

# Systeme d'acquisition une voie pour la R&T R2D2

Décteur pour la détermination  
de la nature de la masse  
des neutrinos

Responsable Technique: Patrick Hellmuth

Développement: Patrick Hellmuth

Raphael Bouet

Frédéric Druillole



- Présentation de l'instrument
- Description du système
- Firmware
- Logiciel embarqué
- Contrôle et Commande
- Premier resultat
- Reste à Faire

# Etude de détecteurs gazeux haute pression

**Objectif:** Etude de TPC à gaz Xenon haute pression pour la recherche de la désintégration double bêta ( $\beta\beta_{0\nu}$ ) sans bruit de fond.

→ Le signal extrait du détecteur est riche en information et peut être exploité pour l'identification de particules.

## 3 TPCs construites

→ Resolution en energie : 1% fwhm à 1,5 MeV



Proto 1: TPC Sphérique (creusée dans un cylindre) non certifiée



Proto 2: TPC Sphérique certifiée 40 bars



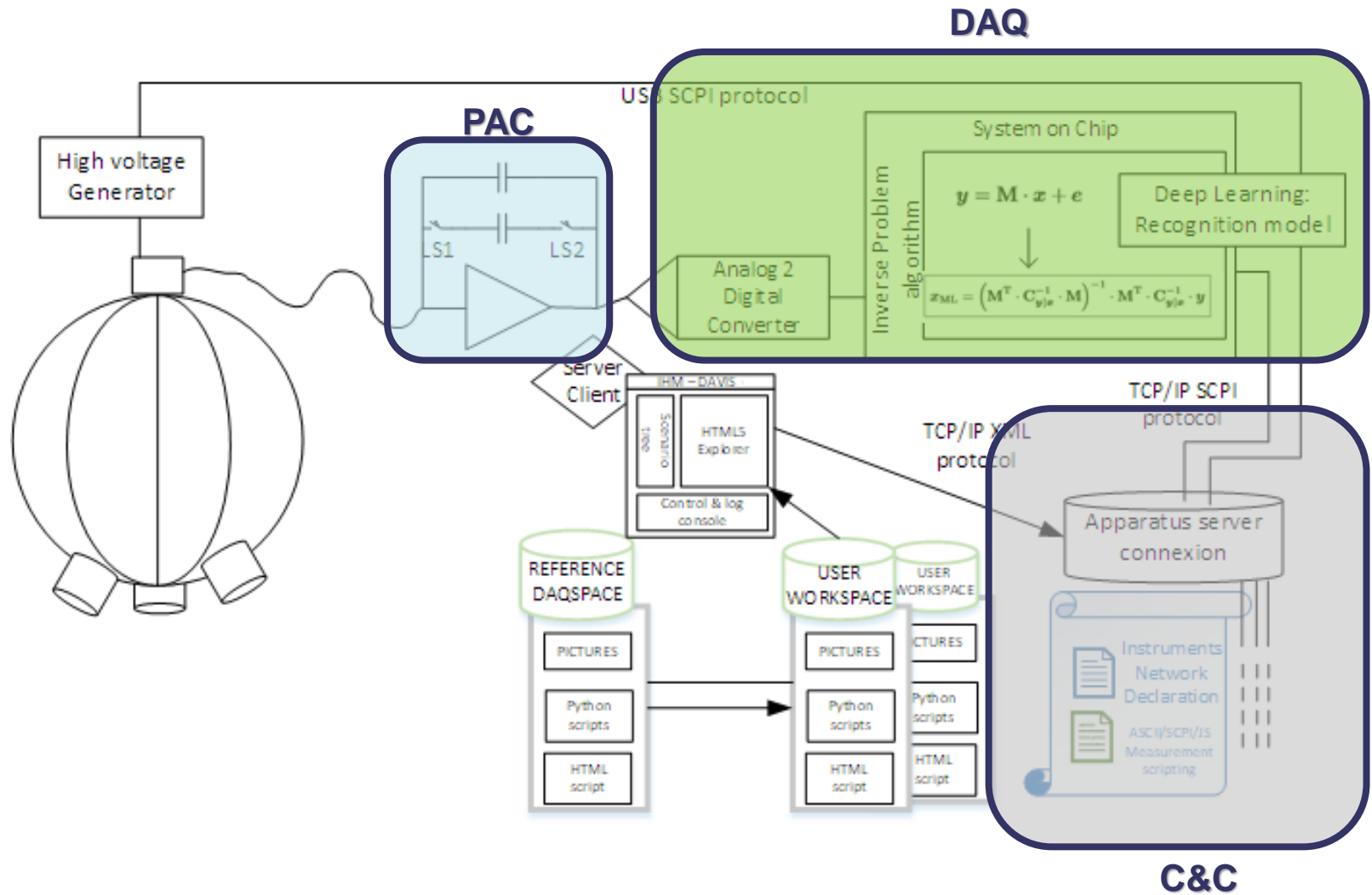
Proto 3: TPC cylindrique non certifiée

## Objectifs:

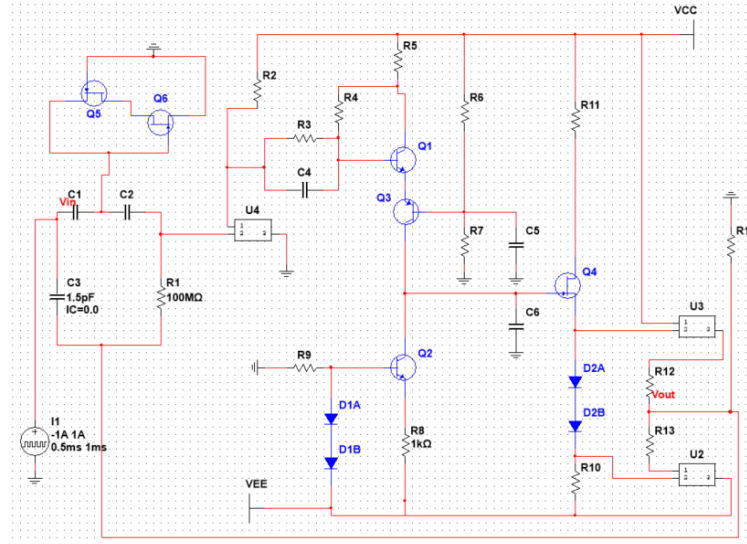


- ❑ Développement d'un amplificateur de charge versatile.
- ❑ Développement d'une voie d'acquisition avec incorporation d'algorithme de reconnaissance de forme.
- ❑ Création d'algorithmes d'IA pour reconnaissance de forme d'ondes.
- ❑ Nouveau système de contrôle & commande adaptable et évolutif au fil des tests.

