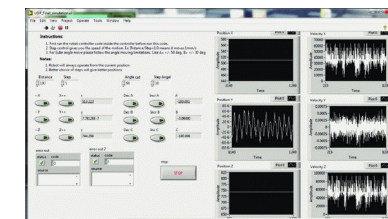
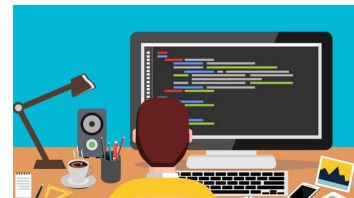
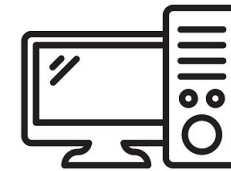
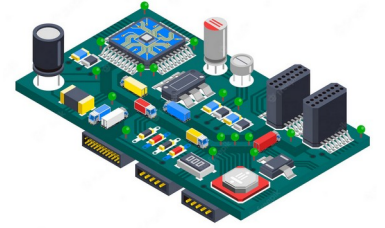


Afficheur où sera affichée
la valeur de la T°C
partie « Interface
Homme Machine » (IHM)



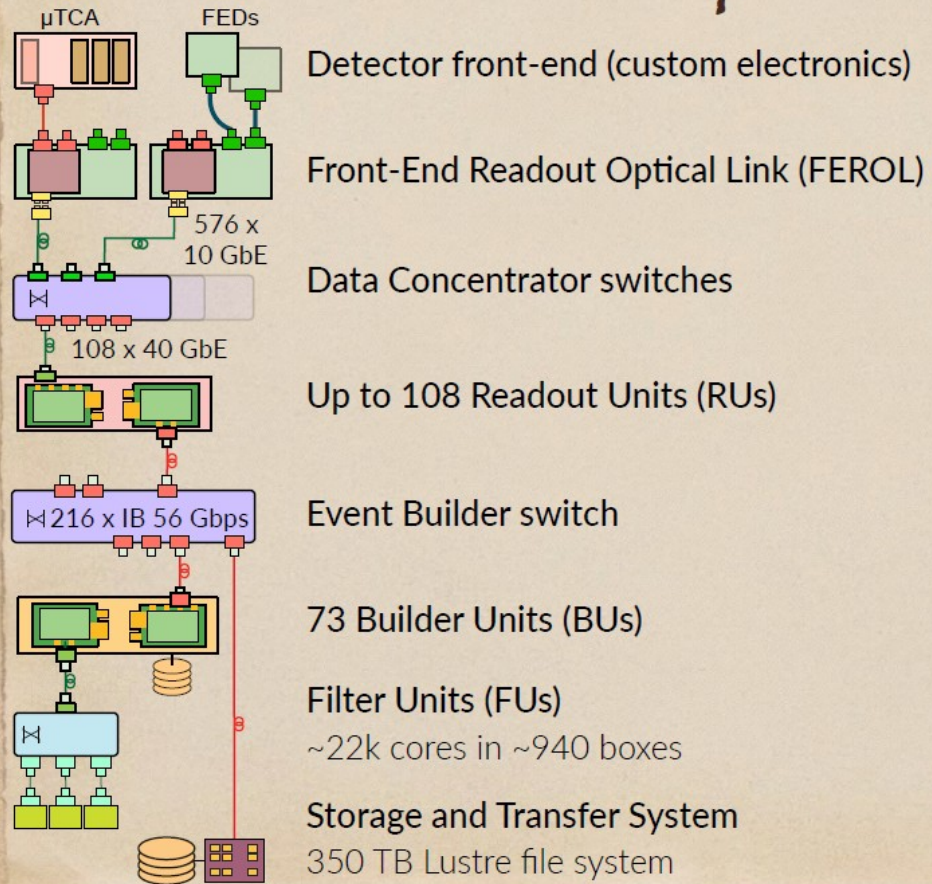
Éléments utiles pour concevoir une DAQ

- Cartes électroniques
 - Composants électroniques
 - Capteurs (circuits intégrés)
 - Circuits « intelligents » pour du calcul (microprocesseurs ou CPU, GPU, FPGA)
 - Mémoires
- Interfaces entrées/sorties avec liaisons de communication pour relier à des périphériques (Ethernet, USB, etc.)
 - Câbles de connexion (câbles électriques, fibres optiques)
- Mécanique (boîtier, châssis)
- Matériel / infrastructure informatique (si nécessaire)
 - PC, routeurs Ethernet, serveurs, baies ou armoires, ferme de calcul
- Programmes
 - Codes embarqués (firmware)
 - Logiciels sur PC (software)
 - De traitement et d'analyse, de contrôle, de diagnostic, d'affichage

