



**HAL**  
open science

## PyScada : Logiciel de supervision libre et générique

Benoit Larroque, Franck Luthon, Camille Lavayssiere

► **To cite this version:**

Benoit Larroque, Franck Luthon, Camille Lavayssiere. PyScada : Logiciel de supervision libre et générique. CNRIUT 2023 Saint Pierre - La Réunion, IUT de Saint Pierre, Jun 2023, Saint Pierre (La Réunion), La Réunion. hal-04326702

**HAL Id: hal-04326702**

**<https://hal.science/hal-04326702>**

Submitted on 6 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## PyScada : Logiciel de supervision libre et générique

---

Benoît Larroque<sup>1</sup>

Camille Lavayssière<sup>2</sup>

Franck Luthon<sup>3</sup>

benoit.larroque@univ-pau.fr camille.lavayssiere@univ-pau.fr franck.luthon@univ-pau.fr

<sup>1</sup> IUT de Bayonne Pays Basque, Université de Pau et des Pays de l'Adour, E2S UPPA  
Laboratoire des Sciences de l'Ingénieur SIAME, Anglet, EA 4581

<sup>2</sup> Laboratoire Informatique LIUPPA, Anglet, EA 3000  
Laboratoire des Sciences de l'Ingénieur SIAME, Anglet, EA 4581

<sup>3</sup> IUT de Bayonne Pays Basque, Université de Pau et des Pays de l'Adour, E2S UPPA  
Laboratoire Informatique LIUPPA, Anglet, EA 3000

**THÈMES** – *Informatique - Électronique*

**RÉSUMÉ** – *Cet article présente un logiciel de supervision libre nommé PyScada permettant l'acquisition de données, la supervision et le contrôle distant de systèmes embarqués connectés (internet des objets IoT) en adoptant une approche pluridisciplinaire et interopérable. Cet outil générique, au regard des logiciels existants, prend en compte des notions d'interopérabilité afin d'être utilisés sur des systèmes informatiques abordant divers domaines d'application. PyScada repose sur une infrastructure logicielle dont l'architecture permet de faire évoluer, de modifier et d'adapter l'interface aux besoins de chaque utilisateur. Quelques cas d'application sont présentés et s'articulent autour des systèmes embarqués dans les domaines de l'éducation, du suivi énergétique du bâtiment et des manipulations de laboratoires de recherche .*

**MOTS-CLÉS** – *supervision, open-source, IoT*

# 1 Introduction

Depuis la fin du 20<sup>ème</sup> siècle la présence de l'informatique dans l'enseignement supérieur, plus particulièrement dans les sciences et techniques de l'ingénieur, la recherche et l'industrie a permis aux travailleurs (étudiants, enseignants, chercheurs, techniciens et ingénieurs) d'avoir des outils d'acquisition de données (mesure, alarme, retour d'état de fonctionnement) et de paramétrage des processus (contrôle, configuration). Les logiciels ont été dans un premier temps conçus pour un type d'application, ils sont parfois dédiés à un domaine particulier et ne peuvent représenter qu'une partie d'une chaîne d'acquisition, le traitement des données étant parfois réalisé ultérieurement par l'homme. Afin de définir les lignes directrices de logiciels génériques dans le domaine de la supervision, de l'instrumentation et de l'expérimentation, il est nécessaire de définir de manière abstraite les besoins d'un utilisateur.

Afin de présenter une application capable de gérer les différents niveaux d'interopérabilité le logiciel **PyScada** développé à l'origine par Martin Schröder de l'Université Technique de Berlin, est utilisé. PyScada se définit comme : *Un système SCADA open source avec une IHM HTML5, construit à l'aide du framework Django.*

PyScada se présente donc sous la forme d'une page web utilisant les standards du web accessible sur tous les dispositifs récents. Le code source est disponible sur *GitHub*<sup>1</sup> et la documentation sur *ReadTheDocs*<sup>2</sup>.