# Synoptique du système

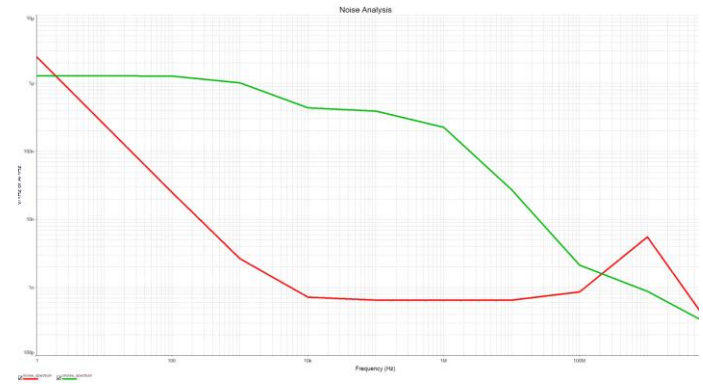
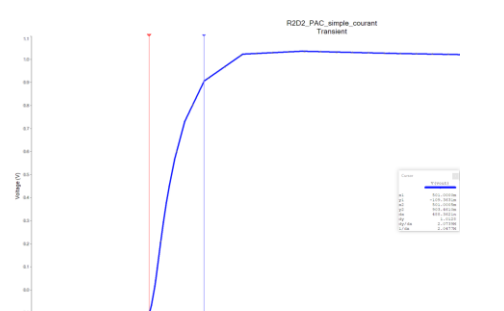
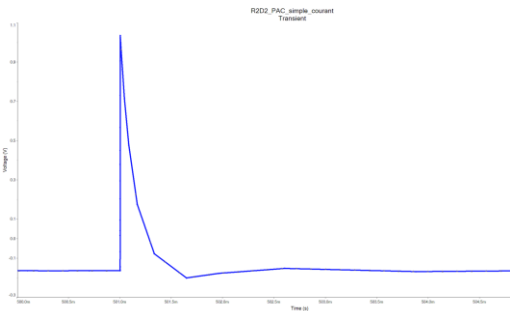
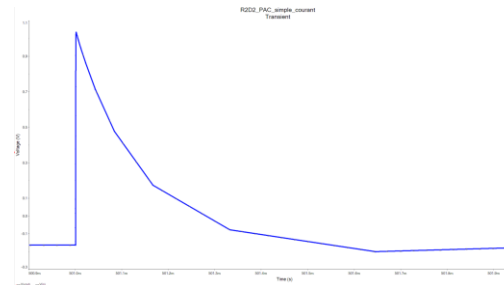
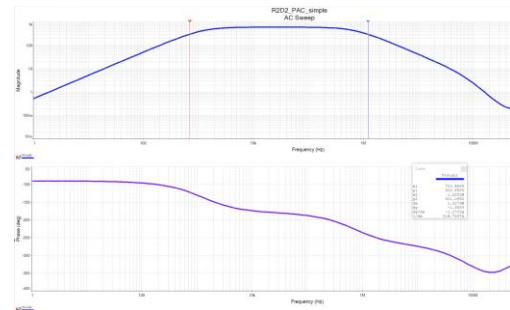