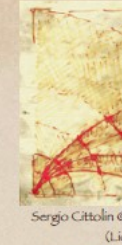


DAQ - expériences LHC CERN

CMS Data Acquisition System



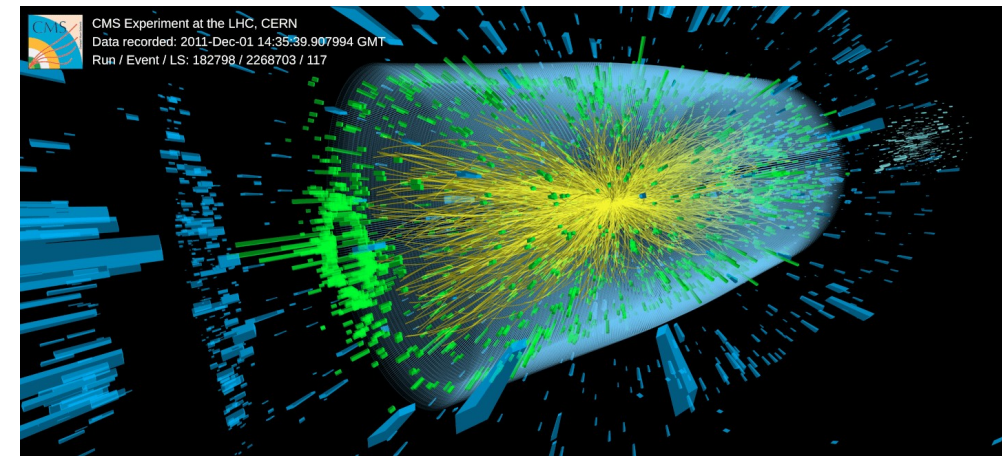
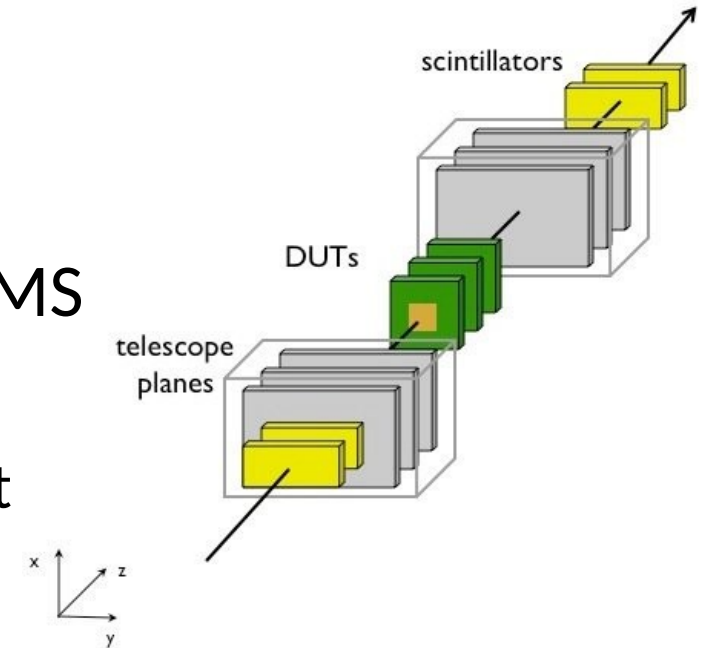
- ~700 front-end drivers (FEDs)
- 0.1 - 8 kB fragments at 100 kHz (1.2 MB event size)
- Custom protocol from FEDs
- Optical 10 GbE TCP/IP
- Data to Surface over ~200m
- Aggregate into 40 GbE links
- Combine FEROL fragments into super-fragment
- Buffer fragments
- Infiniband FDR 56 Gbps CLOS network
- Event building & temporary recording to RAM disk
- Run HLT selection using files from RAM disk
- Select O(1%) of the events for permanent storage
- Merge output files from filter unit
- Transfer files to tier 0 or online consumers at pt.5



- Très complexe
- Nombreuses technologies de pointe
- Vitesse de transfert énorme
- Dans l'équipe : essentiellement étage 1 de la chaîne

