

# Session 3 – Wattmètre AC en temps réel Partie 1



EE-390b TP en Conversion d'Énergie

Sylvain Robert, Simone Rametti

- Que va-t-on faire dans ce TP ?
- Introduction
  - Applications en temps réel
  - National Instruments Compact RIO
- Architecture du Compact RIO
  - Front End
  - FPGA
  - DMA FIFO

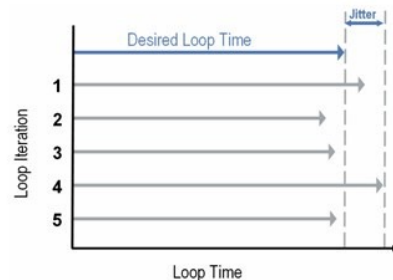
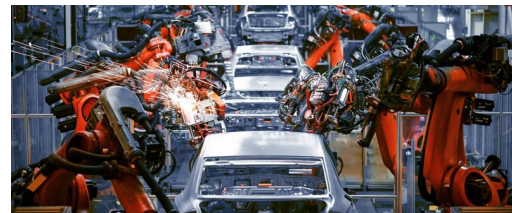
# Que va-t-on faire dans ce TP ?

- Créer un wattmètre AC monophasé afin de mesurer la puissance consommée au primaire d'un transformateur
- Pour ça nous avons besoin :
  - Quelque chose à mesurer (le transformateur)
  - Une mesure de tension
  - Une mesure de courant
  - Un système d'acquisition de données dans lequel :
    - On acquière les mesures de tension et de courant en temps réel (partie FPGA)
    - On les traite afin d'en calculer la puissance (partie RTCPU)
- Aujourd'hui on va se concentrer sur la partie FPGA

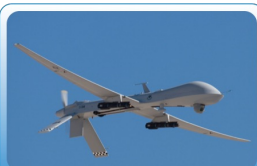
# Introduction

## Applications en temps réel

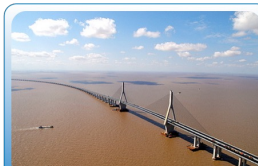
- Une application en temps réel fonctionne avec un timing très précis et un haut degré de fiabilité.
- Applications où un retard peut entraîner un risque pour la sécurité.
- Un système en temps réel doit avoir un **temps maximum connu** pour chacune des opérations critiques qu'il effectue (par exemple, le déploiement d'un airbag).
- Déterminisme : la synchronisation d'une application peut être garantie avec une certaine marge d'erreur (**gigue/jitter**).
- Exemples : Ligne d'assemblage automatisée, systèmes de contrôle en boucle fermée, surveillance 24 heures sur 24 et 7 jours sur 7.



Event Response



Closed-Loop Control

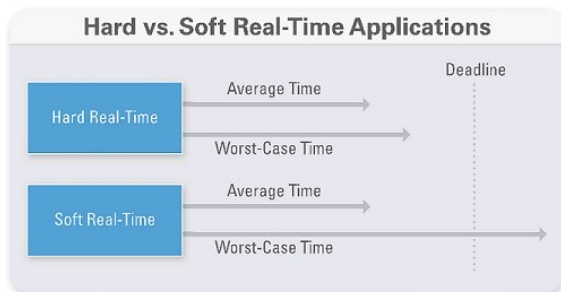


Embedded Monitoring

# Introduction

## Applications en temps réel

- Le contraire d'un système d'exploitation en temps réel (Real-Time Operating System (RTOS)) tel que NI Linux Real-Time est un système d'exploitation à usage général (General-Purpose Operating System (GPOS)), tel que Windows.
- Le RTOS applique un système de priorité strict pour garantir le déterminisme des tâches sensibles.
  - Distinction entre les tâches de haute priorité, ou critiques, qui nécessitent un déterminisme et les tâches de faible priorité qui peuvent être exécutées de manière non déterministe (par exemple, la communication).
  - Une conception plus minutieuse des applications est nécessaire pour identifier les tâches sensibles et leurs priorités.
- GPOS garantit un certain temps de processeur pour chaque tâche.
  - Ils sont conçus pour exécuter un grand nombre de tâches simultanément et, par conséquent, les priorités ne sont pas toujours respectées.