# Le préamplificateur de charge

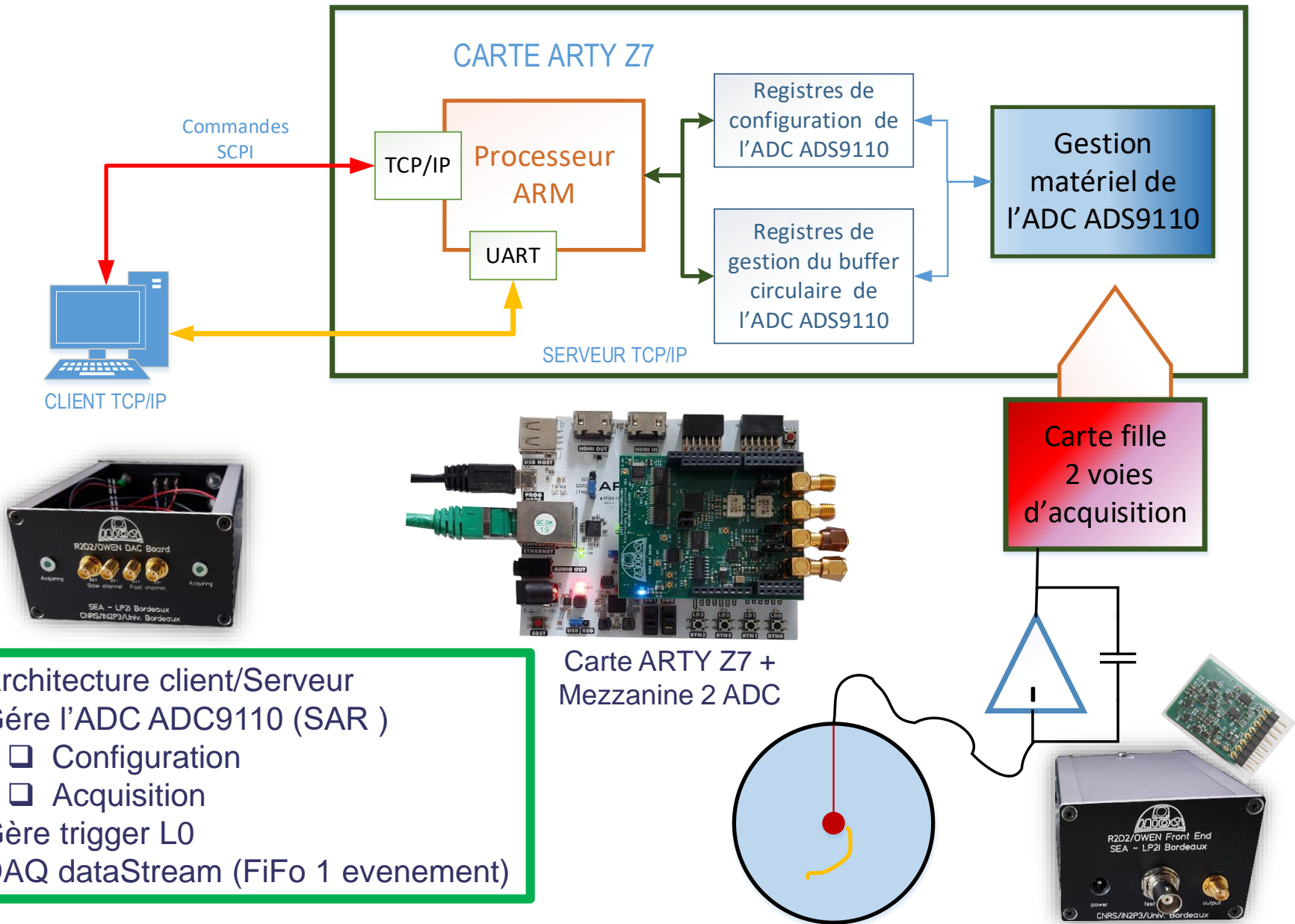


PAC cascode + Ampli  
classe A  
 $C_f = 1,5\text{pF}$   
 $V_{out} = +4,3\text{V}/-3,4\text{V}$   
 $T_{reset} = 360\mu\text{s}$   
 $BP = 1,2\text{MHz}$   
 $\text{Noise} = 418\mu\text{Vrms}$

## BODE



# Système d'acquisition : Architecture



- Architecture client/Serveur
- Gère l'ADC ADC9110 (SAR )
  - Configuration
  - Acquisition
- Gère trigger L0
- DAQ dataStream (FiFo 1 evenement)

# Système d'acquisition : ADC AD9110

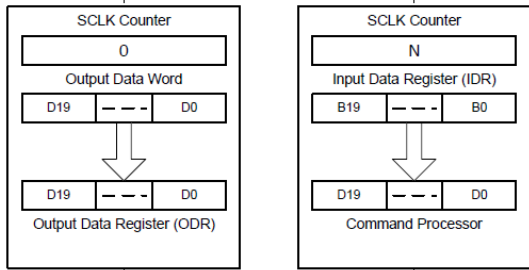
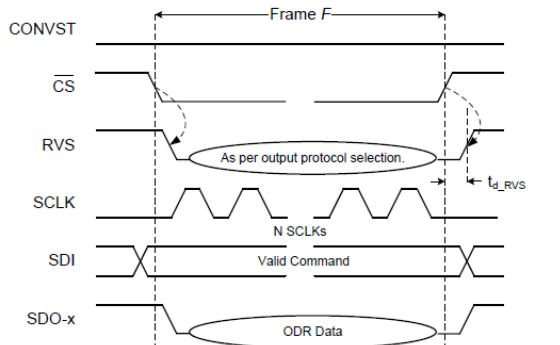
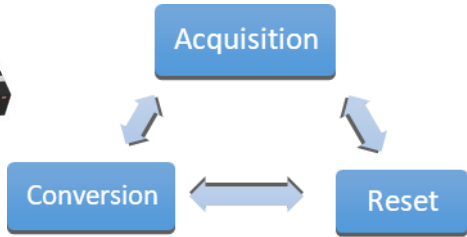
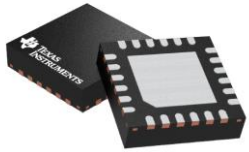


Figure 46. Data Transfer Frame

Configuration & Acquisition en même temps

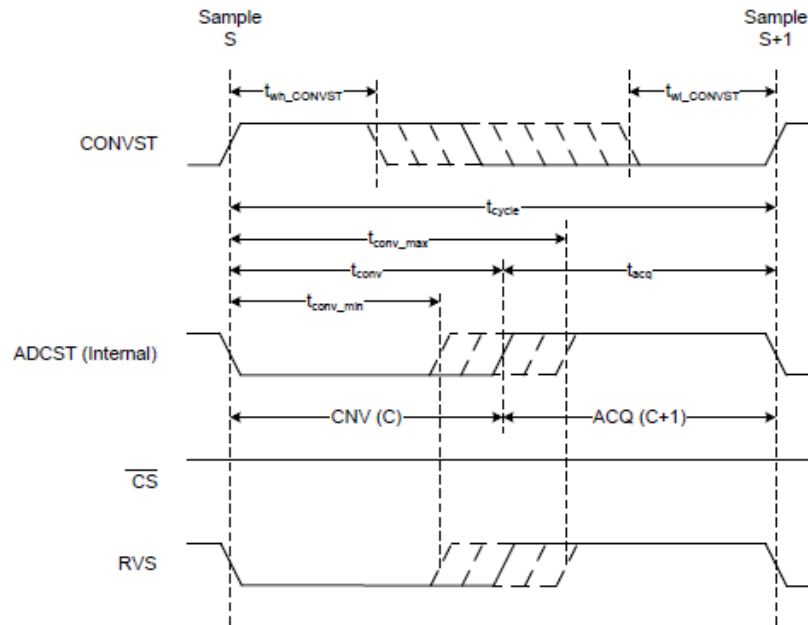
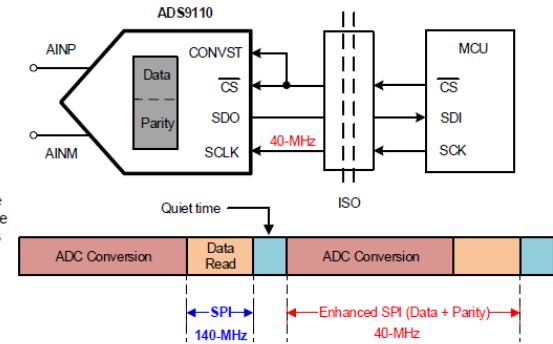
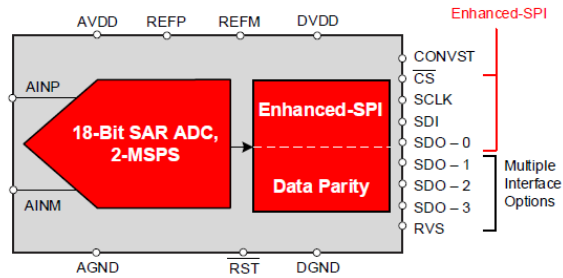


