

TP1

Réalisation un système IoT de base

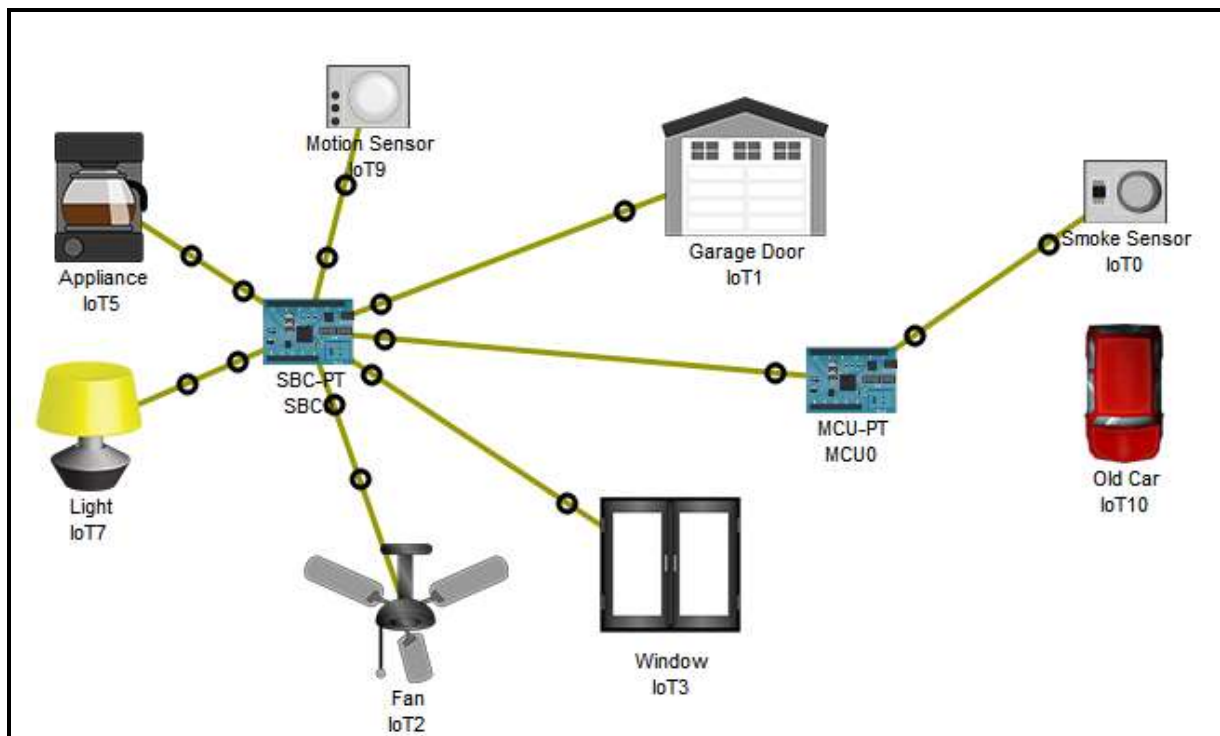
Objectif

L'objectif de ces travaux pratiques est d'apprendre à utiliser Python pour programmer un ordinateur mono carte SBC (Single Board Card) et une unité MCU (MicroController Unit) de Packet Tracer afin de réaliser un simple système IoT composé d'un ensemble de capteurs permettant d'analyser l'environnement. Cette analyse permettra la réaction de quelques équipements IoT intelligents.

Pour cela, les étapes du travail demandé se résument aux points suivants :

- Réaliser le réseau composé de capteurs, de cartes de traitement et d'équipements intelligents
- Programmer l'unité MCU pour détecter la fumée et envoyer l'information à la carte SBC
- Programmer la carte SBC pour activer le fonctionnement des équipements IoT qui y sont connectés

Topologie



Contexte & description du système IoT

Dans ce TP, vous allez réaliser un système IoT qui permet de :

1. Mettre en marche une machine à café et allumer une lampe se trouvant sur le plan de travail en détectant la présence d'une personne dans la cuisine
2. Ouvrir la porte du garage et la fenêtre et mettre en marche un ventilateur, en cas d'existence de fumée dans la maison.

