

Journée Alpesview : Présentation William Boucher

Mon parcours

Thomson-CSF : Thèse CIFRE: Distribution de clé quantique par codage temporel

PHASICS (2005-2020) : Ingénieur de recherche en développement d'applications en métrologie optique

- Ingénieur optique expérimentale : faisabilité
- Ingénieur R&D : maquettes de définition et prototypes
- Responsable du service R&D optique
 - développement de produits et solutions spécifiques pour des clients
 - stratégie de développement
 - garant des spécifications



Addup (2021-2024) : Ingénieur R&D

- référent optique et laser
- instrumentation et métrologie
- veille technologique



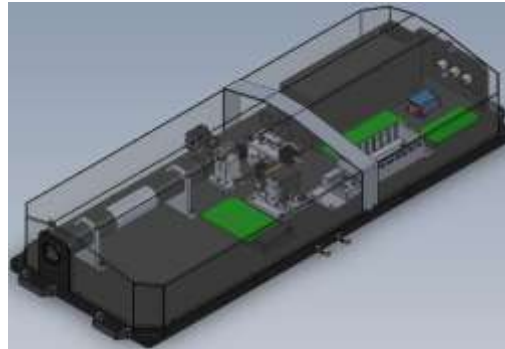
LaMP (depuis juin 2024) : Ingénieur de recherche (CNRS)

- soutien technique et expérimental sur mesures des aérosols
- aide à la construction de projets
- formation sur microphysique des nuages et sur l'instrumentation dédiée

Réalisations

Plus de 15 produits et bancs de métrologie spécifiques : spatial, militaire, automobile, industrie optique

- moyenne de 2 conceptions instrumentales par an
- plusieurs prototypes de logiciel de pilotage et traitement de données



PROMOPLI : prototype de mesure topologie de surface du zinc en phase liquide (galvanisation) :

- conception instrumentale : température, sécurité laser, vitesse bande à 2m/s
- développement logiciel de pilotage et acquisition
- sous-traitance : électronique et mécanique



Produit KaleoMTF : qualification automatique d'objectifs pour automobile (MTF, distorsion, OPD)

- conception instrumentale : sources LED, motorisation
- développement d'algorithmes complexes
- développement prototype logiciel pilotage et acquisition
- sous-traitance : mécanique et design