# Introduction

## National Instruments Compact RIO

- Un Compact RIO est un contrôleur en temps réel industriel (real-time industrial controller).
- Il utilise un NI Linux RTOS et un châssis qui contient une **FPGA** programmable par l'utilisateur. Il prend en charge les modules d'E/S industrielles de NI et de fournisseurs tiers.
- L'architecture d'E/S (I/O) reconfigurables LabVIEW (RIO) associe **LabVIEW** à du matériel prêt à l'emploi reconfigurable. Elle repose sur quatre composants :

- Un processeur (CPU)
- Une FPGA
- Des entrées/sorties (Front End)
- LabVIEW



# Architecture du CompactRIO



## Contrôleur :

- Où le RTOS tourne
- Généralement appelé simplement RT

## Châssis :

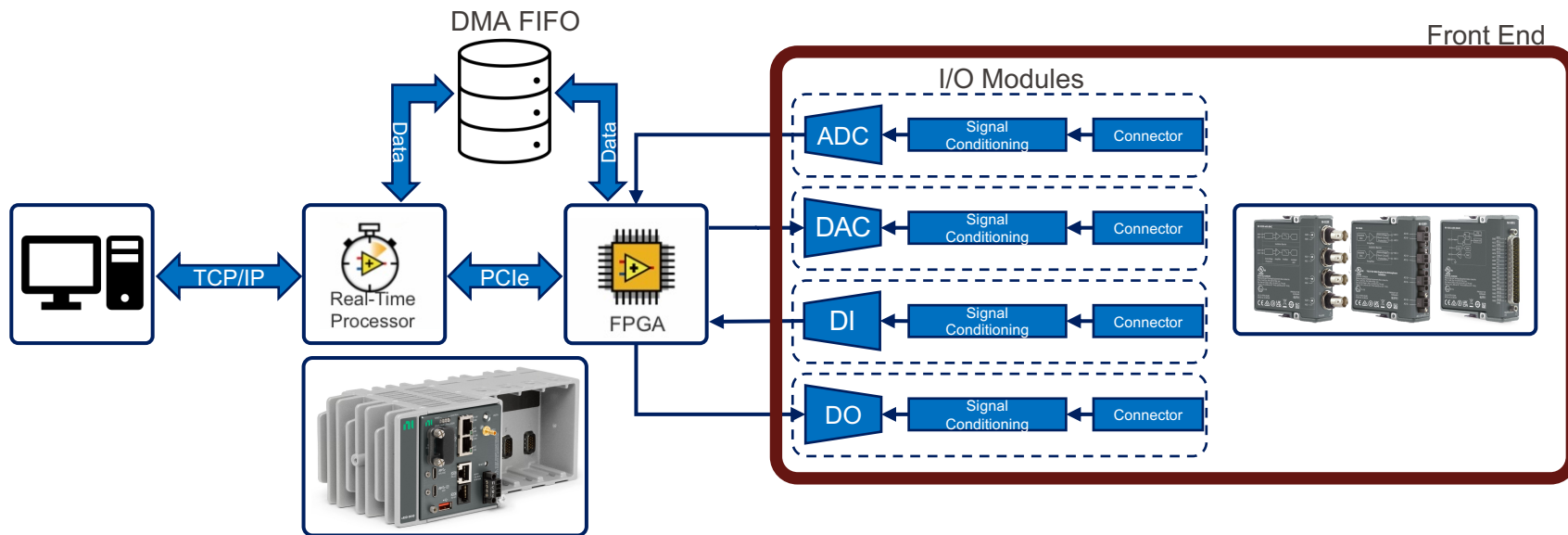
- Centre du système
- Directement connecté aux E/S (I/O)
- Connexion directe à la FPGA

## Modules E/S (I/O Modules) :

- Circuit pour conditionner les signaux
- ADC/DAC
- Interface entre la partie frontale (front-end) et la FPGA

# Architecture du CompactRIO

■ SEL-GE



# Architecture du CompactRIO

## Front End

- C'est l'interface permettant d'acquérir les mesures et de les convertir de l'analogique au numérique (ADC).
- Nous pouvons simplifier le processus de conversion en 4 sous-processus :
  - Sample / Echantillonnage
  - Hold / Maintien
  - Quantize / Quantification
  - Encoder / Encodage