Equipe Technique CMS

- DAQ de lecture des modules senseurs du tracker de CMS
- Mise en œuvre de DAQ prototypes (bancs d'essai)
 - Qualifier et tester les modules en labo ou sous faisceau test
- Mise en œuvre de DAQ de production
 - Prise de données de physique
- Intégration électronique
 - Assemble, monte, connecte
- Conception de codes embarqués et logiciels
- Support technique aux physiciens

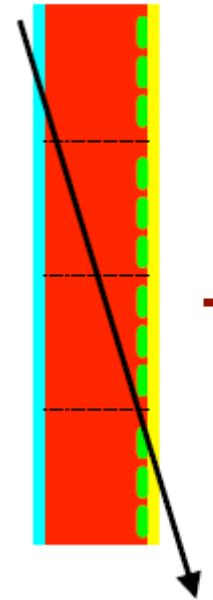


Numérisation

Analogique

Numérique

Sensor



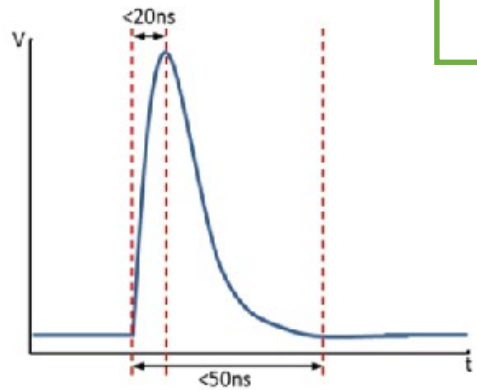
I_{in}

X Nombre de voies (micro-pistes)
du senseur

Collection
des charges
et mise en
forme du
signal

Paramètres

CBC pulse shape



Comparaison

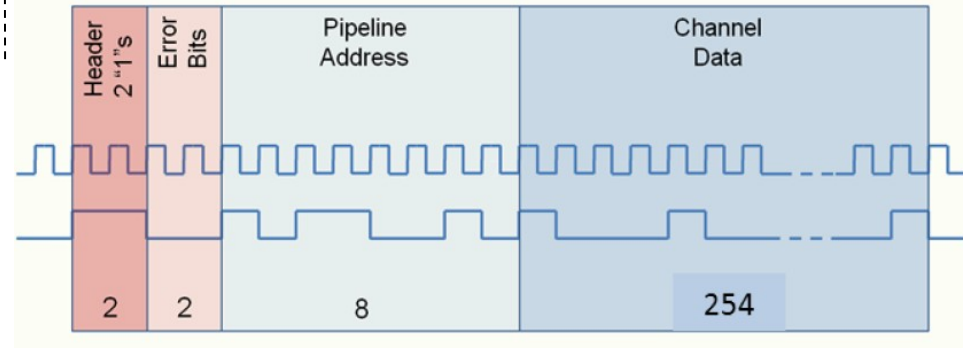
Seuil de
détection

Hit



Trigger (bon
évènement à sortir)