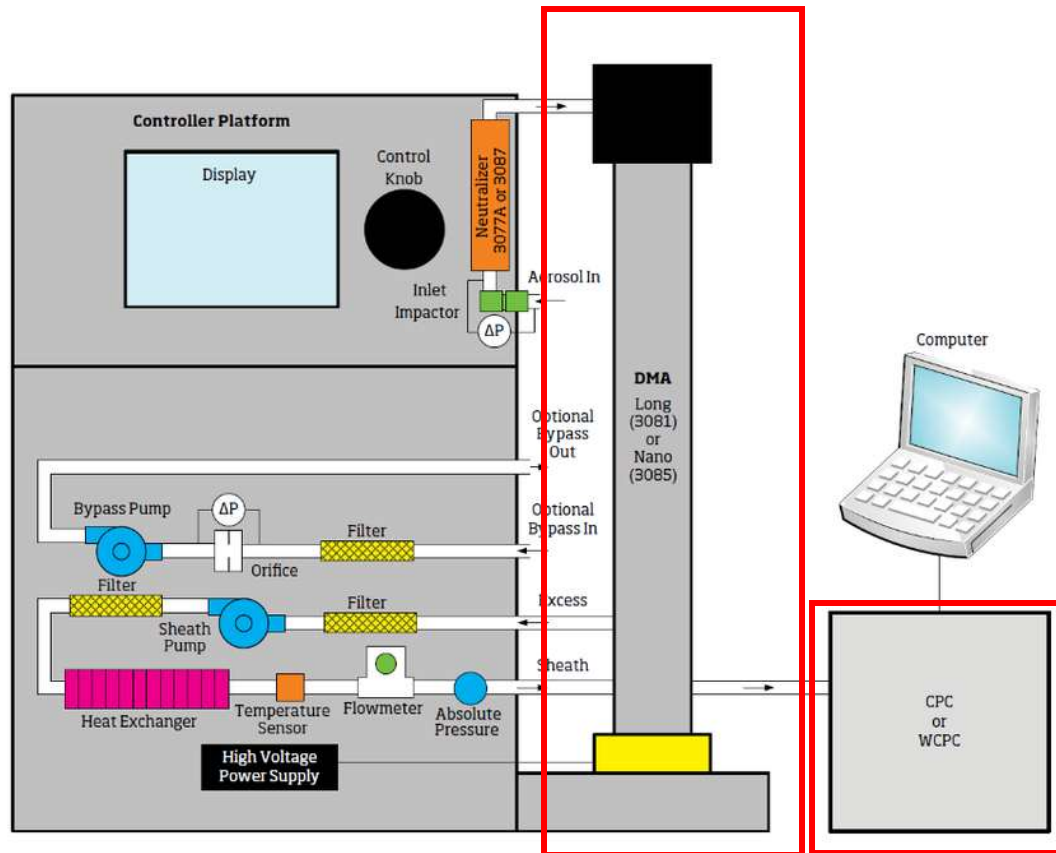
Compétences en Labview

- Instrumentation :
 - pilotage caméra et acquisition images
 - pilotage moteurs scientifiques (Newport, Thorlabs) et industriels
 - acquisitions capteurs (température, règle incrémentale...)
 - gestion ordonnancement / priorités via queue, evenement , rendez vous
- Traitement de données :
 - algorithmie et traitement d'image (module vision)
 - utilisation de bibliothèque de calcul en C /python
- Outils codage :
 - IHM gérée en évènements
 - structure en machine d'état
 - gestion des erreurs
 - transfert de code

Problématiques du moment

- Logiciel de pilotage SMPS :
 - acquisition capteurs
 - contrôle NIDAQ
- Architecture distribuée en labview :
 - utilisation arduino / Raspberry
 - utilisation de BDD (influxDB, postgre ou TDMS)
 - grafana

Principe SMPS



SMPS (Scanning Mobility Particle Sizer) :

Instruments utilisés pour analyser la distribution de taille des particules

Il est composé principalement de :

DMA (Differential Mobility Analyzer)

CPC (Condensation Particle Counter)

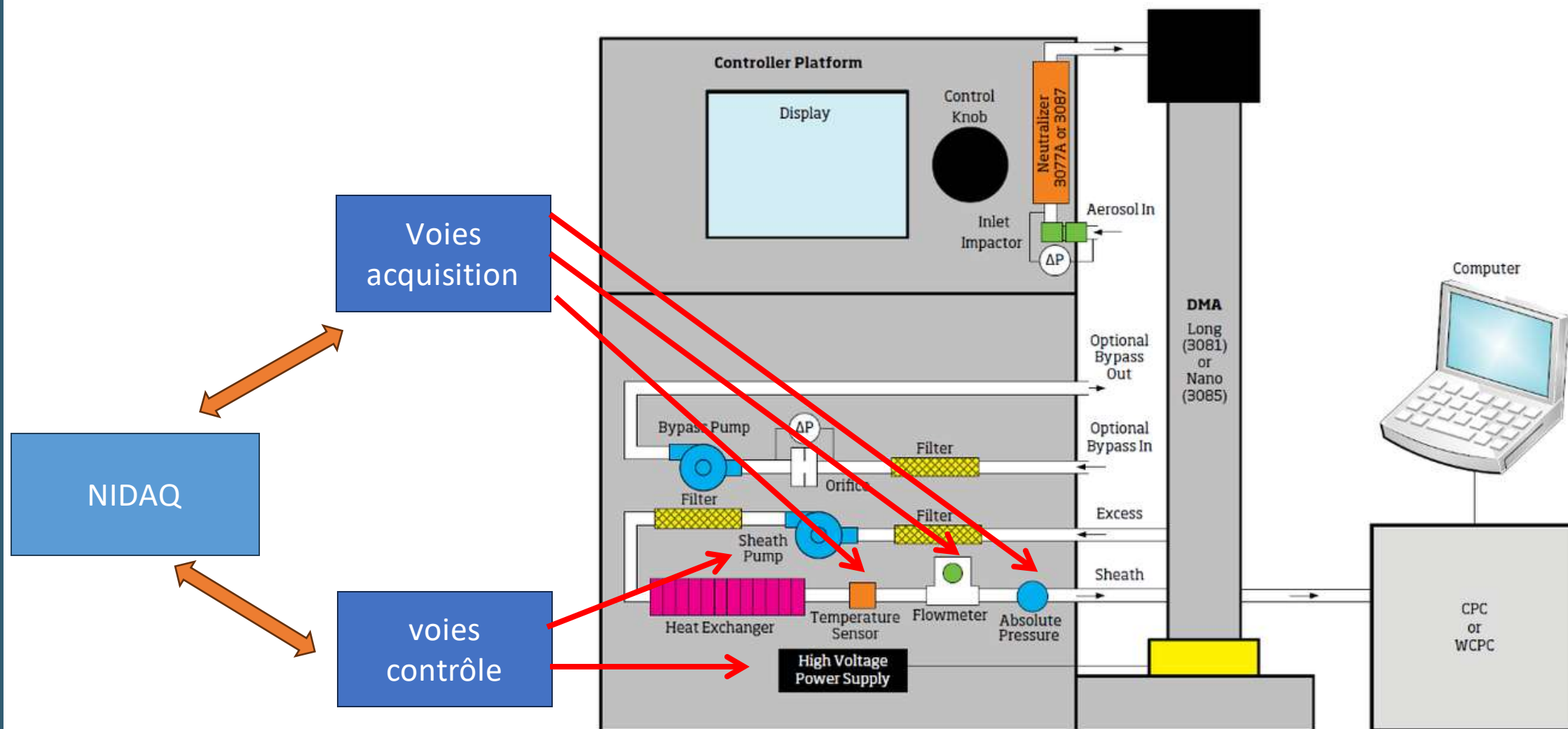
DMA (Differential Mobility Analyzer):

