

# TRAVAIL PRATIQUE SYNTHÈSE – PROJET

Cours 420-4B5-MO – Objets connectés

## CONSIGNES ET REMISE

- **Date de remise du travail : Vendredi 19 mai 2023 avant 23h**
- Utilisation de GitHub Classroom et du dépôt créé.
- Remettre dans un fichier compressé (ZIP) nommé *TPSyntheseVotreNom.ZIP* avec les éléments suivants :
  - Un rapport de montage (voir document dans Moodle à la semaine 5).
  - Votre montage complété (photo ou vidéo).
  - Dossier contenant votre projet MicroPython et ses fichiers.
- Le travail pratique doit être fait **INDIVIDUELLEMENT**
- **OBLIGATOIRE POUR LA CORRECTION DU TRAVAIL : L'exécution du résultat de votre projet durant la période de laboratoire à la semaine 14 ou 15.**

## PROJET DE VOTRE CHOIX AVEC LE RASPBERRY PICO ET MICROPYTHON

Dans ce travail pratique, vous devrez programmer un projet MicroPython/Python de votre choix avec le Raspberry Pico et les capteurs proposés dans les spécifications de l'énoncé dans une interface graphique Tkinter. Votre projet devra aussi avoir une structure POO et vous devrez utiliser une base de données locale pour le stockage d'éléments de vos capteurs.

### 1. RAPPORT DE MONTAGE POUR VOTRE PROJET RASPBERRY PICO

Vous devez créer un rapport de montage comprenant le schéma, une description détaillée et une photo du projet terminé. Un canevas, en format de document Word, est disponible à la semaine 5 sur Moodle.

### 2. VALIDATIONS DU TRAVAIL PRATIQUE : SEMAINE 14-15 (OBLIGATOIRE)

Présentation du montage GPIO des capteurs et de son exécution au professeur **en classe ou sur rendez-vous au collège à mon bureau.**

Groupe	Description	Date
Groupe 01	Montage GPIO : capteurs	Avant jeudi 18 mai:
Groupe 02	Exécution de l'atelier avec les GPIO	<ul style="list-style-type: none"><li>• <b>En laboratoire : avant 17h</b></li></ul>
<b>Validation</b>		
Exécution complète de votre interface et de votre montage avec les capteurs. Collecte de données et sauvegarde dans la base de données.		

### 3. CONFIGURATION DE GIT ET GITHUB CLASSROOM

Vous devez utiliser Git et GitHub Classroom pour le travail. Dès le début, assurez-vous d'utiliser le dépôt à distant (référentiel) créé dans GitHub Classroom comme étant votre *master*. Voici l'assignation :

Assignations	Description	Lien
travailpratiquesynthese	Dépôt du travail pratique synthèse	<a href="https://classroom.github.com/a/M473hDsJ">https://classroom.github.com/a/M473hDsJ</a>

**Assurez-vous de découper votre programme en plusieurs parties. Vous devez faire au moins un commit pour chaque partie de votre projet.** Lors de vos *commit* sur votre projet, ajoutez une description suivante :

**FAIT :** < une description d'une tâche nouvellement terminée et/ou d'un problème résolu >

## TRAVAIL À FAIRE (SPÉCIFICATIONS) :

Pour votre application, **vous devez obligatoirement respecter les critères suivants** :

- Avoir une classe ou un type personnalisé de votre choix (par exemple : création d'un objet `mesure`).
- Utilisation de votre classe ou type personnalisé
- Utilisation d'une collection pour vos données (Liste, dictionnaire ou autre)
- Utilisation **d'au minimum, trois capteurs dans les suivants, dont deux capteurs différents** : bouton-poussoir, potentiomètre, détection de mouvement PIR, distance ultrason, inclinaison, température/humidité, thermistance, photorésistance, clavier numérique, module RFID, moteur, servomoteur, joystick, "buzzer", autre (à valider avec le professeur)
- Utilisation **d'au minimum deux lumières LED ou d'une lumière RVB**.
- Utilisation **d'au minimum, d'un capteur d'affichage** : écran LCD, écran 7-segments, barre graphique LED.
- Création d'une interface utilisateur avec un titre
- Votre interface utilisateur doit respecter le modèle POO (interface utilisateur en objet) pour le codage
- Dans l'interface utilisateur, vous devez utiliser, au minimum, les widgets suivants : boutons, champ de saisie (entry), champ texte (text) et étiquettes (label).
- Communication bidirectionnelle entre le pico et votre interface (machine hôte) pour la transmission des données
- Vous devez avoir plusieurs interactions entre votre interface et les capteurs de votre platine

## TESTS UNITAIRES : CRÉATION DE 2 TESTS

Vous devez tester votre application en utilisant un module de tests possible : `unittest`. Vous devez au moins **me créer deux tests unitaires** avec le module choisi. Dans chacun des cas, vous devez me donner les résultats des deux tests.