Mise en
mémoire et
transmission
de données
binaires



←→
Senseur 254 voies

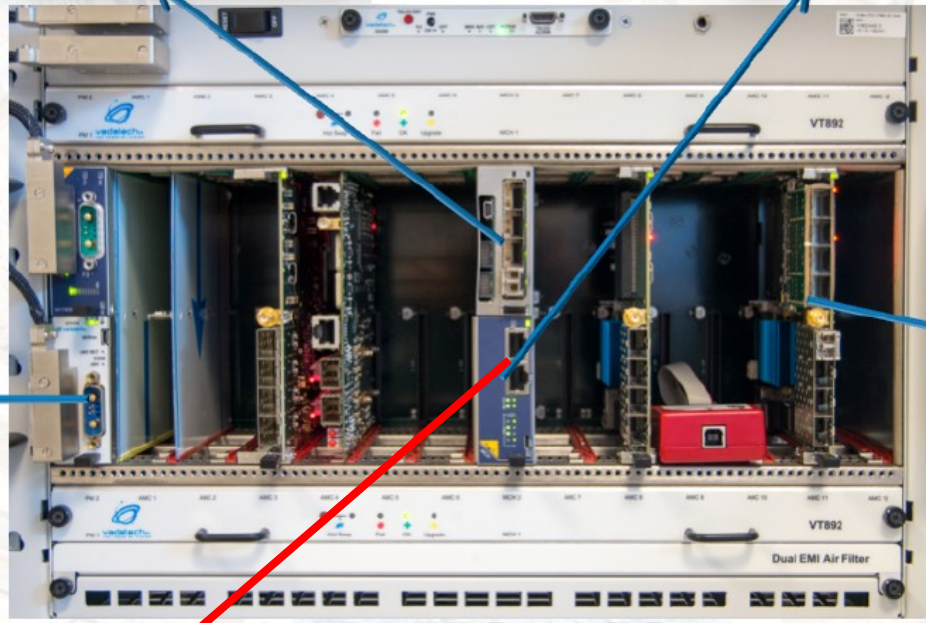
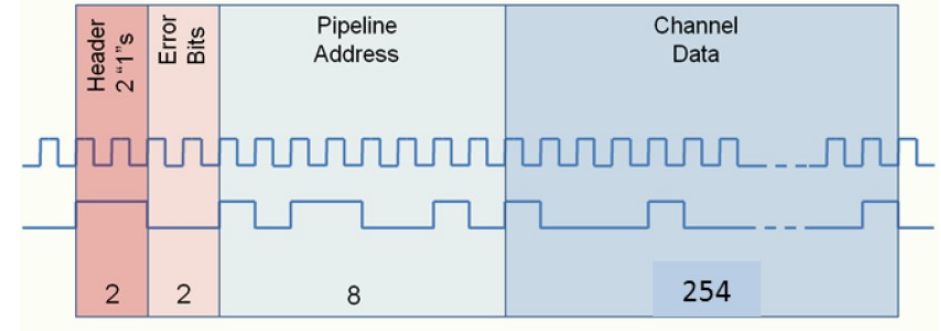
←→
Circuit de lecture (Readout Chip)