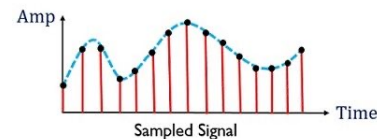
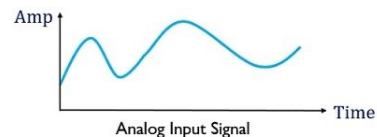
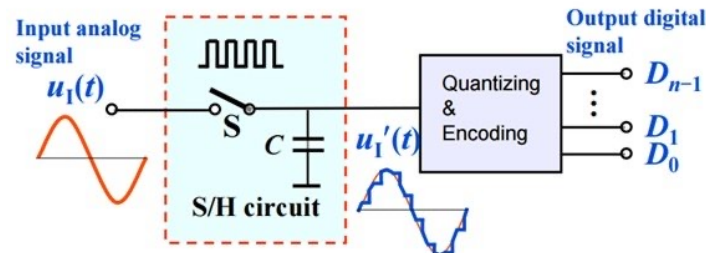
### I/O Modules:

- Circuit pour conditionner les signaux
- ADC/DAC
- Interface entre la partie frontale (front-end) et la FPGA
- Nous utiliserons le AI module NI9215
- $\pm 10$  V, 16-Bit, 4 channels

# Architecture du CompactRIO

## Front End

- **Sample (Echantillonnage):** le signal analogique est échantillonné à un intervalle de temps spécifique (période d'échantillonnage)
- **Hold (Maintien):** maintenir l'amplitude de l'échantillon constante jusqu'au prélèvement de l'échantillon suivant
- **Quantize (Quantification):** Convertit l'amplitude analogique du bloc de maintien en une amplitude discrète
- **Encoder (Encodage):** convertit le signal discret en forme binaire



Input and Output Response

Electronics Desk

# Architecture du CompactRIO

## Front End

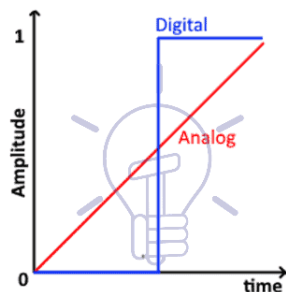
- **Resolution (Résolution):** nombre de bits représentant l'amplitude du signal numérique.
- **Width of the step (Largeur du pas):**

$$\Delta V = \frac{V_{range}}{2^n}$$



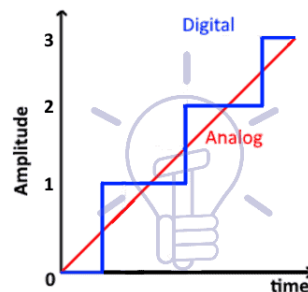
$$\Delta V = \frac{20\text{ V}}{2^{16}} = 305\text{ }\mu\text{V}$$