Trie les particules selon leur taille via un champ électrique. Seules celles avec une mobilité donnée atteignent la sortie.

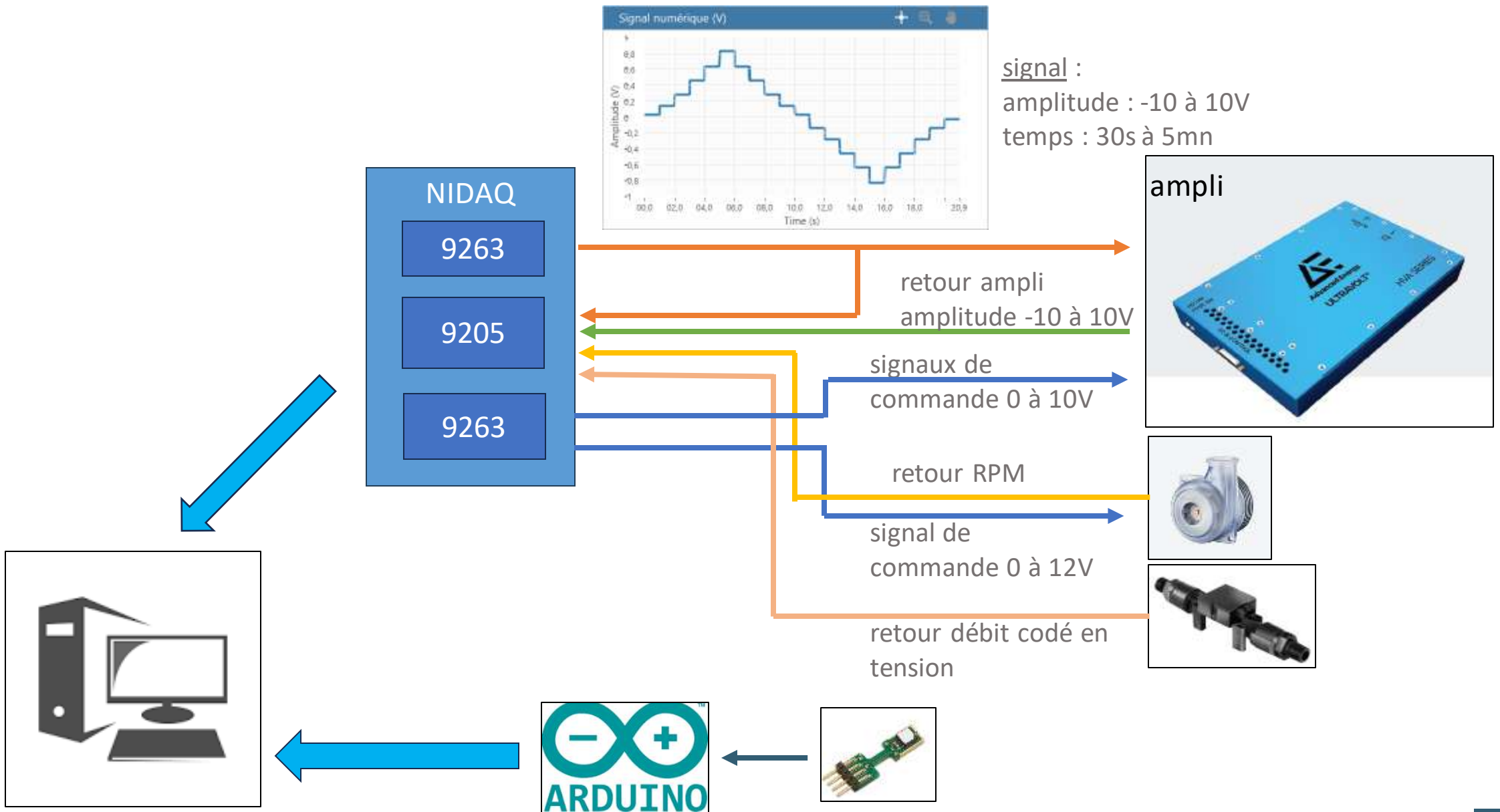
CPC (Condensation Particle Counter):

Fait grossir les particules par condensation.