Transmission

AMC13



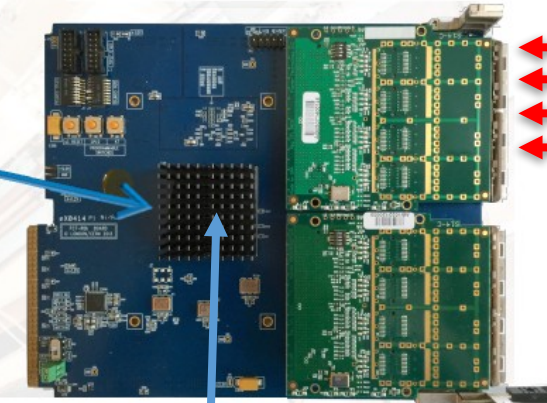
NAT MCH



VADATECH PSU



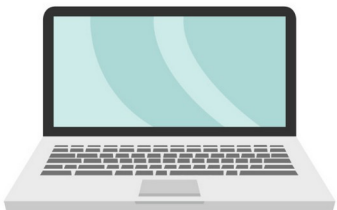
VADATECH μTCA
Crate



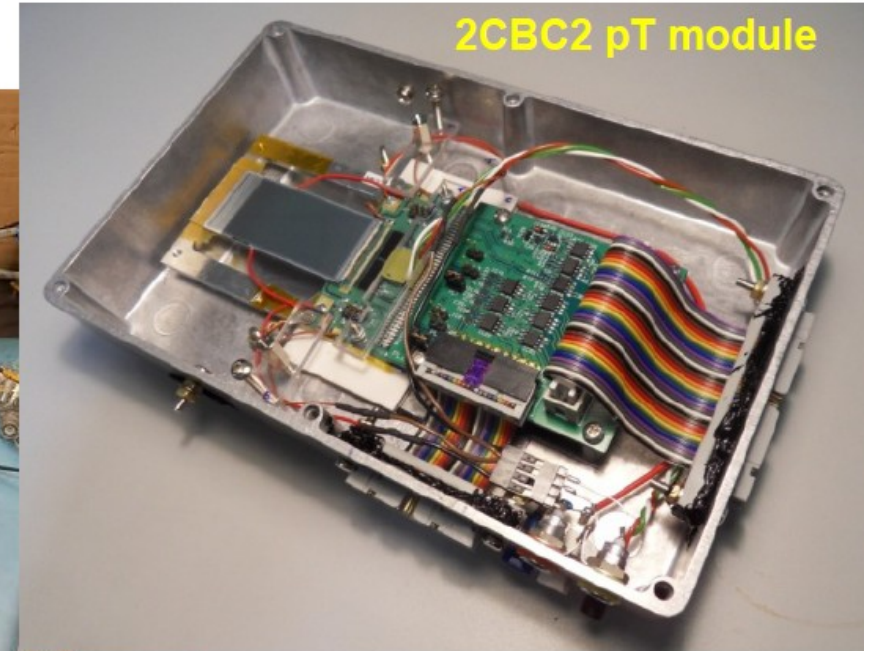
FC7

Carte
d'acquisition

FPGA Circuit de
traitement et
calcul



Premières DAQ de test sur table



DAQ plus intégrée dans la ligne CMS au cyclotron Cyrce

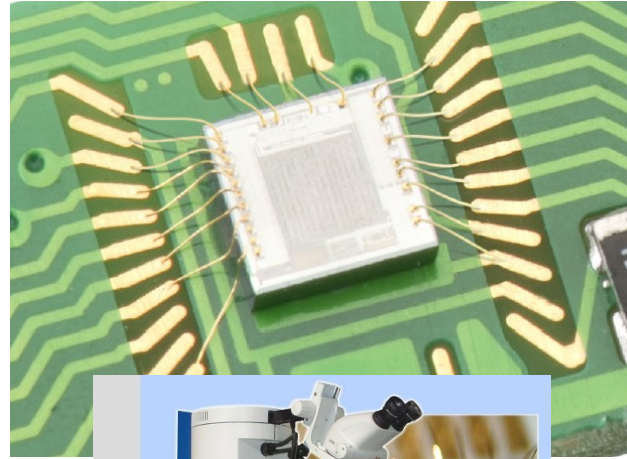


Métiers

Ingénieur microélectronique

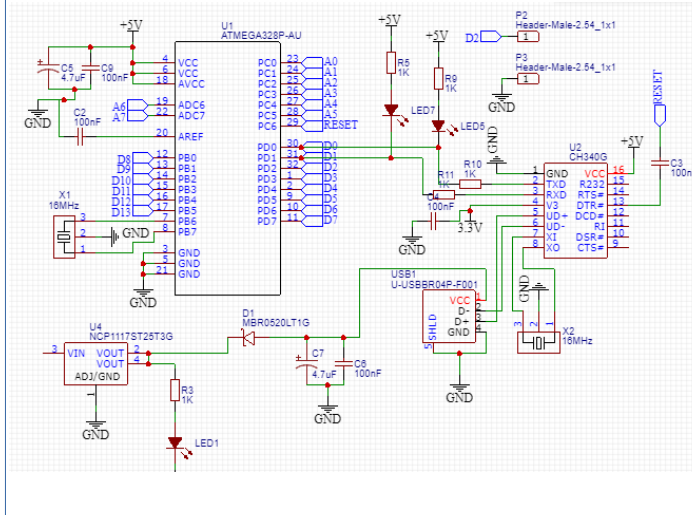


Technicien/Ingénieur microtechnique

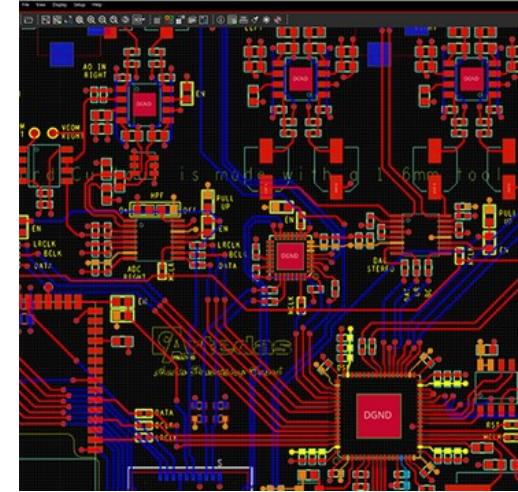


Technicien/Ingénieur électronique

Schéma carte



Routage carte (PCB)



Ingénieur Développeur logiciel

```

file_in = open("in.txt", 'r')
file_out = open("out.txt", 'w')
try:
    print('cannot open', file_name)
except IOError:
    pass
i = 1
for line in file_in:
    print(line.rstrip())
    file_out.write("line " + str(i) + "\n")
    i = i + 1
file_in.close()
file_out.close()
    
```