Number of steps =  $2^1 = 2$



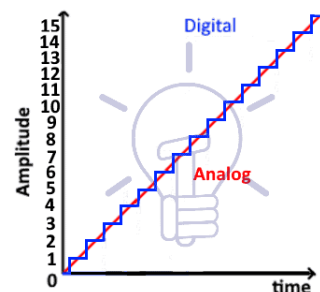
1-bit Resolution

Number of steps =  $2^2 = 4$



2-bit Resolution

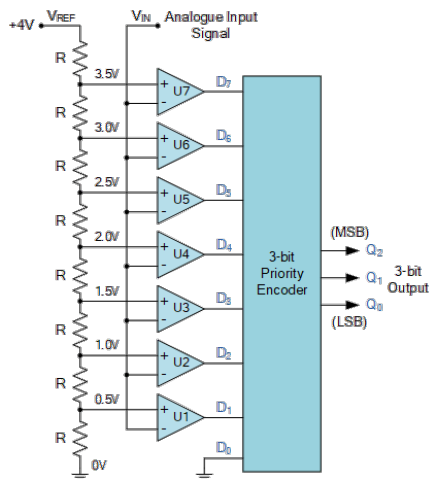
Number of steps =  $2^4 = 16$



4-bit Resolution

# Architecture du CompactRIO

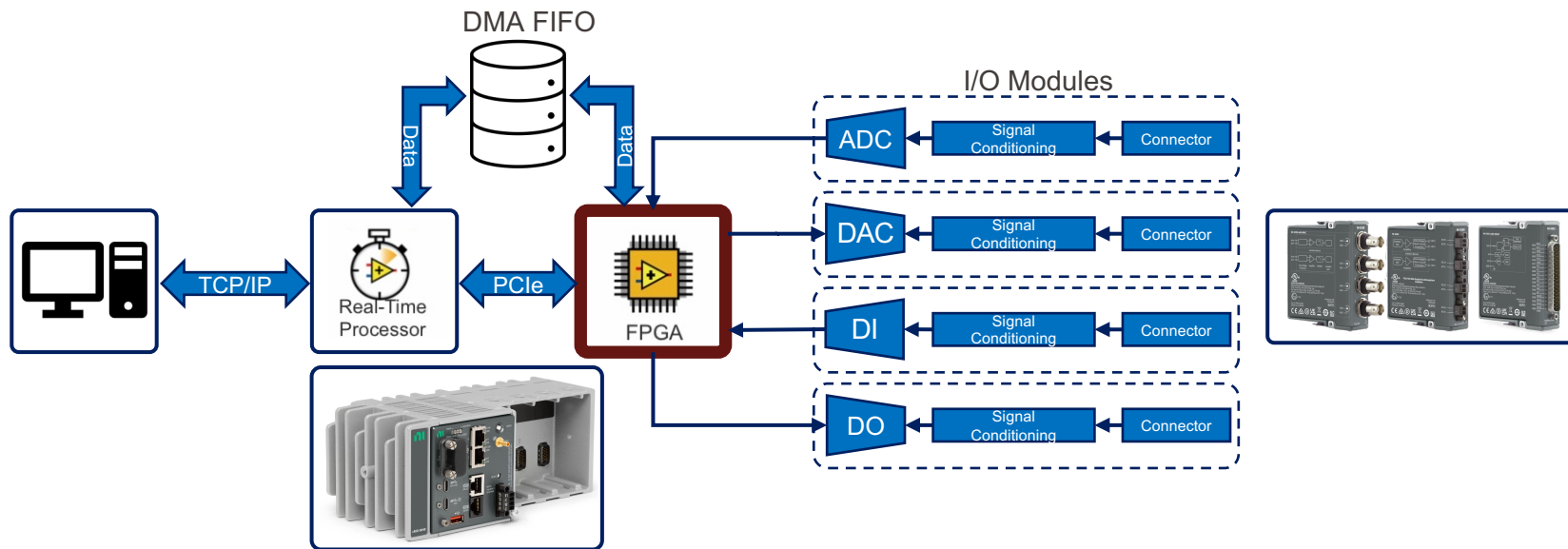
## Front End



Analogue Input Voltage ( $V_{IN}$ )	Comparator Outputs								Digital Outputs		
	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
0 to 0.5 V	0	0	0	0	0	0	0	0	0	0	0
0.5 to 1.0 V	0	0	0	0	0	0	1	X	0	0	1
1.0 to 1.5 V	0	0	0	0	0	1	X	X	0	1	0
1.5 to 2.0 V	0	0	0	0	1	X	X	X	0	1	1
2.0 to 2.5 V	0	0	0	1	X	X	X	X	1	0	0
2.5 to 3.0 V	0	0	1	X	X	X	X	X	1	0	1
3.0 to 3.5 V	0	1	X	X	X	X	X	X	1	1	0
3.5 to 4.0 V	1	X	X	X	X	X	X	X	1	1	1