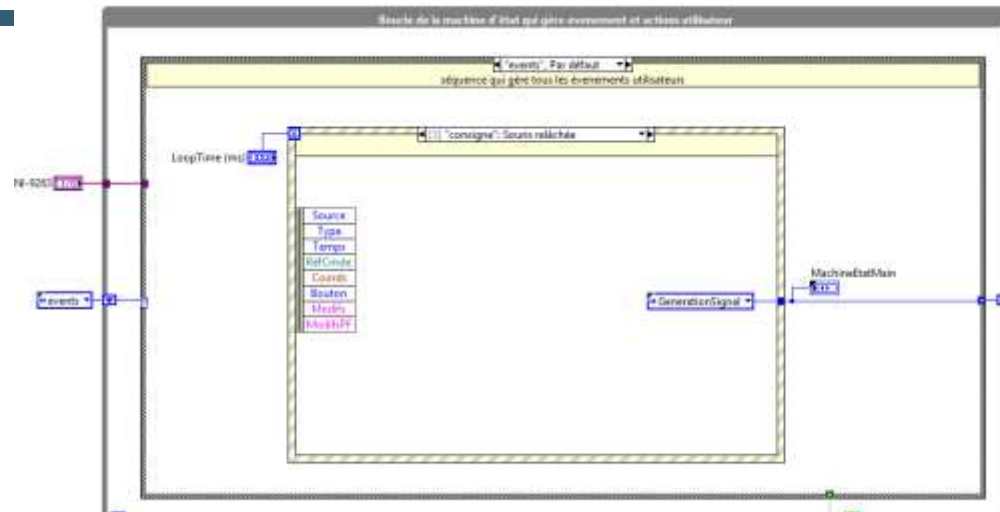
Les détecte et les compte par diffusion lumineuse.