XDAQ
designed by Gildas A. Briere

FEC Tester

SRO	0x1e90	SR1	0x0	
14	TCR_READY	1	15	RESET_OUT
13	DATA TO FEC	0	8	TIMEOUT
12	PENDING IRQ	0	7	RESET_LINK
11	LINK_INITIALIZED	1	6	RESET_TRCH
10	TRANSMIT FIFO EMPTY	1	5	ERROR
9	TRANSMIT FIFO FULL	0	4	SELECT_SEE
8	RETURN FIFO FULL	0	3	SELECT_SEE
7	RECEIVE FIFO FULL	0	2	SELECT_XTAL_C
6	RECEIVE RUNNING	0	1	SEND
5	RECEIVE RUNNING	0	0	ILLEGAL_SEQUENCE
4	RECEIVE RUNNING	0	0	ILLEGAL_DATA
3	RECEIVE RUNNING	0	0	ILLEGAL_DATA
2	RECEIVE RUNNING	0	0	ILLEGAL_DATA
1	RECEIVE RUNNING	0	0	ILLEGAL_DATA
0	RECEIVE RUNNING	0	0	ILLEGAL_DATA

Hard RESET

Firmware version: 0x1800. TK FEC 2015-3-23 02_07 @ Ring 0

Destination	Source	Length	Data
80	00	04	00
07	10	A0	00

5.4.218-GIC-GM6E2 Ipbus Control (1 card) on the IP bus at 192.168.1.10

Glibbox cards | Oscillograms of card n°SGLB

Detect Ipbus Card IP: 192.168.1.10 Card: [G]L[B]

Add	Name	Description	Value
0x00	Board_ID	The board identifier code	RD 32 30 76 31
0x01	System_ID	The system identifier code	RD 00 00 00 00
0x02	Firmware_ID	The firmware date and version number	RD 00 01 10 12
0x03	TestReg	Register for test purposes only	RW 00 00 00 00
0x04	Ctrl	Controls the external clocking circuitry	RW 10 AA 20 71
0x05	Ctrl	Phase control	RW 00 00 00 02
0x06	Status	Status from various external components	RD 00 00 00 00
0x07	Status	Currently not used	RD 00 00 00 00
0x08	Ctrl_SRAM	SRAM interface control	RW 00 01 00 01
0x09	Status_SRAM	SRAM interface status	RW 00 00 00 00
0x0A	Ctrl_SIP	SIP interface: data from FPGA to clock synthesizer	RW 00 00 00 00
0x0B	Status_SIP	SIP interface: data from clock synthesizer to FPGA	RW 00 00 00 00
0x0C	Ctrl_Protocol	IP interface: configuration (priority, phase, frequency, etc.)	RW 00 00 00 00
0x0D	Status_Protocol	IP interface: configuration (bus select, frequency, etc.)	RW 00 00 00 00
0x0E	Ctrl_IC	IC interface: transaction parameters (slave address, data to slave, etc.)	RW 00 00 00 00
0x0F	Status_IC	IC interface: transaction parameters (slave address, data to slave, etc.)	RW 00 00 00 00
0x10	Ctrl_IC_reply	IC interface: transaction reply (transaction status, data from slave etc.)	RW 00 00 00 00

Load parameters... Save parameters... Ask card Update card Update all cards

Management of acquisition data: Save to file

Acquisition Selected card, on IP bus: [G]L[B] [S]G

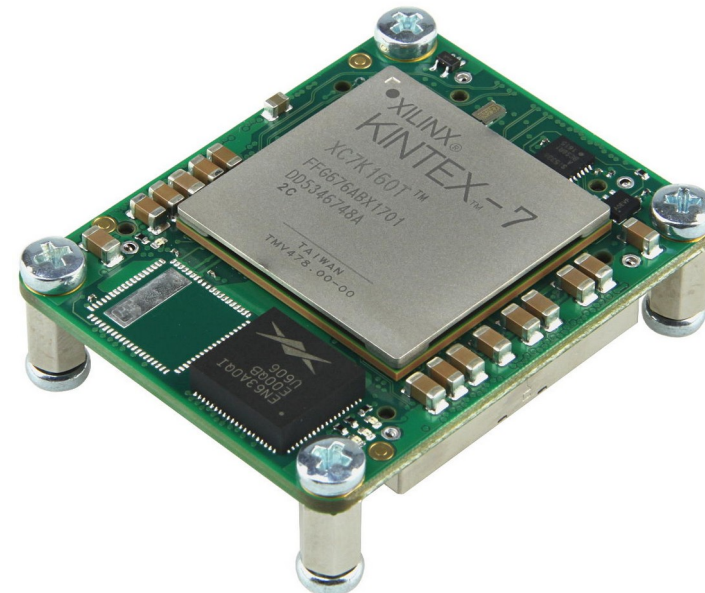
Start all cards

Total bytes received from all cards: 78 154 888

Backup dir: C:\Glibbox\DATA\2014_11_17_Lund\Run1

FPGA - Composant principal des cartes d'acquisition (CMS et CERN en général)