## 2 Logiciel PyScada

### 2.1 Présentation

PyScada est un logiciel *open source* et libre depuis sa création, sous licence GPL-3.0 et fonctionne sur toute plateforme UNIX et sur Windows à l'aide de conteneurs Docker. L'interopérabilité concernant les systèmes d'exploitations est ainsi assurée. Le dépôt officiel de PyScada contient le code source et les méta-données du logiciel. PyScada est également disponible sur le Python Package Index (PyPI)<sup>3</sup>. Un canal de discussion est disponible sur Matrix et les questions ou problèmes sont traités au fur et à mesure qu'ils se présentent. PyScada respecte donc l'ensemble des recommandations énoncées par [1].

L'architecture de PyScada est développée autour de Django, un *framework* Python, et vise à être simple à déployer. Django fournit deux interfaces web lui permettant de configurer et d'interagir avec des instruments utilisant l'un des protocoles suivants : Modbus, BACNet, VISA, OneWire, SMBus/I2C, GPIO, PT104, OPC-UA, SML, Serial, Webservice. L'ajout de nouveaux protocoles est facile grâce à l'architecture modulaire de PyScada. Pour générer les pages web, PyScada utilise le serveur web NGINX et la

passerelle Python WSGI Gunicorn.

Le matériel et les variables sont configurés dans PyScada par une couche abstraite et une couche spécifique du protocole utilisé. Cette interopérabilité protocolaire et matérielle de PyScada permet de traiter toutes les variables de manière uniforme en lecture ou en écriture. Ces données sont utilisées pour créer des événements, des alertes ou des scripts afin d'exécuter des algorithmes, d'interagir avec l'utilisateur et d'automatiser des processus.

### 2.2 Django

Django se définit lui-même comme un cadre web Python de haut niveau qui encourage le développement rapide et une conception propre et pragmatique. Il est gratuit, *open source*, rapide, sécurisé et évolutif. Sa grande communauté, ses mises à jour régulières et ses applications disponibles permettent de créer simplement une application web.

Les données et la configuration sont stockées dans une base de données. Django prend en charge une multitude de type de base de données : PostgreSQL, MySQL, MariaDB, Oracle, SQLite.

Django utilise trois couches d'abstractions : le modèle, la vue et le gabarit (MVT de l'anglais *Model-View-Template*) :

- Le **modèle** est utilisé pour structurer et manipuler les données.
- La **vue** traite les demandes des utilisateurs (via un distributeur d'URL) et renvoie une réponse après avoir organisé les données à l'aide de gabarits.
- Les **gabarits** sont utilisés pour générer des pages web HTML de manière dynamique.

L'architecture MVT, séparant la logique de création de site web en trois parties, rend facilement évolutif le site web, les développeurs pouvant travailler simultanément sur les différentes parties de l'application. Un modèle peut avoir différentes représentations (vues) et leur maintenance est facilitée. A l'aide de cette architecture, PyScada peut répondre aux besoins de supervision et d'instrumentation pour des applications variées via l'abstraction de l'expérimentation.

Les modules de PyScada sont définis dans l'application par défaut ou ajoutés via un *plugin*. Ils peuvent communiquer avec des instruments, des capteurs, des actionneurs ou le système hôte en implémentant des bibliothèques *open source* utilisant un protocole de communication. La création de l'interface utilisateur est générée via le module d'IHM de PyScada et par l'intermédiaire de Gunicorn et NGINX. Un *plugin* peut proposer de nouveaux composants en ajoutant de nouvelles **vues** et de nouveaux **gabarits**, comme par exemple un thème pour l'interface utilisateur ou des types de graphiques à l'aide des bibliothèques.

1. <https://github.com/pyscada/PyScada>

2. <http://pyscada.rtfid.io>

3. <https://pypi.org/project/PyScada/>

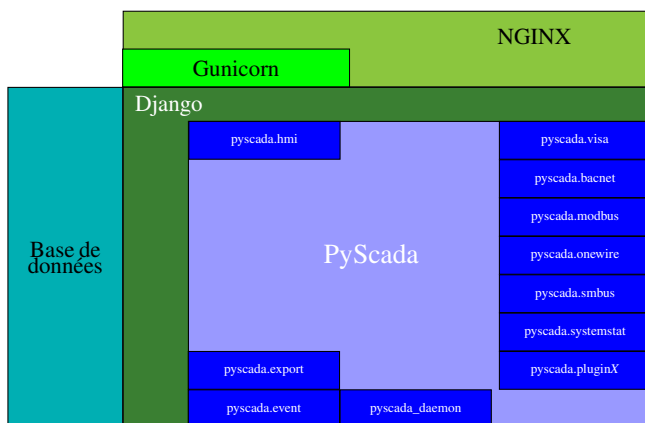


FIGURE 1 – Schéma de la structure de PyScada dans l’environnement Django.