Figure 43. Typical Conversion Process

Un mode de lecture sélectionné : Echantillon sur 4 voies à 40MHz

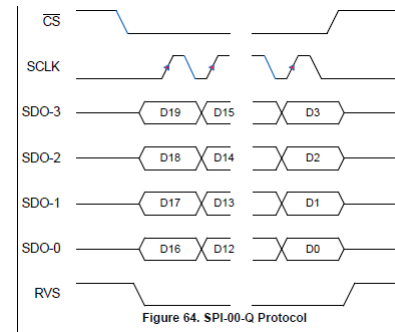
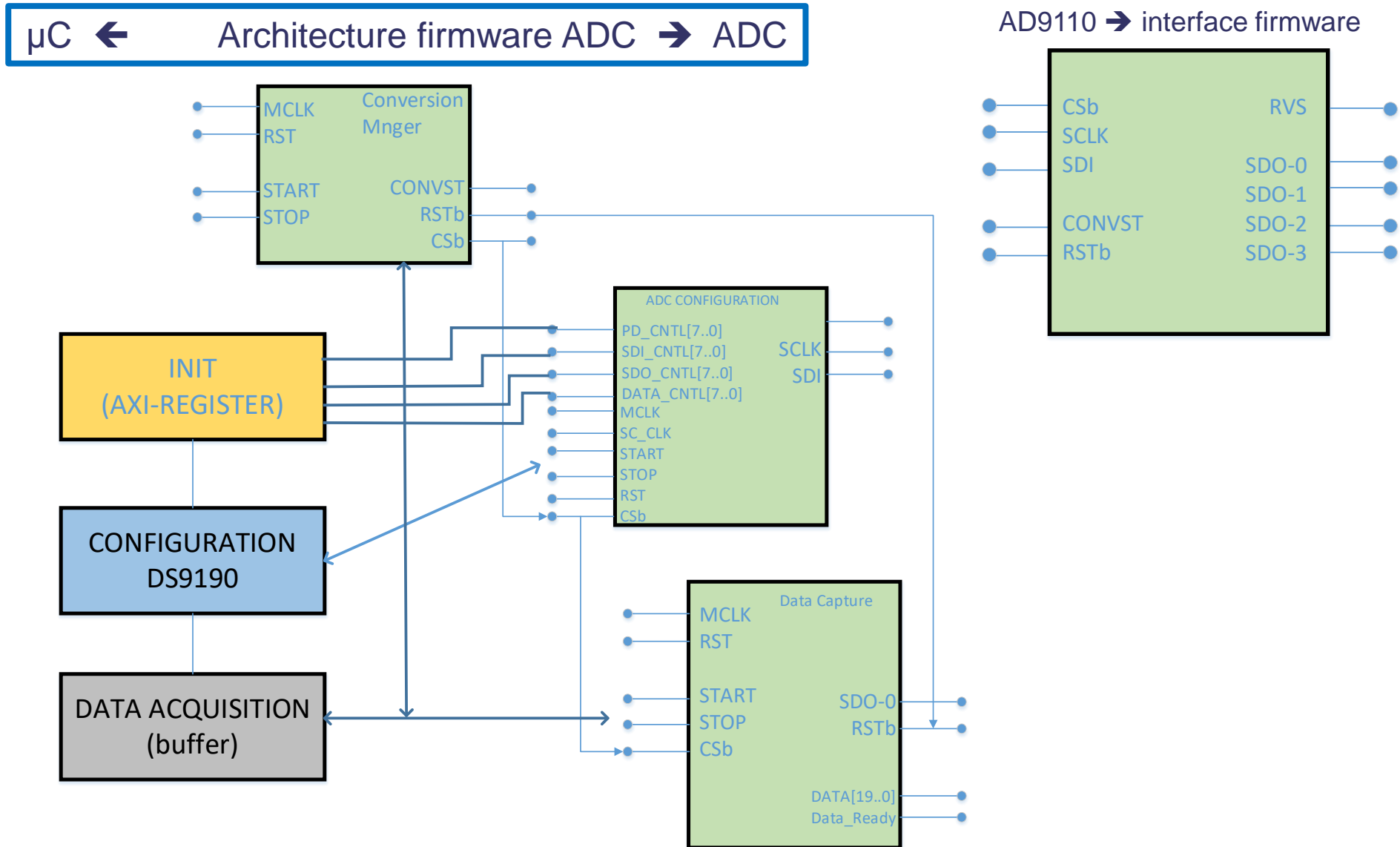