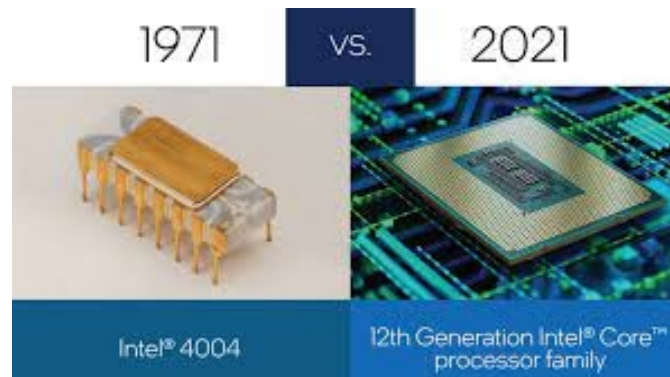
- Aujourd'hui la plupart des systèmes d'acquisition des expériences au CERN utilise des composants électroniques FPGA dans les parties demandant vitesse et parallélisme de calcul, liaisons rapides, grande bande passante, nombre important d'entrées/sorties



FPGA c'est quoi?



- C'est un circuit intégré logique ou réseau logique reprogrammable
 - Circuit intégré est une puce électronique gravée sur silicium (semi-conducteur). Les ressources internes sont miniaturisées et regroupées dans un unique boîtier à souder sur une carte électronique
 - Reprogrammable : on peut changer sa fonction/application à volonté
- Il a été inventé en 1984
 - Le premier microprocesseur (Intel 4004), base de l'informatique moderne, date de 1971



FPGA vs CPU

	CPU	FPGA
Hardware	Figé	Flexible (personnalisable par l'utilisateur)
Architecture	Haut-niveau (programmes exécutés suivant une archi et un jeu d'instructions défini)	Bas-niveau (niveau portes, éléments mémoires, blocs multiplieurs, additionneurs)
Type d'exécution	Séquentiel	Parallèle
Temps d'exécution		Très rapide
Consommation énergétique		Moins bonne
Temps de développement		Long car syntaxe et outils complexes
Langage de programmation haut-niveau	Oui	Non (mais cela arrive)

Quelques applications FPGA

FPGA

Traitement d'image et video

IA et réseaux neuronaux (inference)

