



INTERFAÇAGE NUMÉRIQUE

Travaux Pratiques

Semestre 6

Banc de mesure de rayonnement

4 séances

Ce sujet est disponible au format électronique sur le site du LEnsE - <https://lense.institutoptique.fr/> dans la rubrique Année / Première Année / Interfaçage Numérique S6 / Banc de mesure de rayonnement lumineux.

Banc de mesure de rayonnement

À l'issue des séances de TP concernant le **bloc d'acquisition d'un diagramme de rayonnement**, les étudiant·es seront capables de :

- Développer et mettre en œuvre une **solution d'électronique embarquée** pour **acquérir des données analogiques** et commander un élément mobile.
- Mettre en œuvre un **protocole simple de communication** entre un ordinateur et un microcontrôleur pour transmettre des commandes et lire des données
- Optimiser une **interface informatique** de pilotage et d'affichage de données

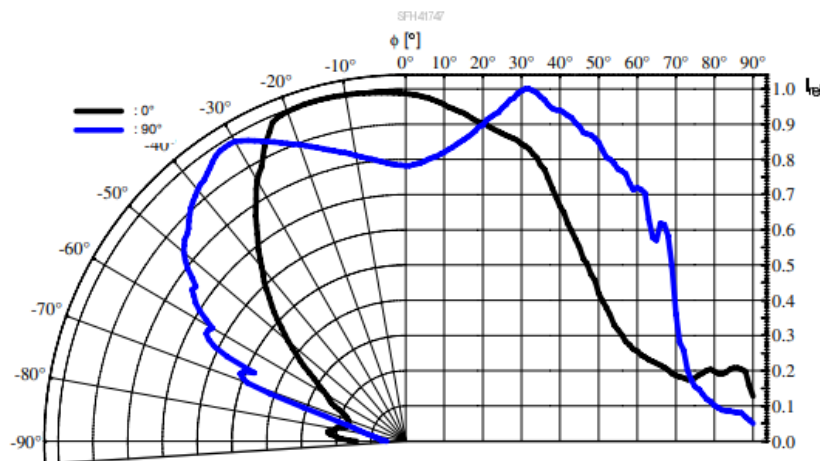
Objectifs du mini-projet

L'objectif principal de ce mini-projet est d'**automatiser un banc de mesure de rayonnement lumineux** à l'aide d'un ordinateur et d'une interface en Python. Le matériel est piloté par une carte à microcontrôleur à laquelle il faudra envoyer des commandes selon un protocole série.

Un diagramme de rayonnement lumineux est la **représentation graphique** de la **distribution angulaire** d'une grandeur caractérisant le rayonnement d'une source lumineuse.

Radiation Characteristics 7), 8)

$$I_{e,rel} = f(\varphi)$$



Exemple de diagramme de rayonnement d'une LED OSRAM SFH 41747 (documentation technique)

Vous aurez à votre disposition une **maquette** permettant la mise en mouvement d'un système de photo-détection autour d'une source de puissance à LED. Cette maquette est pilotée par une carte Nucléo (contenant un microcontrôleur).

Déroulement du bloc

La liste des étapes à suivre pour la réalisation du programme embarqué de la plateforme de rayonnement lumineux est donnée à titre indicatif. L'ordre et le choix des différentes étapes sont laissés à l'appréciation des différents binômes.

Afin de faciliter la réutilisation des codes, il pourra être intéressant de définir des fonctions pour le pilotage des différents éléments.

Séance 1 / Arduino et Nucléo-STM32 (sans maquette !!)

Le sujet de cette séance est fourni dans un document annexe, disponible aussi sur le site du LEnsE - <https://lense.institutoptique.fr/> dans la rubrique Année / Première Année / Interfaçage Numérique S6 / Bloc Systèmes embarqués / Intro Arduino et STM32.

Etape 0 - 30 min Installer les drivers STM32 et tester un premier programme

Etape 1 - 45 min Piloter des sorties numériques - LED

Etape 2 - 45 min Acquérir des données numériques - Bouton-poussoirs

Etape 3 - 45 min Mettre en œuvre des interruptions sur des événements externes

Etape 4 - 45 min Utiliser des sorties modulées en largeur d'impulsion (PWM) - LEDs

Etape 5 - 60 min Acquérir des données analogiques - Potentiomètre

Séance 2 / Prise en main de la maquette et automatisation de la mesure

Etape 6 - 60 min Piloter l'intensité des LEDs de la maquette

Etape 7 - 60 min Piloter le servomoteur de la maquette

Etape 8 - 90 min Définir et tester une première structure de code permettant de piloter le servomoteur en faisant une acquisition de la luminosité à pas régulier

Etape 9 - 60 min Récupérer les données à l'aide du moniteur Série du logiciel Arduino

Séance 3 / Mise en place d'un protocole de communication

Etape 10 - 90 min Mettre en place un protocole de communication basé sur une liste de commandes et intégrer à la structure du code embarqué

Etape 11 - 90 min Utiliser la bibliothèque pySerial en Python pour envoyer les commandes à la carte Arduino

Etape 12 - 90 min Tester la communication entre la carte Arduino et le script Python pour afficher les données

Séance 4 / Application complète

Cette dernière séance sera consacrée à la finalisation des différents programmes : embarqué sur la carte Arduino pour la mesure automatique et sur l'ordinateur pour le pilotage de la maquette et l'affichage des données.