La présence d'une personne dans la cuisine sera détectée à travers un capteur de mouvement de sortie numérique, et l'existence de fumée sera repérée par un capteur de fumée analogique

I. Tâche 1 : Ajout & connexion des appareils nécessaires

Étape 1: Réaliser le système IoT illustré dans la figure en dessus, en ajoutant les appareils suivants :

- Une machine à café est disponible sous **End Devices >> Home >> Appliance**
- Une lampe est disponible sous **End Devices >> Home >> Light**.
- Un ventilateur est disponible sous **End Devices >> Home >> Ceiling Fan**.
- Une fenêtre intelligente est disponible sous **End Devices >> Home >> Windows**.
- Une porte de garage est disponible sous **End Devices >> Home >> Garage door**.
- Un détecteur de mouvement est disponible sous **Components >> Sensors >> Motion Sensor**
- Un détecteur de fumée est disponible sous **Components >> Sensors >> Smoke Sensor**
- Un ordinateur mono carte SBC est disponible dans **Components >> Boards >> SBC-PT**.
- Un microcontrôleur MCU est disponible dans **Components >> Boards >> MCU-PT**

Ajouter une voiture de collection à placer devant le capteur de fumée pour le test du fonctionnement du système. Cette voiture est disponible dans **End Devices >> Smart City >> old car**.

Étape 2: connecter les appareils IoT aux cartes SBC et MCU en utilisant des câbles « lot Custom Cable » sous connections. Utiliser le tableau suivant :

SBC		MCU	
périphérique	port	périphérique	port
Machine à café	D0	SBC	D0
lampe	D1	Capteur de fumée	A0
Ventilateur	D5		
Fenêtre	D4		
Porte de garage	D2		
Capteur de mouvement	D3		
MCU	D6		

Pourquoi faut-il connecter le capteur de fumée à la MCU et pas la carte SBC ?

II. Tâche 2 : Programmation de la MCU

La MCU se comporte comme un microcontrôleur Arduino, dont l'un des avantages est la possibilité de le programmer avec Python.

L'objectif de cette tâche est de programmer la MCU pour lire une donnée depuis l'entrée A0 et l'envoyer sur le pin de sortie D0

Remarque : le Python utilisé dans PT est un interpréteur Python-JavaScript open source qui n'est pas mis à jour vers Python 3.0. C'est pourquoi, il peut y avoir de légères différences de syntaxe entre le code dans PT et celui des appareils qui utilisent Python 3.0.

Étape 1: Cliquez sur la MCU, puis sélectionnez l'onglet **Programming**. Y a-t-il déjà du code chargé ?

Cliquer sur le bouton **New** situé au-dessus du panneau de gauche pour créer un nouveau fichier. Appeler-le **main.py**, puis sélectionnez le type de modèle **Empty - Python**. Cliquez sur **Créer**. Votre nouveau fichier « main.py » doit maintenant s'afficher sur le panneau de gauche.

Dans toute la suite, utiliser le help de l'API Python accessible depuis le lien file:///C:/Program%20Files/Cisco%20Packet%20Tracer%207.3.1/help/default/iot_python_api.htm

Étape 2: Commencer le programme main.py par inclure les modules « **gpio** » et « **time** » pour supporter les fonctions à utiliser dans la suite du programme.

Définir dans la ligne suivante la fonction « **main()** ».