### 3 Exemples d’application

#### 3.1 LaboRem : Travaux pratiques à distance

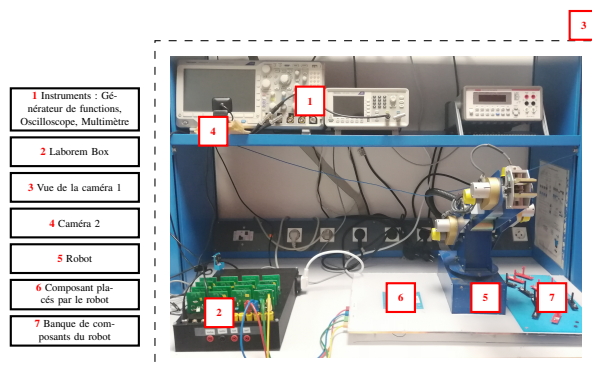


FIGURE 2 – Vue d’ensemble du banc d’essai Laborem.

PyScada est utilisé pour gérer l’interaction entre l’apprenant et une plate-forme de travaux pratiques à distance permettant à l’étudiant de réaliser des travaux d’électronique à distance. Les instruments de mesures connectés à la Laborem-box [2], le robot et le système de vision fournissent les outils nécessaires à la réalisation de ce type de manipulation.

#### 3.2 Suivi énergétique de bâtiment

Les différents objets connectés dans le bâtiment tels que des compteurs énergétiques ou une station météo sont connectés et archivés sur la box via PyScada. Un accès temps réel aux données où un accès aux archives fournissent des données exploitables pour optimiser les consommations énergétiques dans le bâtiment. Déployé sur le bâtiment de l’IUT de Bayonne le plan de comptage du systèmes permet d’accéder à 200 variables.

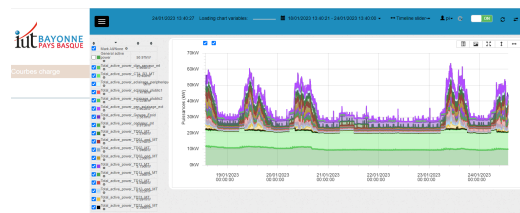


FIGURE 3 – Courbes de charge de l’IUT de Bayonne.

#### 3.3 Expérimentation in-situ - digue de l’Artha

Une réseau de 16 capteurs de pression et un système de vision ont été implémenté sur la digue de l’Artha dans la baie de Saint-Jean de Luz. PyScada contribue à l’étude de l’impact des vagues sur une structure de défense côtière en alliant une captation vidéo avec une prise de mesure des pressions d’impact.



FIGURE 4 – Vue de la digue de l’Artha.

### 4 Conclusion

PyScada est un logiciel organisé autour d’une architecture permettant de présenter l’ensemble d’une expérimentation distante dans une interface web accessible depuis tout terminal informatique récent. L’approche abstraite autorise l’échange d’informations avec des capteurs et actionneurs de systèmes physiques quel que soit le domaine d’application. La possibilité d’implémenter des scripts pour traiter des données en leur donnant du sens et de la valeur. PyScada permet de s’adapter aux différents types d’utilisateurs en proposant différents niveaux techniques d’accès à une instrumentation.

### Références

- [1] Jiménez RC, Kuzak M, Alhamdoosh M et al., *Four simple recommendations to encourage best practices in research software*, F1000Research 2017, 6 :876
- [2] C. Lavayssière, B. Larroque, F. Luthon, *Laborem Box : A scalable and open source platform to design remote lab experiments in electronics*, HardwareX, Volume 11, 2022, e00301, ISSN 2468-0672