## STOCKAGE DES DONNÉES DES CAPTEURS

Vous devez créer une base de données locale SQLite pour stocker vos données des capteurs utilisées dans votre projet. Vous devez avoir au moins une table. **Pour créer votre base de données, vous devez créer une classe qui va permettre de gérer la ou les tables et les données captées.**

Les éléments stockés dans la base de données sont les suivants: id unique, la date et l'heure de la captation, une description et la donnée captée. Assurez-vous de garder vos données antérieures déjà captées de votre base de données.

## RAPPEL : RESPECTS DES CONSIGNES

Assurez-vous de respecter une nomenclature au niveau de la déclaration de vos variables et vos méthodes. Ajoutez des commentaires pour certaines parties de votre code. Par exemple : partie graphique, méthode pour le clic du bouton, méthode pour l'écriture dans le fichier, méthode pour la lecture dans le fichier, etc.

Assurez-vous que votre programme ne plante pas lors de l'exécution. Donc, ajoutez des validations et des exceptions pour tous les cas. Par exemple : fichier existant dans le dossier, dossier relatif et non absolu, lecture/écriture dans un fichier, lecture/écriture/ouverture/fermeture dans une base de données, etc.

## QUELQUES IDÉES DE PROJETS POSSIBLES

- Clavier numérique : validation d'un code d'accès par LCD et interface
- Potentiomètre et barre graphique LED
- Direction/vitesse d'un moteur en utilisant un joystick ou deux boutons (+ et -) et l'interface
- Servomoteur et la direction en utilisant un joystick ou deux boutons (+ et -) et l'interface
- Dé électronique (jeu) : bouton pour le brassage aléatoire et affichage (LCD, écran 7-segments)
- Jeu avec joystick ou 2-4 boutons, interface et LCD pour résultats
- Jeu pour dribler un objet en utilisant le capteur de photorésistance
- Manette de jeu avec le capteur Joystick et 2-4 boutons, 4 LED, interface jeu et écran LCD
- Jeu pour deviner le chiffre avec écran 7-segment, clavier numérique et boutons pour les deux joueurs.
- Module RFID : lecture et écriture de données sur la carte ou la puce et validation de l'accès. Lecture de la carte et mettre les informations dans la base de données

## AUTRE POSSIBILITÉ :

C'est possible aussi de prendre l'option du TP4 que vous n'avez pas fait, d'ajouter d'autres capteurs et de modifier l'interface graphique.

Par exemple :

### Option A du TP4

- On peut ajouter d'autres capteurs de mesures (distance, température, barométrique, thermistance, photorésistance, inclinaison, etc.) dans le projet. On fait un choix dans l'interface graphique (liste ou liste déroulante ou autre) pour prendre la mesure selon le capteur sélectionné. On pourrait aussi ajouter un clavier numérique pour choisir le capteur de mesure.

### Option B du TP4

- Système d'alarme : on peut ajouter le clavier numérique pour entrer le code d'accès. On entre le code d'accès sur l'écran LCD et non dans l'interface graphique.

**Grille d'évaluation - Travail pratique synthèse (30% de la note finale)**

<b>Interface graphique</b>	<b>Note</b>
<ul style="list-style-type: none"><li>• (5) Interface graphique : présentation, création, affichage des données, widgets demandés</li><li>• (5) Fonctionnalités : envoie/réception des données, démarrage des capteurs, gestion des données</li><li>• (5) Programmation : qualité du code, validations, structure POO, collection, classe personnalisée</li><li>• (5) Stockage des données : sauvegarde dans une BD</li></ul>	<b>20</b>
<b>Programmation des capteurs</b>	<b>Note</b>
<ul style="list-style-type: none"><li>• (5) Fonctionnalités : choix des capteurs, interactions des capteurs avec l'interface graphique</li><li>• (5) Affichage/communication : affichage de l'information sur l'écran LCD, envoi des données vers l'interface</li><li>• (5) Programmation : qualité du code, validations, fils d'exécution</li></ul>	<b>15</b>
<b>Exécution du programme</b>	<b>Note</b>
<ul style="list-style-type: none"><li>• (5) Exécution du programme en mode graphique</li><li>• (5) Exécution du programme : GPIO et capteurs</li></ul>	<b>10</b>
<b>Validations et tests</b>	<b>Note</b>
<ul style="list-style-type: none"><li>• (8) Rapport de montage du projet GPIO</li><li>• (8) Exécution du montage GPIO (capteurs) avec explications</li><li>• (4) Étapes Git avec GitHub Classroom</li><li>• (4) Tests unitaires</li></ul>	<b>24</b>
<b>Note finale</b>	<b>69</b>