Moteurs de recherche (Google, Bing) dans des Data Centers

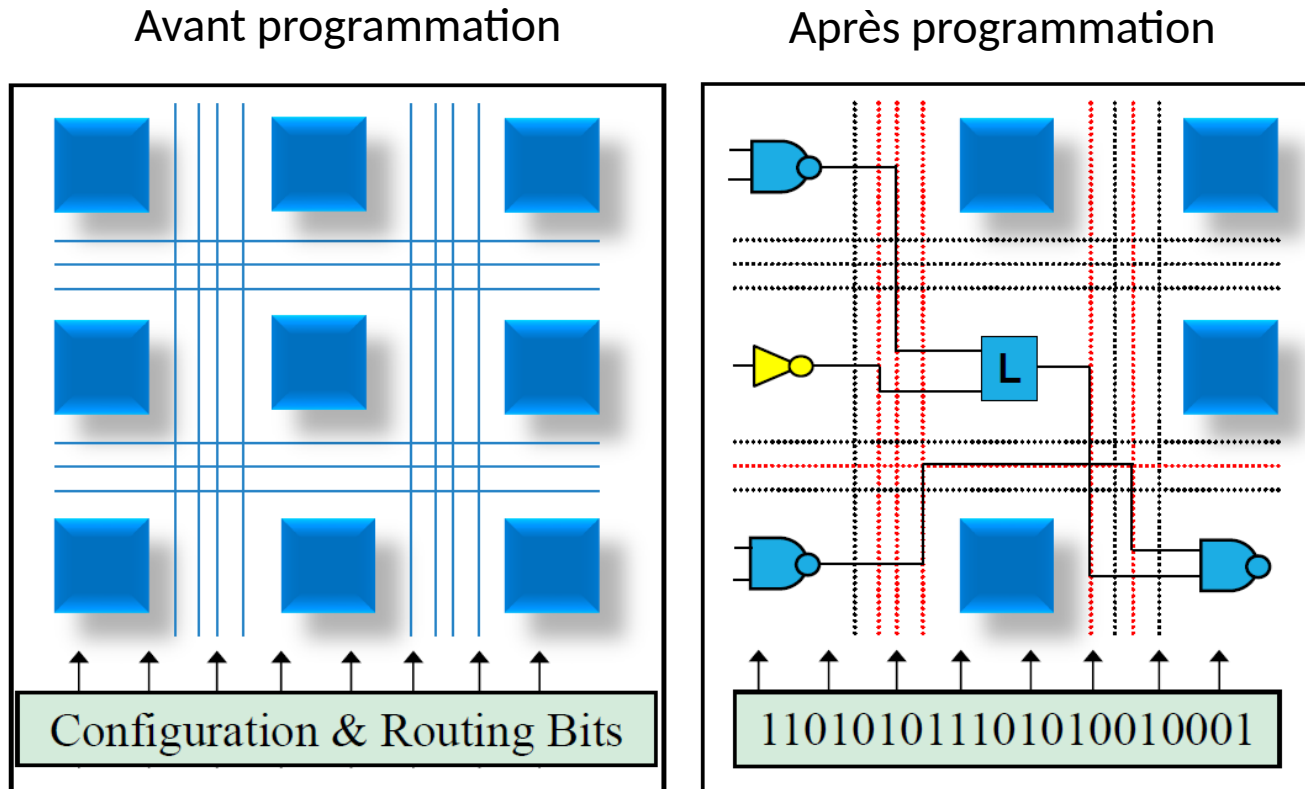
Calcul haute performance (HPC ; fermes de calcul)

Trading haute vitesse

Cartes d'acquisition de données (CERN LHC)

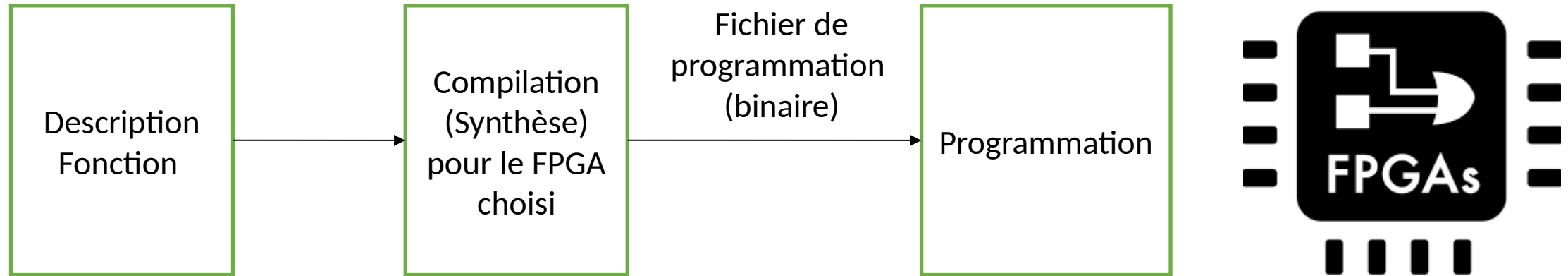
L'intérieur d'un FPGA (simplifié)

- On peut réaliser et router n'importe quelle(s) fonction(s) logique(s) par programmation du circuit



- ❑ Conception / programmation du schéma logique
- ❑ La notion de programmation des FPGA revient à définir une table
 - de configuration/définition des blocs logiques (sa fonctionnalité logique)
 - d'interconnexion des blocs logiques entre eux
 - de configuration des entrées/sorties avec l'extérieur (entrées capteurs, liens USB, Ethernet, etc.)

Flot de conception FPGA (simplifié)



- La programmation des FPGA passe par un compilateur spécifique basé sur un langage de programmation de type langage de description matériel
- Après compilation de cette description, on obtient un fichier de configuration pour le FPGA choisi
- Une fois programmé, le FPGA exécute ses tâches