Étape 3: Dans la fonction main, ajouter une boucle infinie « **while** » de condition toujours vrai, dans laquelle effectuer le traitement suivant :

- Lire la donnée analogique provenant du pin A0, appeler cette variable « **smoke_sensor** ». Quelle est la fonction à utiliser ?
- Afficher sur la console Python de la MCU la valeur de cette variable, utiliser la fonction « **print()** »
- Tester si cette valeur est > 10 :
 - Ecrire la valeur « **HIGH** » sur le pin numérique D0. Quelle est la fonction à utiliser ?
 - Attendre un délai de 500 s avant de refaire le test
- Si la valeur de la variable « **smoke_sensor** » a atteint 0 :
 - Afficher sur la console Python de la MCU la phrase suivante : « Plus de fumée !! »
 - Ecrire la valeur « **LOW** » sur le pin numérique D0.
 - Attendre un délai de 500 s avant de refaire le test
- Terminer la fonction « **main()** »

III. Tâche 3 : Programmation de SBC

Le SBC se comporte comme un ordinateur monocarte réel tel que le Raspberry Pi. De la même façon que la MCU, on va le programmer en utilisant le Python.

Étape 1: Cliquez sur le SBC, puis sélectionnez l'onglet **Programming**. Cliquez sur le bouton **New** et créer un nouveau fichier « **main.py** » de type « **Empty – Python** ». Cliquez sur **Créer**.

Étape 2: Commencer le programme main.py par inclure les modules « **gpio** » et « **time** ».

Déclarer dans la ligne suivante le pin D6 comme entrée numérique. Quelle fonction faut-il utiliser ?

Définir dans la ligne suivante la fonction « **main()** ».

Étape 3: Dans la fonction main, ajouter une boucle infinie « **while** » de condition toujours vrai, dans laquelle effectuer le traitement suivant :

- Lire les données numériques provenant des pins D2 et D6, appeler ces variables respectivement « **motion_sensor** » et « **smoke_sensor** ».
- Commencer par le traitement relatif au capteur de mouvement : tester si la valeur de la variable correspondante est « HIGH » :
 - Afficher sur la console Python du SBC les phrases : « quelqu'un est présent » & « préparer le café »
 - Envoyer les valeurs qui permettent d'activer la machine café et la lampe sur respectivement D0 et D1. Quelle fonction faut-il utiliser ?
 - Attendre un délai de 6000 secondes
 - Désactiver le fonctionnement de la machine café et la lampe.
 - Attendre un délai de 500 s avant de refaire le test
- Afficher sur la console Python la valeur de la variable « smoke sensor », et tester si sa valeur est « HIGH » :
 - Afficher sur la console Python du SBC les phrases : « il y a du fumée dans l'espace » & « ouvrir la porte du garage et la fenêtre »
 - Envoyer les valeurs qui permettent d'ouvrir la porte du garage et la fenêtre, et activer le ventilateur, sur respectivement les pins D3, D4 et D5.
 - Attendre un délai de 500 s avant de refaire le test
- Tester si la valeur de la variable « smoke sensor » est « LOW » :
 - Envoyer les valeurs qui permettent de fermer la porte du garage et la fenêtre, et désactiver le ventilateur, sur respectivement les pins D3, D4 et D5.
 - Attendre un délai de 500 s avant de refaire le test
- Terminer la fonction « main() »

IV. Tâche 4 : phase de test

Pour tester votre code, exécutez d'abord **main.py** de la MCU et du SBC en cliquant sur Exécuter.

Étape 1: appuyez sur la touche Alt tout en déplaçant votre curseur au-dessus du détecteur de mouvement. Cette action simule un mouvement qui est alors détecté par le détecteur de mouvement Packet Tracer.

Décrire comment réagit le système IoT.

Étape 2: appuyez sur la touche Alt et cliquer sur la voiture de collection pour démarrer son moteur. Une fumée sera détectée par le capteur de fumée.

Décrire comment réagit le système IoT.

Que se passe-t-il si on arrête le moteur de la voiture ?

Étape 3 : Questions :

1. Quelles modifications faut-il effectuer pour éviter d'utiliser l'unité MCU ?
2. Que proposez-vous pour ajouter la fonctionnalité de notifier un utilisateur par SMS en cas de détection de fumée dans la maison ?