Figure 64. SPI-00-Q Protocol

Conversion piloté par commande externe

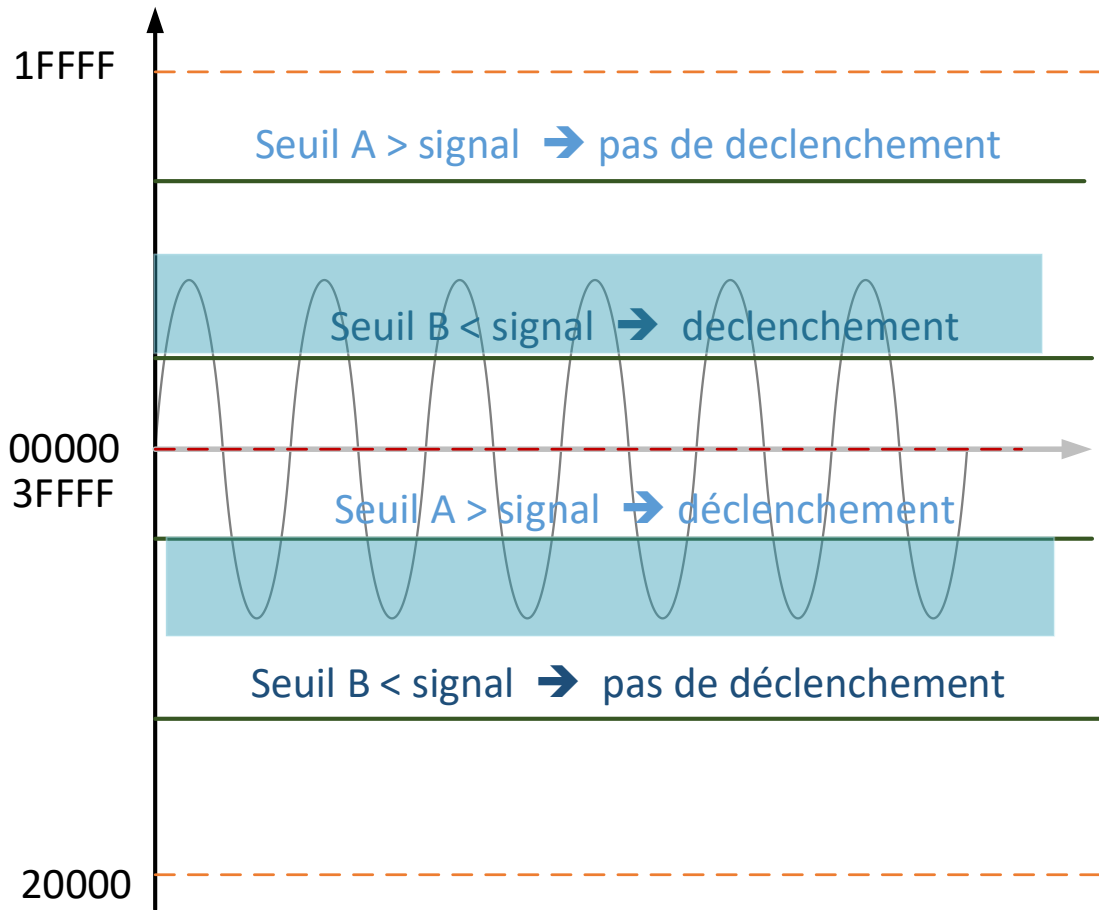


# Système d'acquisition: Firmware ADC

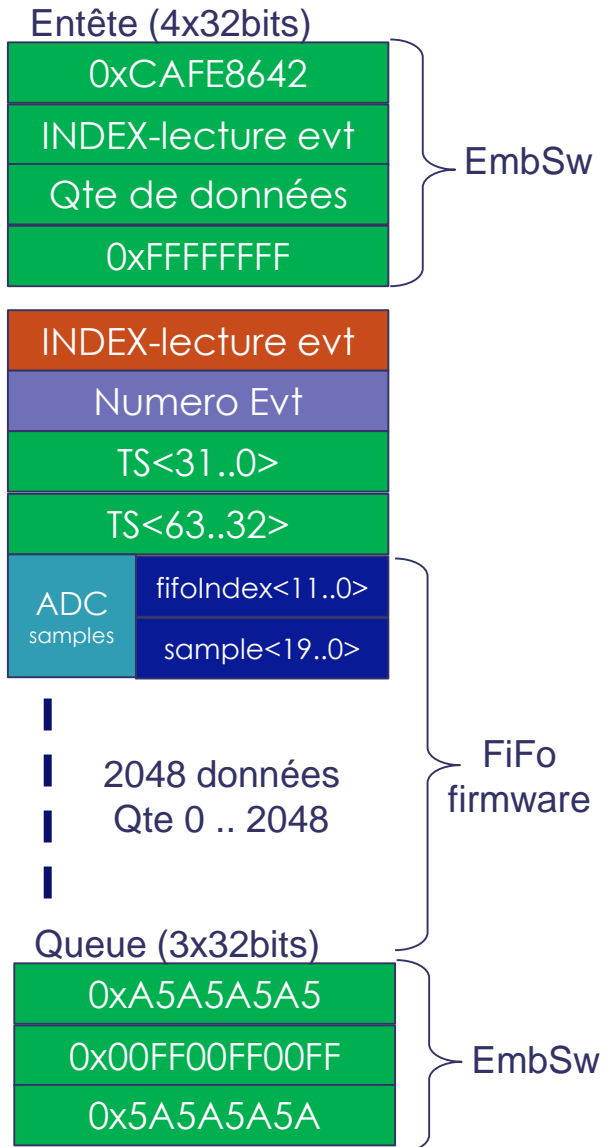




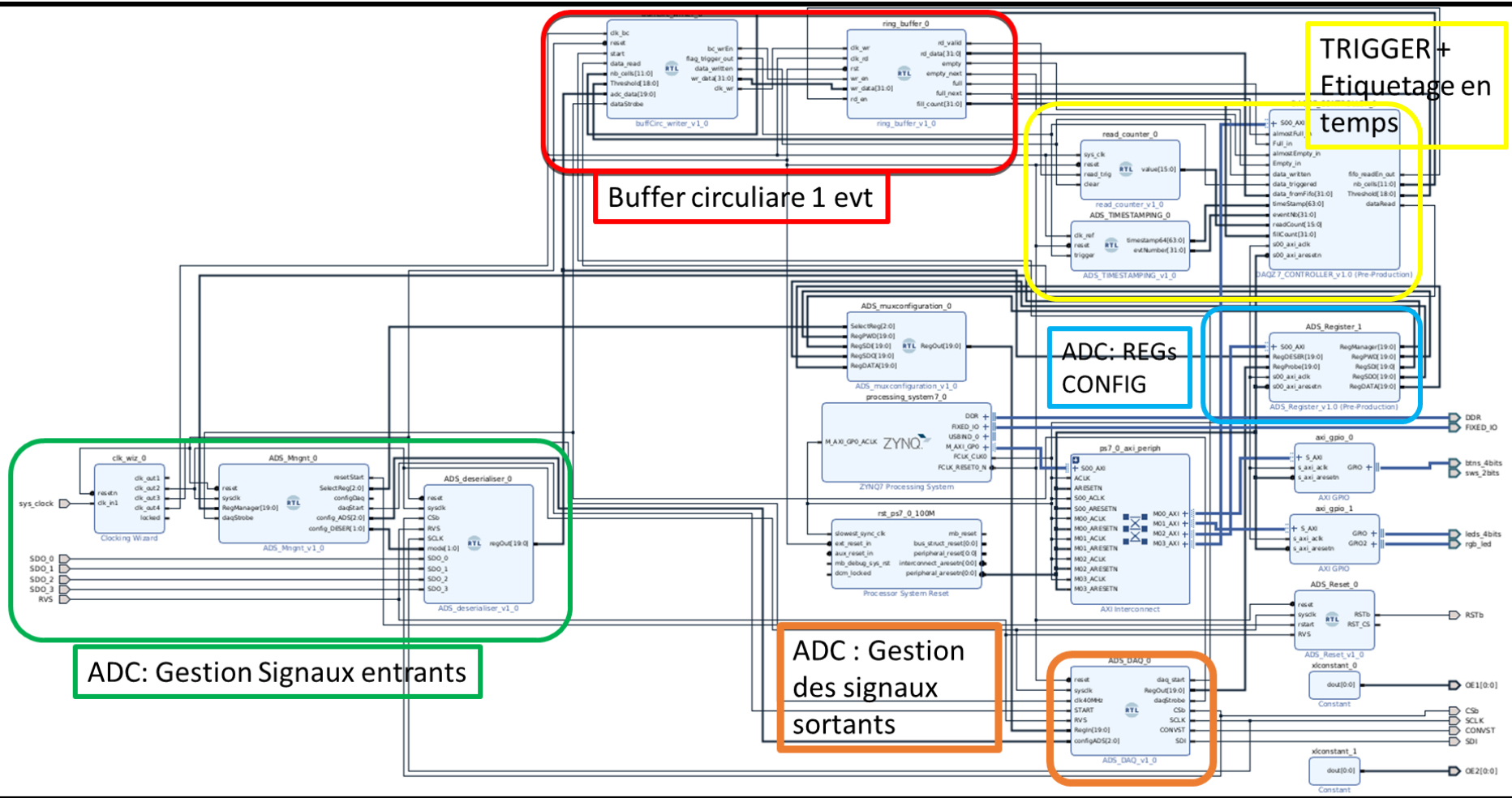
# Système d'acquisition: Trigger & Format