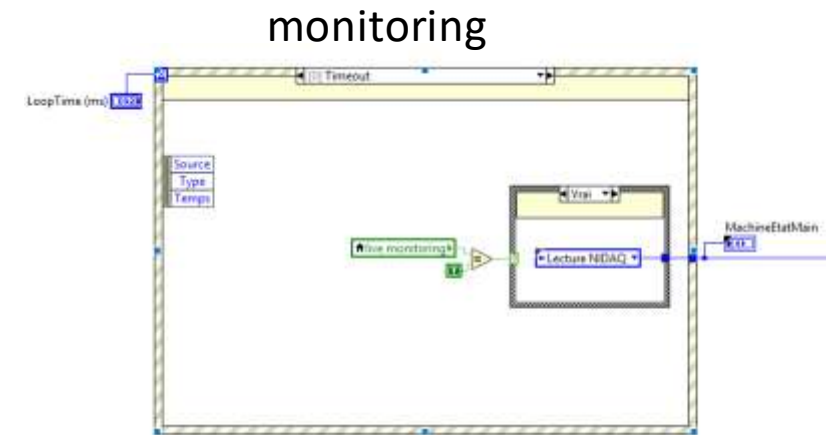
SMPS : synoptique



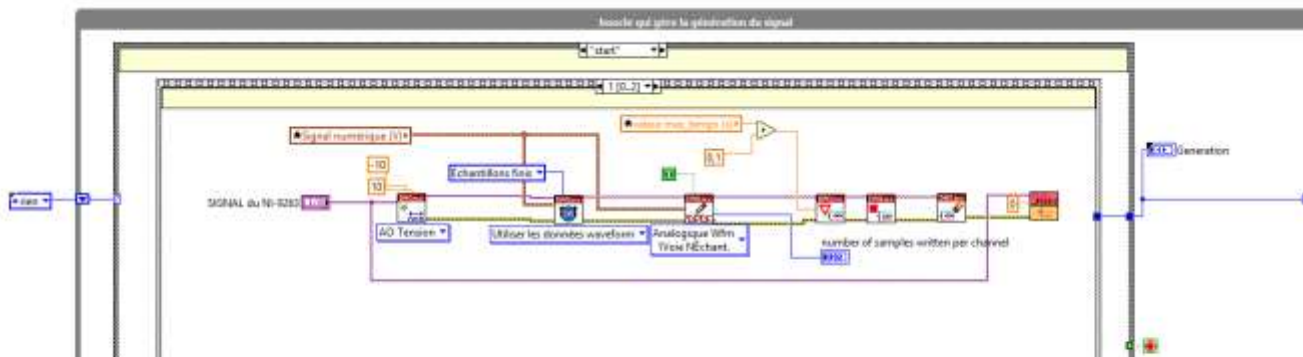
SMPS : test architecture NIDAQ



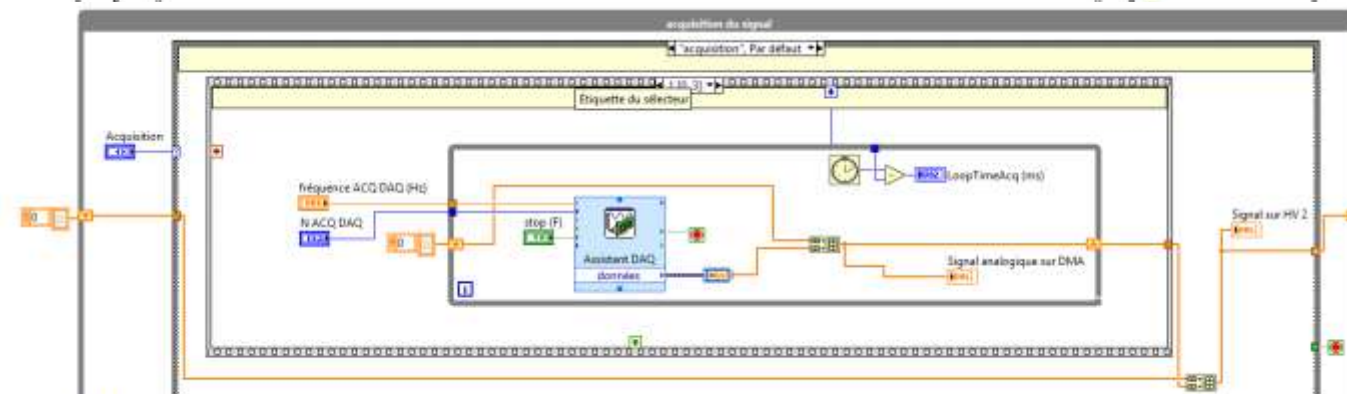
Machine d'état
avec boucle events



monitoring



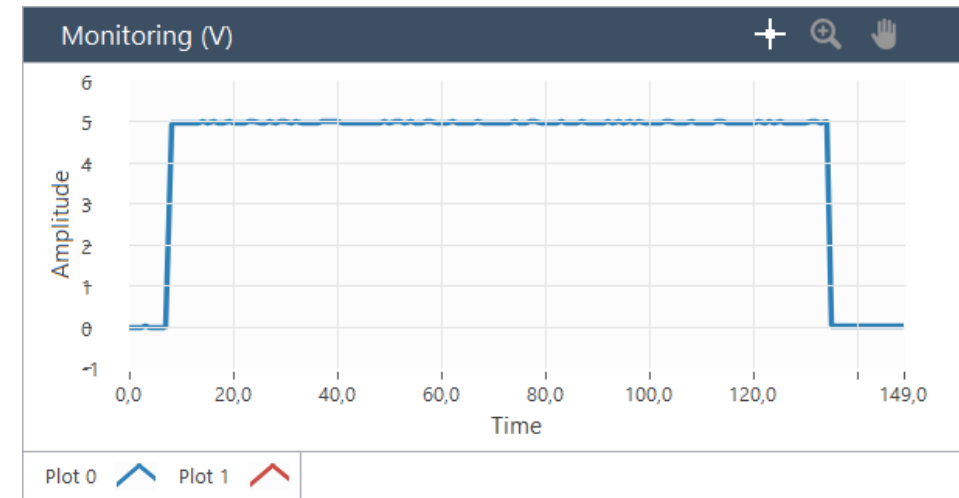
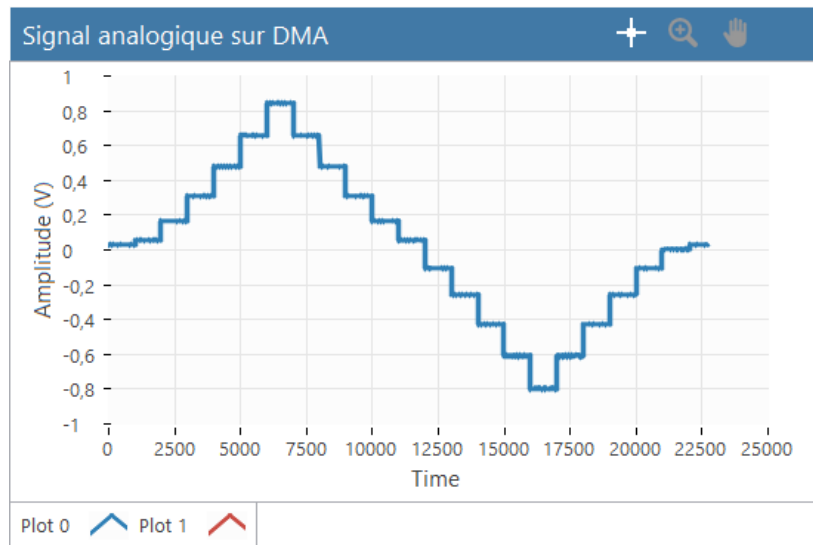
Boucle génération