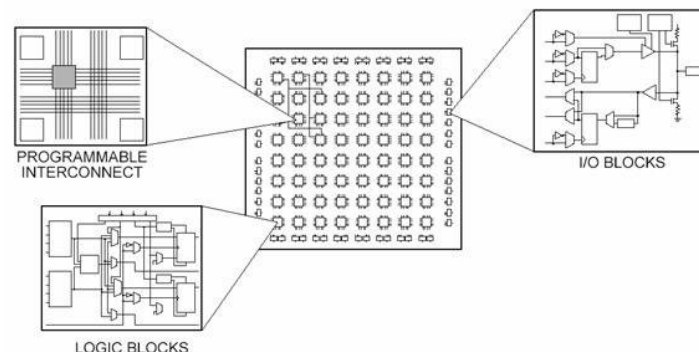
# Architecture du CompactRIO

■ SEL-GE



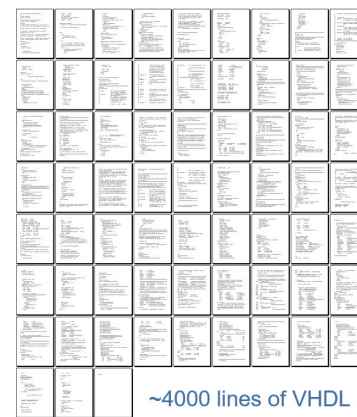
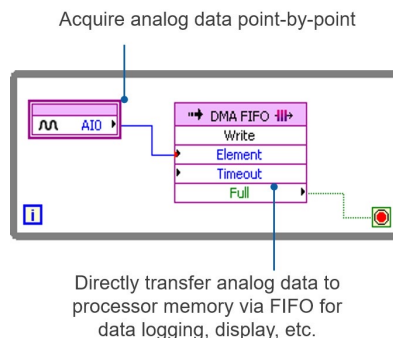
# Architecture du CompactRIO FPGA

- Field-Programmable Gate Array (FPGA) est un circuit intégré reprogrammable qui contient un ensemble de blocs logiques.
- Exécution parallèle des tâches intrinsèquement. Chaque tâche indépendante est assignée à une section dédiée de la puce et fonctionne de manière autonome.
- Il garantit le déterminisme temporel.
- Contrairement au RTOS, la logique de l'application est mise en œuvre dans des **circuits matériels** plutôt que d'être exécutée au-dessus d'un système d'exploitation.
- Advantages:
  - Flexibilité
  - Hardware-timed speed (déterminisme)
  - Fiabilité
  - Parallélisme