Creation :  
 Numero d'évenement 32 bits  
 Etiquetage en temps 64bits x 5ns



# Système d'acquisition: Firmware global



# Système d'acquisition: logiciel embarqué

OS: Free RTOS (Xilinx – Vitis )  
Langage C

Commande reçue

Envoie des paramètres dans les registres du firmware:  
<REG:WRITE @ data>  
<REG:READ @>

RESET ADC

SELECTION MODE DE LECTURE/ECRITURE DE L' ADC

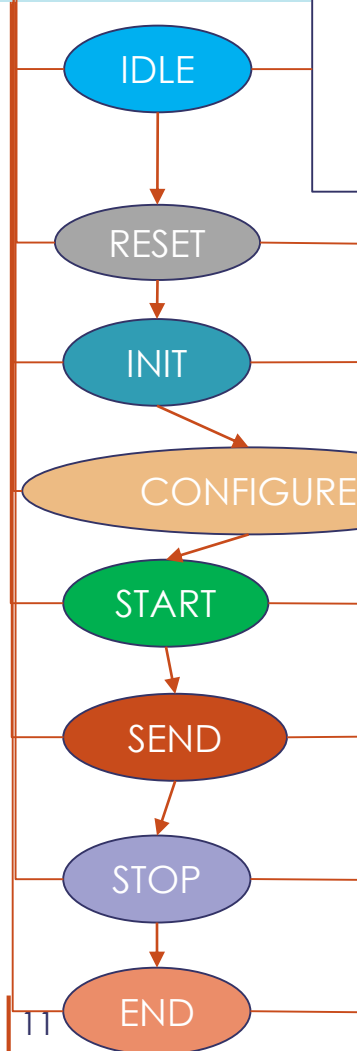
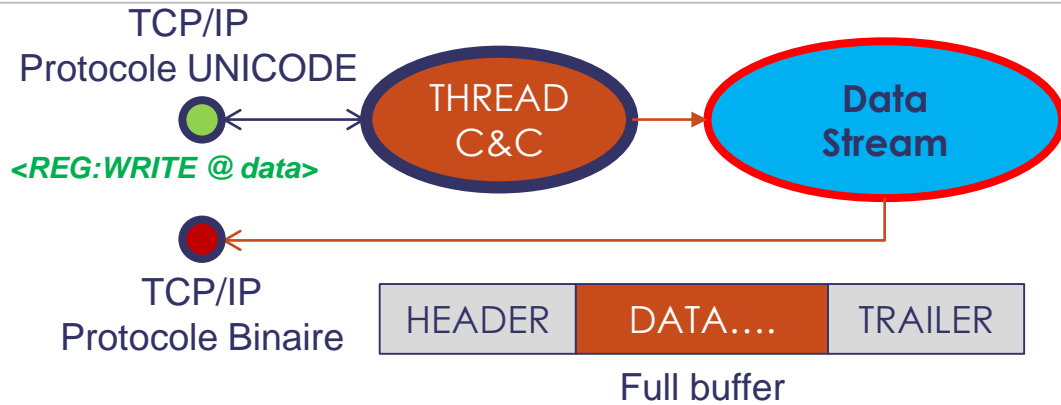
ECRITURE DES VALEURS DES 4 REGISTRES DE L'ADC (SDI, SDO, PWD, DATA)

DECLENCHEMENT DE LA SEQUENCE DE CONVERSION+ACQUISITION ADC. ENREGISTREMENT DE L'EVENEMENT SUR TRIGGER

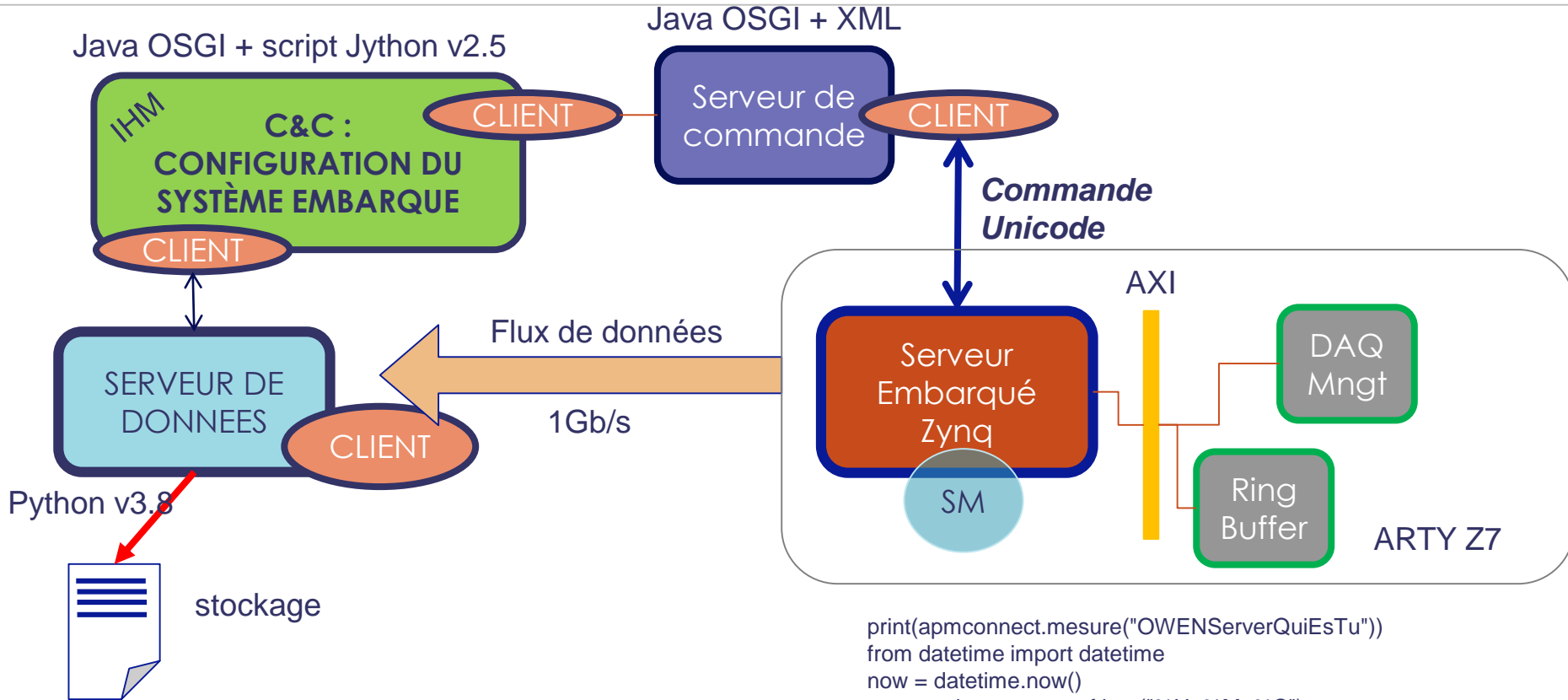
CREATION D'UN Thread DE LECTURE DU BUFFER  
CREATION HEADER:TRAILER  
ENVOIE CONTINU [T] [D] [H]