Boucle acquisition quand état
acquisition est activé

SMPS : test architecture NIDAQ

- Résultats test architecture :
 - système fonctionnel
 - problème : générer et mesurer le signal qui va à l'ampli on doit déconnecter le monitoring sinon erreur NIDAQ



Problématiques actuelles

Contexte :

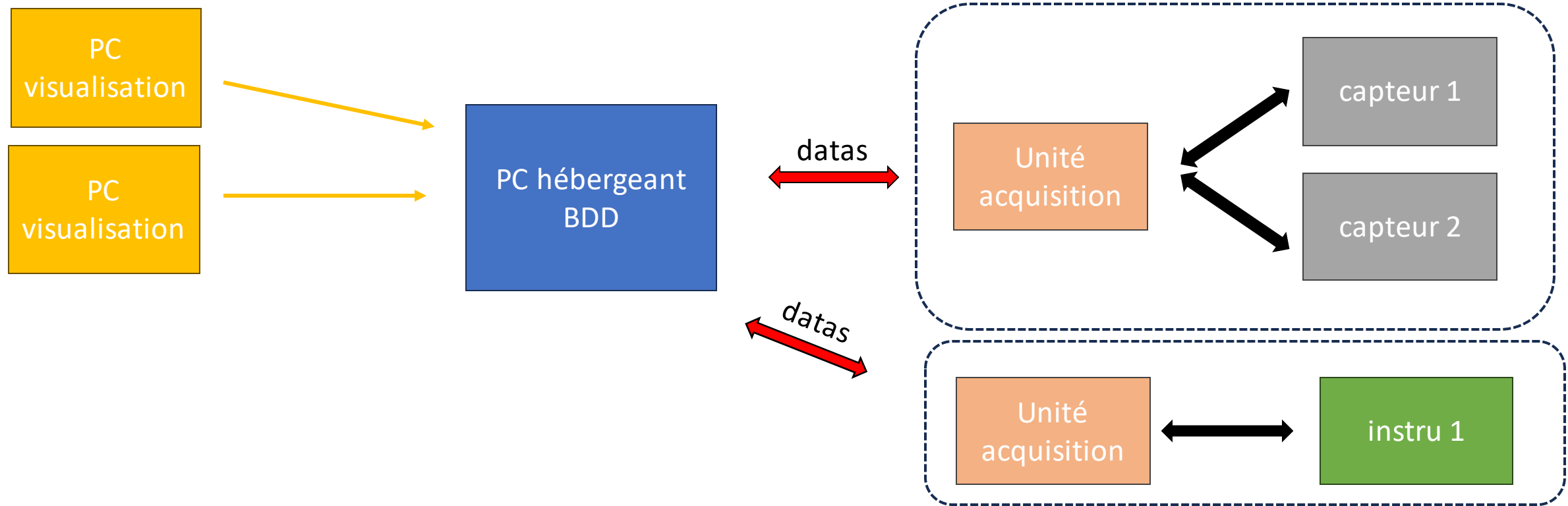
La majorité des développeurs instru au labo utilisent labview

L'architecture la plus utilisée est centralisée ce qui est lourd à maintenir.