Selon le temps restant, il sera possible d'intégrer les fonctions écrites en Python dans une interface graphique, dont la structure de base est fournie.

Séance 2 / Prise en main de la maquette et automatisation de la mesure

Objectifs de la séance

Cette seconde séance est consacrée à la **prise en main de la maquette** et à l'automatisation de la mesure de l'intensité lumineuse d'une source à LED (par exemple).

Description de la maquette

Éléments constitutifs

Alimentation électrique

La tension maximale admissible par le servomoteur est de 6 V !
Le courant maximal admissible par la LED de puissance est de 200 mA !

Brochage

Maquette	Broche Nucléo	Type	Description
LED1	PC7	Sortie / PWM	Led active à l'état haut
LED2	PB13	Sortie / PWM	Led active à l'état bas
SW1	PC6	Entrée	Bouton-poussoir, par défaut état bas
SW2	PC8	Entrée	Bouton-poussoir, par défaut état bas
USERBUTTON	PC13	Entrée	Bouton-poussoir, par défaut état haut
SERVO	PB7	Sortie / PWM	Servomoteur - T = 20 ms
PHOTODIODE	PC3	Entrée analogique	Photodiode (montage simple avec R variable)
RÉSISTANCE THERMIQUE	PC1	Entrée analogique	Résistance Thermique CT10k
SPI	PA5 PA6 PA7	Sortie	Signal d'horloge
SCK		Entrée	Données entrantes
MISO		Sortie	Données sortantes
MOSI			
POT NUM	SPI	MCP4132	Potentiomètre Numérique Monté dans un pont diviseur
CS	PB9	Sortie	Sortie du pont diviseur
Pot Num In	PB0	Entrée analogique	
LED PUISSANCE	SPI	MCP4132	Potentiomètre numérique / Contrôle courant dans Driver BCR430-UW6
CS	PB5	Sortie	

Utilisation de la sortie modulée PB7

```
1 LL_GPIO_SetAFPin_0_7(GPIOB, GPIO_PIN_7, GPIO_AF1_TIM2);
```

Etape 6 / Piloter l'intensité des LEDs de la maquette

Etape 7 / Piloter le servomoteur de la maquette

Etape 8 / Définir et tester une première structure de code

Etape 9 / Récupérer les données à l'aide du moniteur Série du logiciel Arduino

Séance 3 / Mise en place d'un protocole de communication

Objectifs de la séance

Cette séance a pour but de mettre en place un **protocole de communication** entre la carte embarquée et l'ordinateur, d'abord à l'aide d'un moniteur série puis à l'aide d'un script en Python.

L'ordinateur, maître de la communication, devra envoyer des commandes à la carte embarquée, qui devra les exécuter puis acquitter du bon déroulement de l'opération.

Un exemple de code pour la transmission des commandes A et D, à analyser et à tester, est fourni.

Etape 10 / Mettre en place un protocole de communication basé sur une liste de commandes

Code de base avec les commandes A et D

Etape 11 / Utiliser la bibliothèque pySerial en Python pour envoyer les commandes à la carte Arduino

Code de base pySerial pour envoyer les commandes A et D et récupérer des données.

Etape 12 / Tester la communication entre la carte Arduino et le script Python pour afficher les données

A faire : mettre en place la structure de base.

Liste des commandes - côté maître

Commande PC	Description	Réponse Nucléo
!T ?	Test de la transmission	!T ;
!A :VALUE ?	Transmission de l'angle de départ souhaité pour le diagramme valeur entière en degré <i>Si angle non compris entre -90 et 90</i>	!A :VALUE ; VALUE= -100
!B :VALUE ?	Transmission de l'angle final souhaité pour le diagramme valeur entière en degré <i>Si angle non compris entre -90 et 90</i>	!B :VALUE ; VALUE= -100
!C :VALUE ?	Transmission du pas angulaire souhaité pour le diagramme valeur entière en degré <i>Si angle non compris entre -90 et 90</i>	!C :VALUE ; VALUE= -100
!S ?	Lancement de l'acquisition nb est le nombre d'échantillons qui seront acquis par la carte <i>Si les angles fournis sont non compatibles</i>	!S :NB ; NB= 0
!E ?	Test de fin de l'acquisition (Y)es or (N)o	!E :Y/N ;
!D :INDEX ?	Demande de récupération d'une donnée index correspond au numéro de l'échantillon souhaité value correspond à la valeur de l'échantillon souhaité	!D :INDEX :VALUE ;

INTERFAÇAGE NUMÉRIQUE

Travaux Pratiques

Semestre 6

Ressources

Bloc Rayonnement

Liste des ressources

- [Schéma de la carte de la maquette de rayonnement](#)
- [PCB de la carte de la maquette de rayonnement](#)