ARRET DU Thread DAQ  
ARRET ADC  
RETOUR à IDLE

RETOUR à IDLE



# Système d'acquisition : logiciel C&C

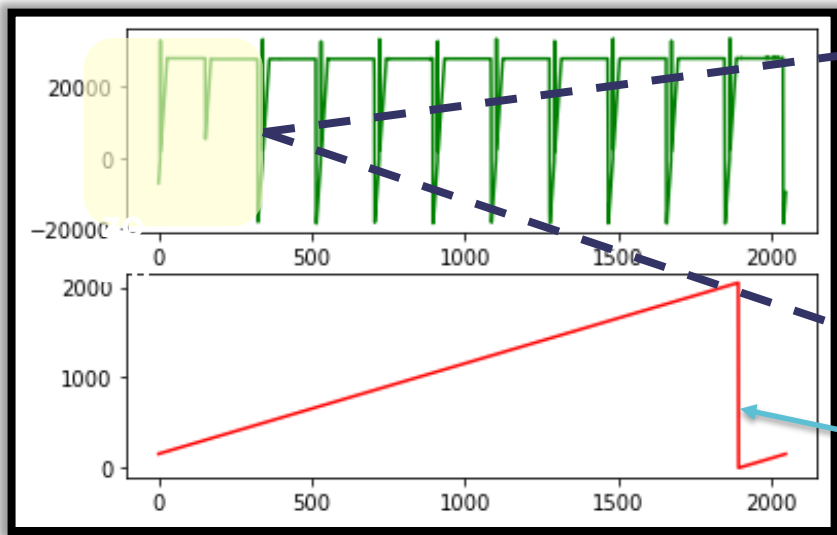


```
print(apmconnect.measure("OWENWRITEREG ID=9 data=0xCAFEA5A5"))
print(apmconnect.measure("OWENWRITEREG ID=11 data=0x5A5AFEDC"))
print(apmconnect.measure("OWENREADREG ID=9"))
print(apmconnect.measure("OWENMODE state=0"))
print(apmconnect.measure("OWENTHRESHOLD thres=0x50032"))
print(apmconnect.measure("OWENNBCELLS NB=2200"))
```

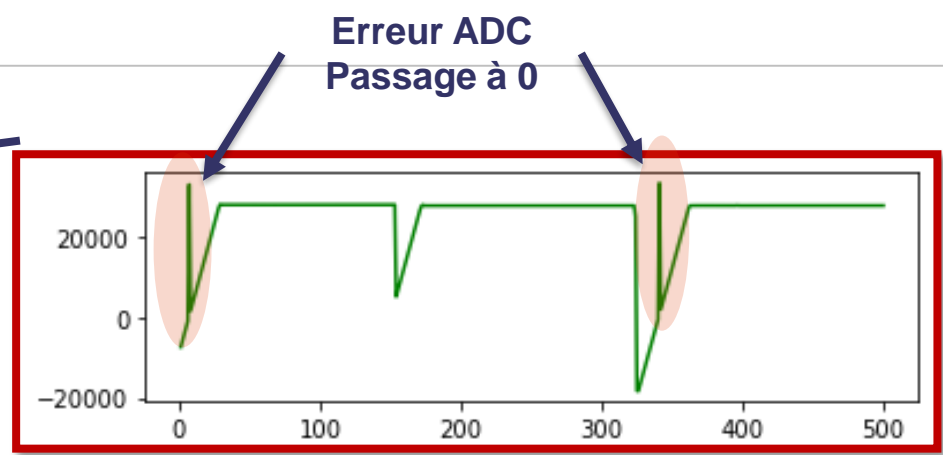
```
print(apmconnect.measure("OWENSTATE state=RESETTING"))
print(apmconnect.measure("OWENSTATE state=INITIALISING"))
print(apmconnect.measure("OWENSTATE state=CONFIGURING"))
print(apmconnect.measure("OWENSTATE state=RUNNING"))
```

```
print(apmconnect.measure("OWENServerQuiEsTu"))
from datetime import datetime
now = datetime.now()
current_time = now.strftime("%H_%M_%S")
DataPath = "D:\\\\DATA"
file2Store = DataPath+"\\\\OWENDATA_"+current_time+'.bin'
print(apmconnect.measure("OWENServerFile fname="+file2Store))
print(apmconnect.measure("OWENServerDatarequest qty=3"))
print(apmconnect.measure("OWENServerStart val=TRUE"))
print(apmconnect.measure("OWENServerThread"))
.....
print(apmconnect.measure("OWENServerStart val=FALSE"))
print(apmconnect.measure("OWENServerDataRecv"))
print(apmconnect.measure("OWENSTATE state=STOPPING"))
print(apmconnect.measure("OWENSTATE state=ENDING"))
print(apmconnect.measure("OWENSTATE state=WAITING"))
```