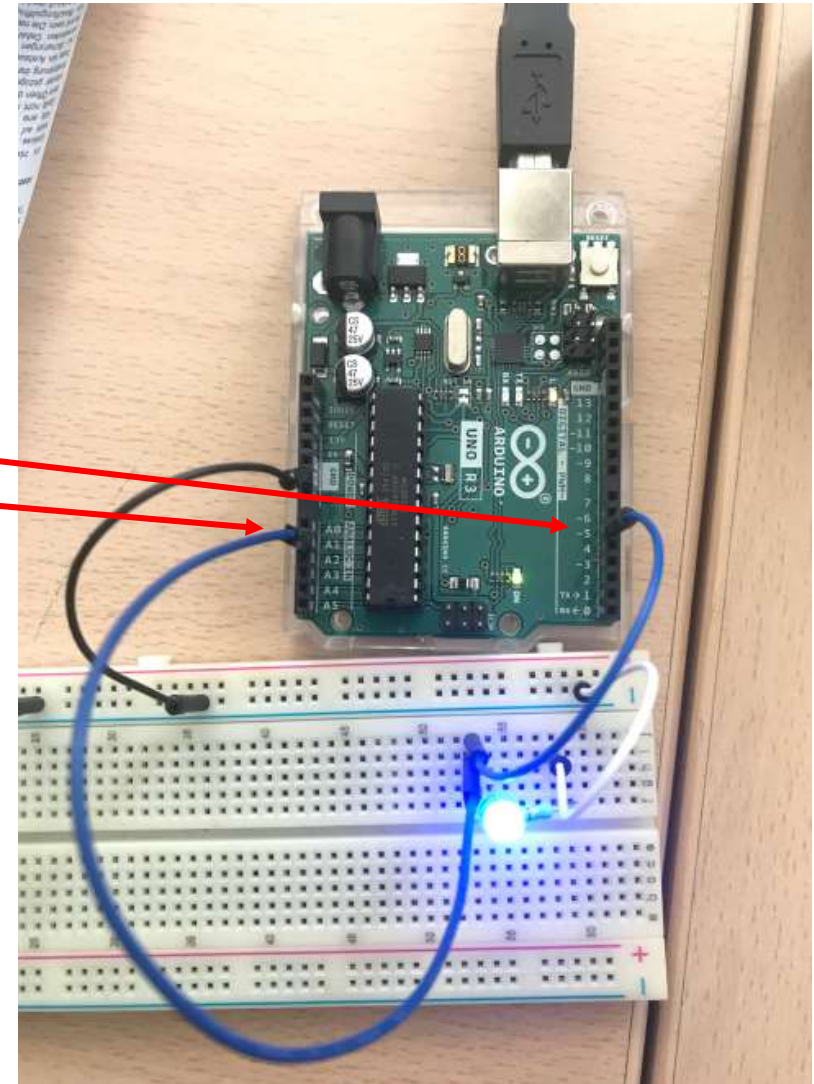
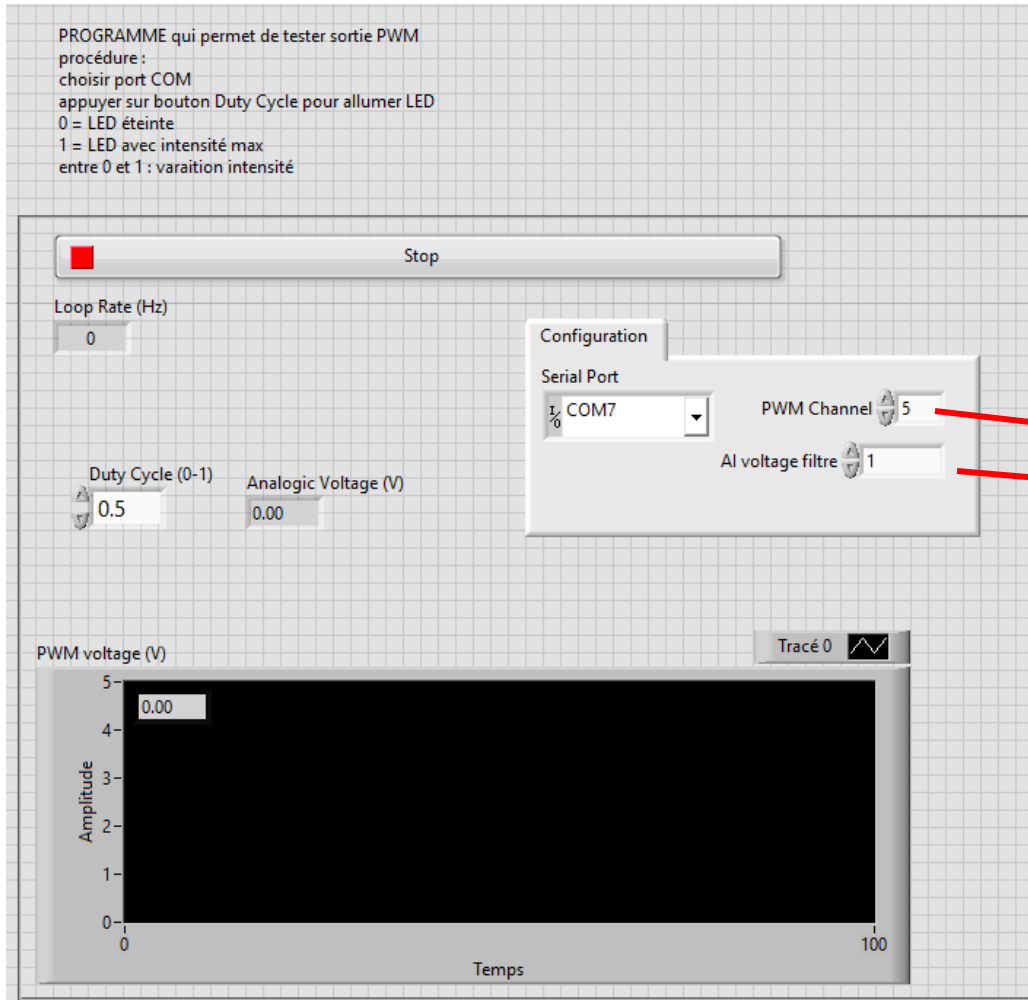
Pour un nouveau projet, on aimerait tester une architecture distribuée plus simple à maintenir

