

Projets de programmation Python LUMI

1. **Interface de récupération de données du site ParisData :**

Créer une classe qui permette de récupérer les données mises à jour en temps réel sur le site ParisData (par exemple le nombre de cyclistes qui passent place de la Nation toutes les 10 minutes). Puis les tracer sur un graphique qui devra être mis à jour automatiquement. Pour aller plus loin on pourra faire une interface graphique qui permet de récupérer d'autres données (que l'on choisira avec un menu déroulant), et éventuellement de traiter un peu les données (écart type, moyenne, etc.).

2. **Créer un oscilloscope et un générateur de fonctions avec une carte ADUC7020 :**

(Carte fournie comportant un CPU, un ADC (Analog to Digital Converter) et un DAC (Digital to Analog Converter)) pour réaliser un oscilloscope et un générateur de signaux arbitraires commandés par l'ordinateur. Un programme en langage C sera déjà intégré à la carte ADUC et permettra de récupérer les données de l'ADC et d'écrire des données sur le DAC via le port série. On réalisera une interface graphique traçant les données de l'ADC en temps réel (réalisant un oscilloscope) et permettant de sortir n'importe quel signal sur le DAC (générateur de signaux). Pour aller plus loin on pourra programmer un analyseur de réseaux et l'utiliser pour mesurer la fonction de transfert d'un circuit RLC.

3. **Ecrire une classe qui permette d'interfacer un oscilloscope et un générateur de signaux arbitraires :**

Dans le même esprit que dans le TP instrument, interfacer un oscilloscope (qui sera fourni) via le port USB. L'utiliser pour mesurer le bruit de phase d'un générateur de signaux. On pourra également interfacer le générateur de signaux et utiliser ces deux appareils pour tracer la fonction de transfert d'un circuit électronique élémentaire.

4. **Interfacer la carte son d'un PC :** écrire une classe qui permette de récupérer un son sur le microphone. (On pourra à titre de test acquérir le son d'un diapason et en déduire son facteur de qualité mécanique.) Faire une interface qui trace en direct sa densité spectrale de puissance. Faire une application graphique qui donne la fréquence d'un son en temps réel (un accordeur pour instrument de musique). Mesurer la vitesse du son (nécessite de pouvoir envoyer un son et le lire en même temps). Faire des mesures continue pendant plusieurs heures (toutes les minutes). Observer la dérive. Est-il possible de mesurer les variations de température ? Tracer la variance d'Allan du signal.

5. **Traceur de rayons :** repartir de la classe développée en TD. Ecrire une fonction qui convertit un fichier .ZMX (disponible pour les lentilles de la marque Thorlabs) en un objet Python. Modifier le code pour gérer la dépendance de l'indice optique en fonction de la longueur d'onde. Ajouter une méthode qui calcule la PSF (point spread function) géométrique en un point de l'axe optique. Pour aller plus loin : faire une interface graphique pour cette classe (avec des boutons pour ajouter des dioptries, tracer les rayons, possibilité d'entrer le nombre de rayons souhaités, ...). Pour aller encore plus loin : modifier l'interface graphique pour offrir la possibilité de déplacer les lentilles dynamiquement avec la souris (et recalculer automatiquement les rayons).

6. **Programmer un synthétiseur de sons** : écrire une classe qui permette d'interfacer la carte son d'un PC. Générer des sons purs, puis éventuellement des sons plus élaborés. Puis faire une interface graphique avec un clavier de piano qui permette de jouer les différents sons en cliquant sur les touches. Un sélecteur permettra de sélectionner le timbre du son. On pourra ajouter un contrôle de volume, et un égaliseur.

7. **Interfacer la webcam d'un ordinateur** : Créer une interface qui permette d'acquérir une image de la webcam d'un ordinateur en cliquant sur un bouton. On ajoutera un graphique avec des barres, générées à chaque acquisition d'image, qui montre la quantité de rouge, vert et bleu, ainsi que la luminosité de l'image. Modifier le programme pour acquérir et afficher un flux vidéo provenant de la webcam et mettre à jour en temps réel l'information colorimétrique. Utiliser votre programme pour comparer du point de vue colorimétrique une lampe à LED, une lampe halogène et une lampe économique à décharge, un ciel bleu et un ciel nuageux. Autre idée : tenter d'estimer la température d'un corps noir en comparant la couleur mesurée à la webcam, à la couleur attendue d'un corps noir.