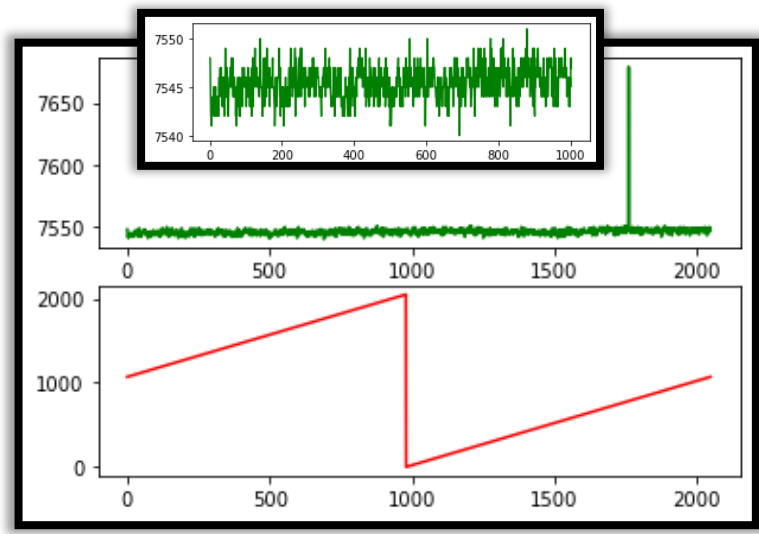
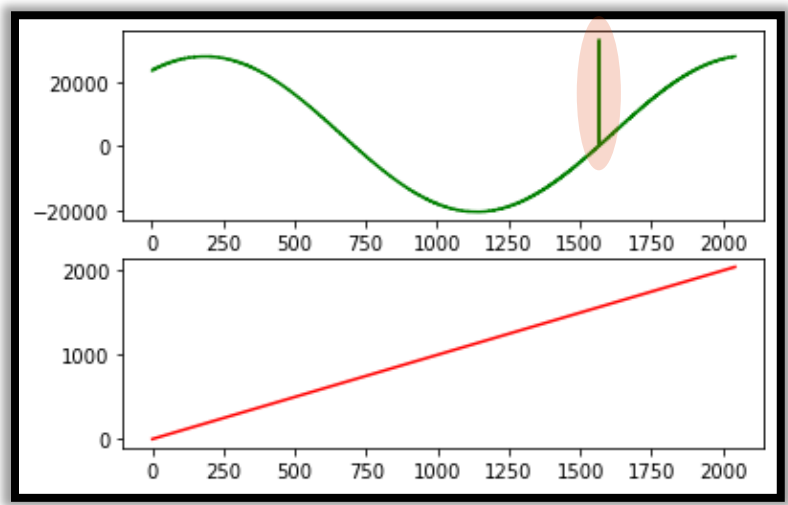
# Résultats préliminaires:



Signal échantillonné



Vérification de la cohérence des données avec indexes (2048 ech.) des échantillons reçus

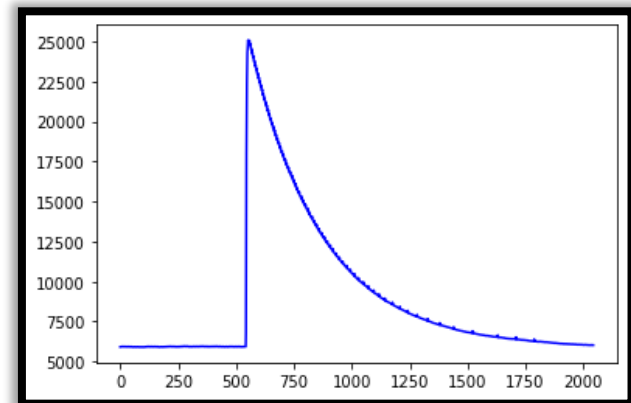
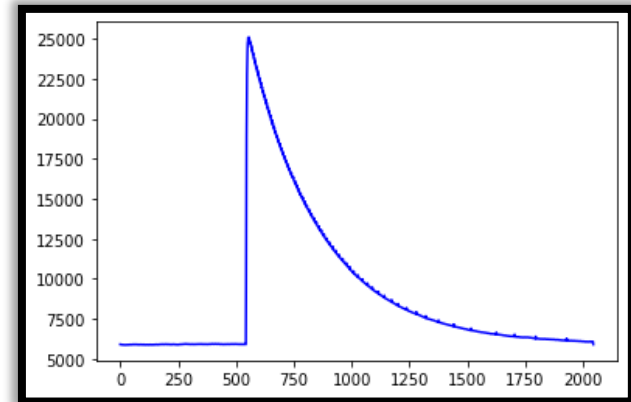
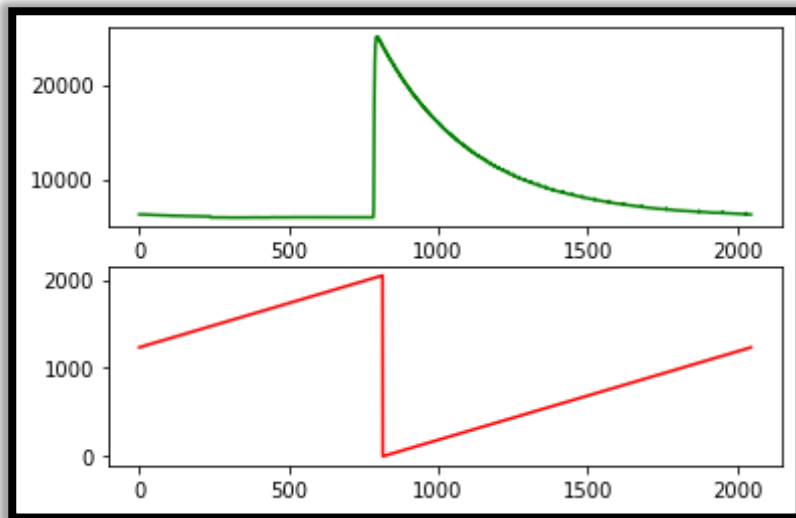


Bruit: +/- 3 LSB

# Résultats préliminaires:



Signal avec PAC + test d'injection au générateur



# Conclusion

- Le système DAQ est fonctionnel
- Il y a adéquation entre le PAC et l'ADC
- On a flux continu de données à 1Gb/s
- Système on module validé
- Validation de l'utilisation de freeRTOS, serveur et client en python, C, Java

- Reste à Valider:
  - Monitoring on Line de l'acquisition
  - Amélioration des modes de déclenchement (inversion de polarité)
- Prise de données sur détecteur cylindrique
  - Analyse en cours
- Développement de la voie rapide (200MHz)
  - ➔ Utilisation avec Algo IA (Think)