- Architecture distribuée en labview :
 - utilisation arduino / Raspberry
 - utilisation de BDD (influxDB, postgre ou TDMS)
 - grafana

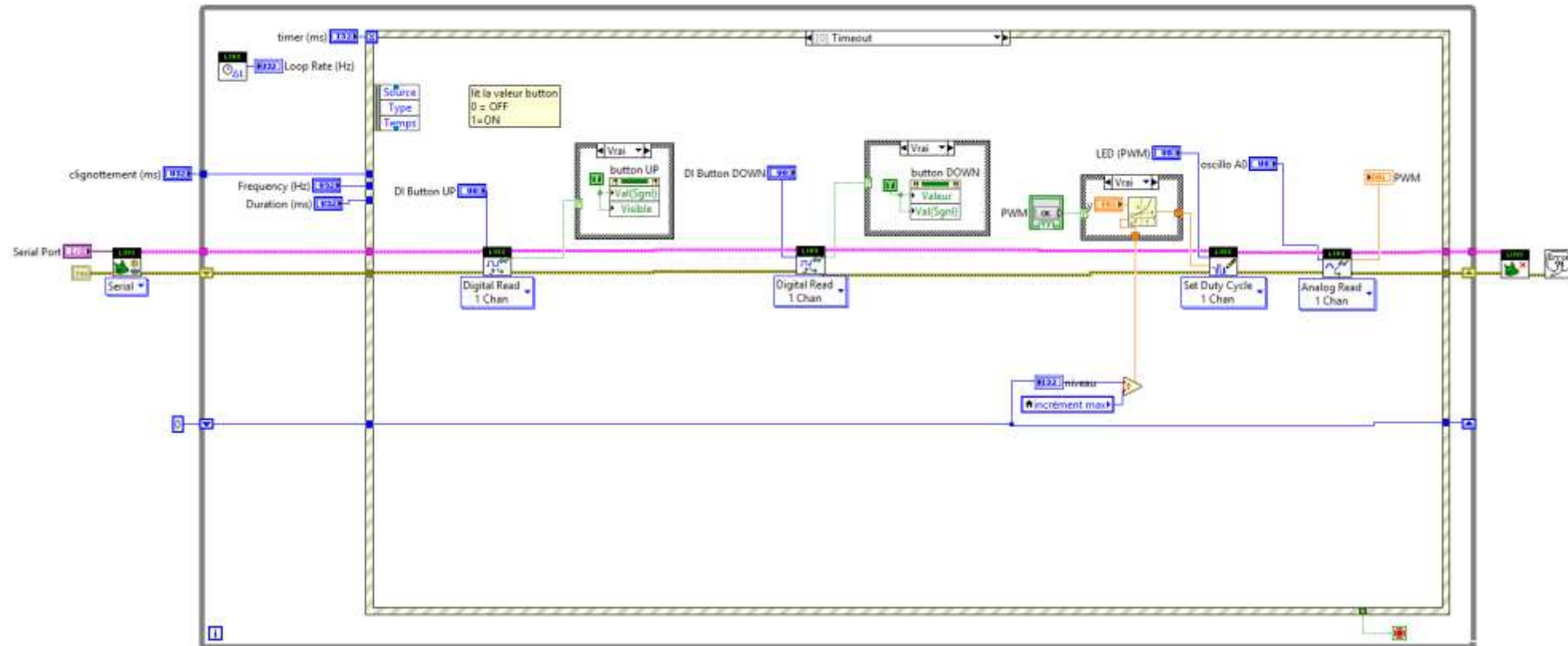
Architecture distribuée en labview ?



LINUX test PWM : face avant



LINX test PWM : diagramme



Architecture distribuée en labview ?

Unité
acquisition



Pour plusieurs types de capteurs ce sera Arduino ou LattePanda.

Avantage LattePanda :

Arduino est déjà présent

PC monocarte avec windows

PC
hébergeant
BDD



Conclusion / perspectives

SMPS :

L'architecture autour d'un NIDAQ est satisfaisante mais des astuces de programmation restent à trouver pour éviter de suspendre le monitoring pendant la mesure.

Architecture distribuée en labview :

Il est possible de réaliser une architecture distribuée en utilisant labview pour langage de programmation. Ceci implique d'utiliser des PC avec windows

Test de la carte d'un executable labview sur carte LattePanda prochainement