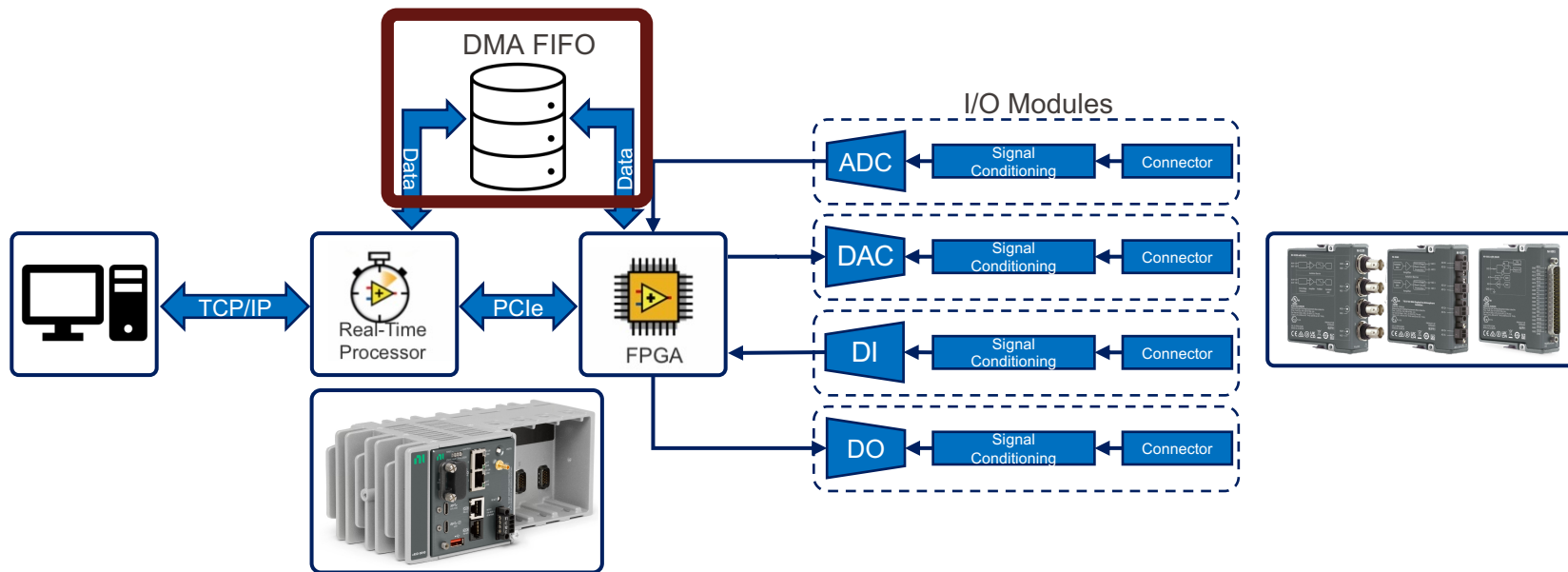
# Architecture du CompactRIO

- La conception de bas niveau des puces FPGA est fastidieuse et nécessite des compétences de programmation très élevées avec des langages de description de matériel (HDL) tels que le VHDL (langage textuel séquentiel).
- LabVIEW a éliminé certains des principaux obstacles du processus de conception HDL traditionnel.
- Le parallélisme est naturellement représenté dans LabVIEW. Il n'est pas nécessaire d'avoir des connaissances en HDL pour programmer un FPGA à l'aide de LabVIEW.
- Le compilateur se charge de traduire le diagramme LabVIEW en fichier de configuration FPGA (bitstream).
- En général, son horloge interne fonctionne à 40 MHz et le temps est exprimé en « ticks ».



