
TRAVAIL PRATIQUE SYNTHÈSE – PROJET

Cours 420-4B5-MO – Objets connectés

CONSIGNES ET REMISE

- **Date de remise du travail : Vendredi 19 mai 2022 avant 18h**
- Travail à faire individuellement.
- Utilisation de GitHub Classroom et du référentiel créé.
- Me remettre tous les documents dans un fichier compressé (ZIP) nommé ***TPSyntheseVotreNom.ZIP***
- Le travail pratique doit être fait **INDIVIDUELLEMENT**

PROJET DE VOTRE CHOIX AVEC LES CAPTEURS RASPBERRY PI ET PYTHON

Dans ce travail pratique, vous devrez programmer un projet Python de votre choix avec le GPIO et les capteurs proposés dans les spécifications de l'énoncé. Pour votre projet, vous devez me remettre les éléments suivants:

- Un rapport de montage (voir document dans Moodle à la semaine 5).
- Votre montage complété avec une photo ou une vidéo ou lors de la validation.
- Structure MVC ou POO pour le projet.
- Codage de votre projet avec Python et les modules créés et utilisés.
- **OBLIGATOIRE POUR LA CORRECTION DU TRAVAIL : Le résultat de votre projet (photos-montage ou vidéo) durant la validation à la semaine 14 et 15.**

VALIDATION DE L'ATELIER: SEMAINE 14-15 (OBLIGATOIRE)

Présentation du montage GPIO des capteurs et son exécution au professeur en classe. **Vous pouvez aussi faire la validation par une vidéo avec des explications.**

Groupe	Description	Date
Groupe 01	Montage GPIO : capteurs Exécution avec les GPIO	Semaine 14 : 5-10 mai
		Semaine 15 : 12-18 mai
Groupe 02	Montage GPIO : capteurs Exécution avec les GPIO	Semaine 14 : 4-10 mai
		Semaine 15 : 11-18 mai

CONFIGURATION DE GIT ET GITHUB CLASSROOM (NON OBLIGATOIRE)

NOTE : pour des raisons de difficultés de connexion avec le Raspberry Pi, vous n'êtes pas obligé d'utiliser GIT

Utilisation de GitHub Classroom pour le laboratoire. Assurez-vous d'utiliser le dépôt à distant créé dans GitHub Classroom comme étant votre *main* ou *master*. Voici l'assignation :

Assignations	Description	Lien
TravailPratiqueSynthese	Dépôt du travail pratique synthèse	https://classroom.github.com/a/r3lepRew

Découpez votre programme en plusieurs parties et faites 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** : LED, LED RVB, bouton-poussoir, détection de mouvement PIR, distance ultrason, inclinaison, température/humidité, thermistance, photorésistance, clavier numérique, module RFID, moteur, servomoteur, joystick, potentiomètre, **autre (à valider avec le professeur)**
- Utilisation **d'au minimum, d'un capteur d'affichage** : écran LCD, matrice LED, écran 7-segments, barre graphique LED.
- Création d'une interface utilisateur avec un titre
- Votre interface utilisateur doit respecter le modèle MVC (Modèle-Vue-Contrôleur) ou 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).
- 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` ou `pytest`. 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.

RAPPORT DE MONTAGE POUR LES PROJETS PYTHON GPIO

Pour votre projet Python GPIO, vous devez me créer un rapport de montage qui comprend le matériel requis, le schéma de votre projet, une description détaillée du projet et une photo du projet terminé. Un canevas, sous forme de document Word, est disponible à la semaine 6.

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

AVEC FREENOVE

- Potentiomètre et barre graphique LED
- Clavier numérique : validation d'un code d'accès par LCD et interface
- Direction/vitesse d'un moteur en utilisant un joystick et l'interface
- Servomoteur et la direction en utilisant un joystick et l'interface
- Dé électronique (jeu) : bouton pour le brassage aléatoire et affichage (LCD, matrice ou écran 7-segments)
- Jeu Snake avec joystick, interface et LCD pour résultats
- Jeu Pong avec joystick, interface et LCD pour résultats
- Autre jeu avec joystick, interface et LCD pour résultats
- Jeu pour dribbler un objet en utilisant le capteur de photorésistance
- Capteur Joystick, interface jeu et écran LCD
- 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

AVEC ABRA OU AUTRE

- Potentiomètre et barre graphique LED
- Clavier numérique : validation d'un code d'accès par LCD et interface
- Direction/vitesse d'un moteur en utilisant deux boutons (+ et -) et l'interface
- Servomoteur et la direction en utilisant deux boutons (+ et -) et l'interface
- Dé électronique (jeu) : bouton pour le brassage aléatoire et affichage (LCD, écran 7-segments)
- Jeu Snake avec 4 boutons, interface et LCD pour résultats
- Jeu Pong avec 2 boutons, interface et LCD pour résultats
- Autre jeu avec 2 ou 4 boutons, interface et LCD pour résultats
- Jeu pour dribbler un objet en utilisant le capteur de photorésistance
- Capteur 4 boutons (gauche, droite, haut et bas), interface jeu et écran LCD
- Détection de mouvement avec le PIR, validation avec le clavier numérique

Grille d'évaluation - Travail pratique synthèse (20% de la note finale)

Spécifications générales	Note
<ul style="list-style-type: none"> (5) Rapport de montage du projet Python GPIO (5) Validation de l'historique du projet dans Git 	5
Projet Python et GPIO : spécifications	Note
<ul style="list-style-type: none"> (4) Type/classe personnalisé et utilisation de celui-ci (3) Utilisation d'une collection (Liste, dictionnaire ou autre) (4) Utilisation des capteurs proposés (3) Création d'une interface utilisateur avec titre (4) Interface : respect modèle MVC ou POO (4) Design de l'interface avec les widgets minimums (4) Interactions entre les capteurs et l'interface (4) Respect des consignes 	30
Tests et déploiement de l'application	Note
<ul style="list-style-type: none"> (5) Création de 2 tests unitaires et résultats 	5
Validation du travail synthèse	Note
<ul style="list-style-type: none"> (5) Description de votre projet d'objets connectés et du matériel choisi : capteurs, actionneurs, actuators (5) Montage de votre projet : connexion GPIO et capteurs (5) Démonstration avec l'interface utilisateur (10) Exécution de votre projet : interactions entre l'objet connecté et l'interface, collecte des données, sauvegarde des données, affichage des données. 	25
Note finale	65