# Architecture du CompactRIO

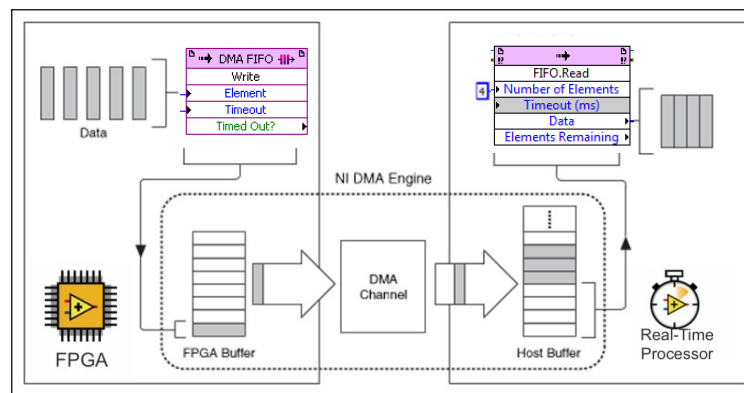
■ SEL-GE



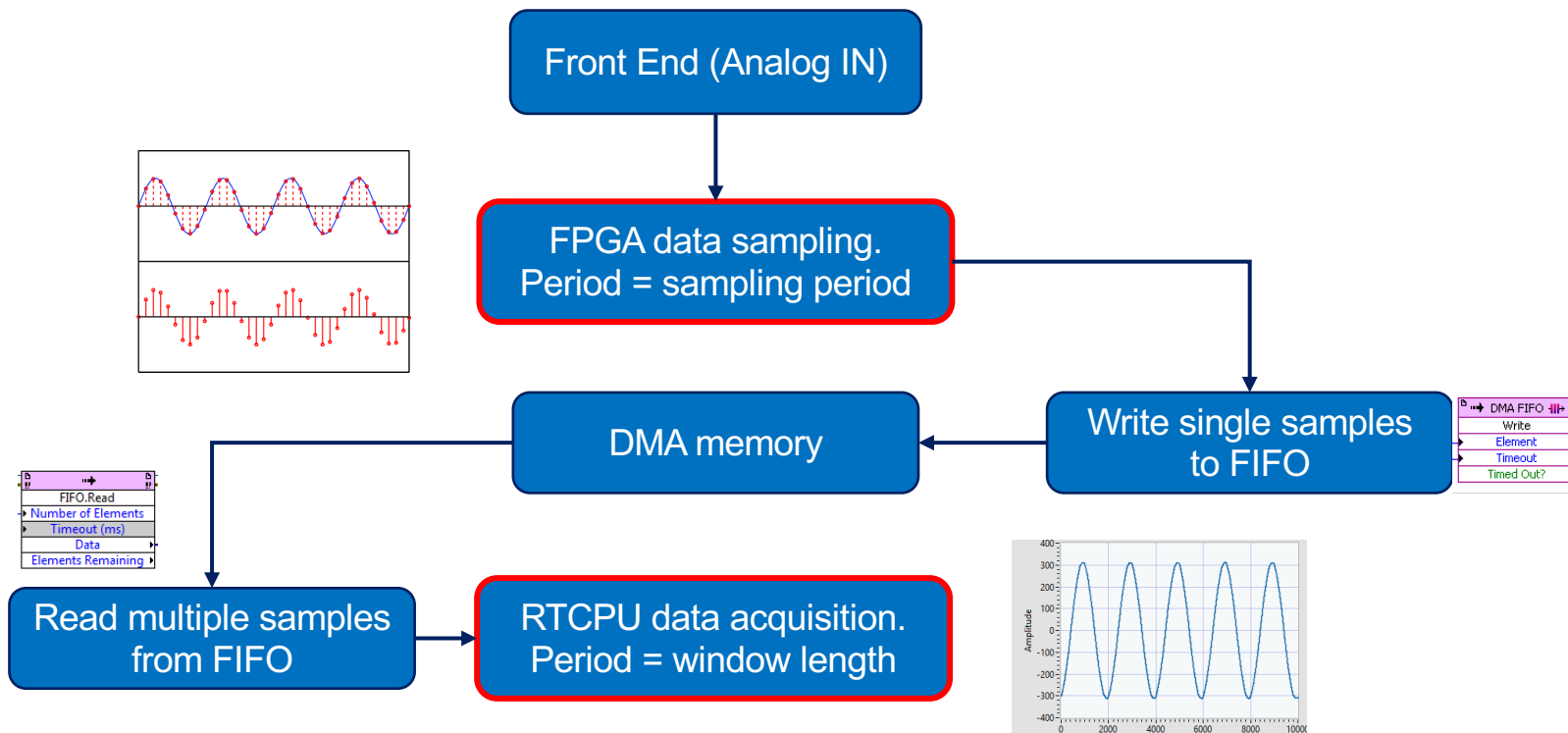
# Architecture du CompactRIO

## DMA FIFO

- Nous utiliserons le Direct Memory Access (DMA) FIFO pour transférer les données (formes d'onde) de la FPGA au CPU du RT.
- 2 DMA FIFOs sont nécessaires. Un FIFO sur la FPGA et un sur le CPU du CPU. Ils se comportent comme une seule et même personne.
- Les DMA FIFOs accèdent directement à la mémoire pour transférer des données. Il alloue la quantité de mémoire demandée à la fois sur la cible FPGA et sur la cible du CPU du RT.
- Les performances du FPGA et du CPU du RT ne sont pas affectées par le transfert de données. C'est une meilleure option lorsqu'une grande quantité de données doit être transférée.



# Structure software du TP



Choosing a Software Architecture for NI Linux Real-Time CompactRIO Controllers

<https://www.ni.com/en/shop/linux/choosing-a-software-architecture-for-ni-linux-real-time-compactrio.html>

What is a Real-Time Operating System (RTOS)?

<https://www.ni.com/en/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-labview-real-time-module/what-is-a-real-time-operating-system--rtos--.html>

Understanding Communication Options Between the Windows HMI, RT Processor, and FPGA

<https://www.ni.com/en/support/documentation/supplemental/16/understanding-communication-options-between-the-windows-hmi-rt-.html#RTHMICommunication>

Direct Memory Access (DMA) FIFOs

<https://www.ni.com/docs/en-US/bundle/labview-nxg-fpga-targets/page/dma-fifos.html>

FPGA Fundamentals: Basics of Field-Programmable Gate Arrays

<https://www.ni.com/en/shop/electronic-test-instrumentation/add-ons-for-electronic-test-and-instrumentation/what-is-labview-fpga-module/fpga-fundamentals.html>