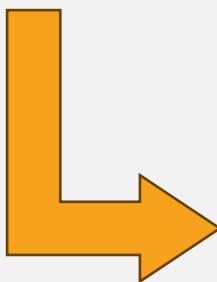


Système didactique

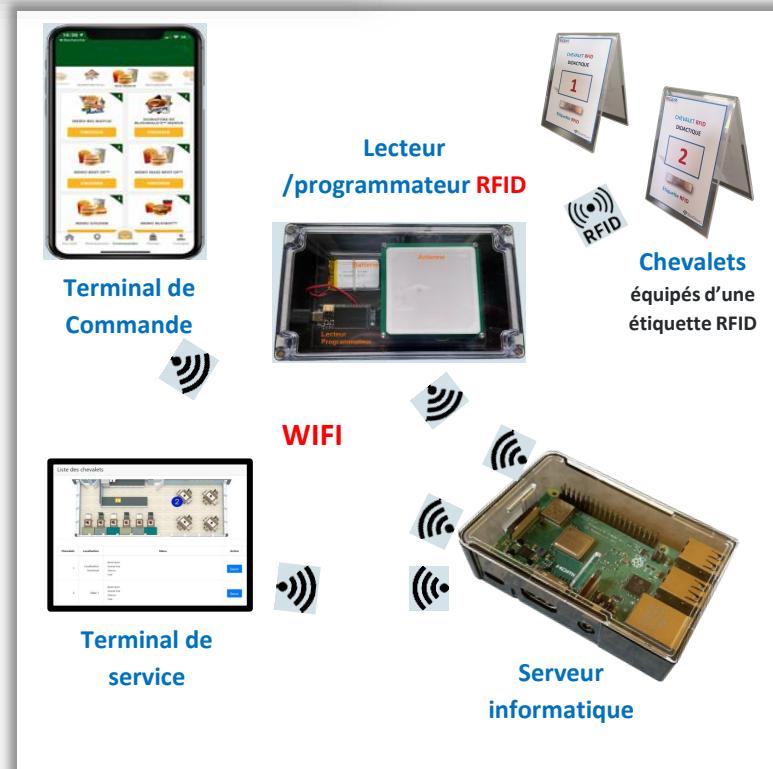
Chevalet de localisation par RFID



Du système réel..



...au Système didactique



Descriptif du système industriel Chevalet RFID

En restauration rapide, un système de localisation, basé sur une solution **RFID**, permet de **géolocaliser un client** pour qu'un serveur puisse lui apporter la commande à la bonne table.

Pour ces restaurants, les avantages du système industriel permettent un gain de temps avec un service à table rapide et sans erreurs, le client pouvant s'attabler à l'endroit souhaité.

Le traitement des commandes est simplifié et augmente la productivité et la rentabilité. Le **phasage** est le suivant :

Le client commande à la borne libre-service, prend et saisit le numéro d'un chevalet.

A



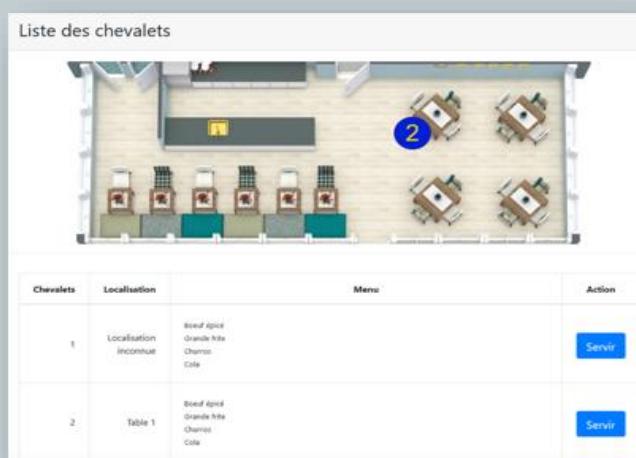
Le client s'installe dans la salle du restaurant et pose le chevalet sur sa table.

B



Un employé prépare la commande. L'employé(e) regarde sur le plan de salle affiché à l'écran, où se trouve le client.

C



Dès que la commande est préparée, l'employé(e) apporte la commande au client et reprend le chevalet.

D



DESCRIPTIF DE L'OST

SYSTÈME DIDACTIQUE INSTRUMENTÉ

L'OST (Objet et Système Technique), système didactique chevalet **RFID** permet la mise en oeuvre totale du process allant de la commande d'un menu à sa livraison sur la table client géolocalisée. Il reprend le même **phasage** décrit précédemment pour la solution industrielle. Les composants du système Chevalet RFID sont une adaptation didactique du système réel et assurent les mêmes fonctions techniques.

Les éléments en vert sont ajoutés pour une exploitation pédagogique.

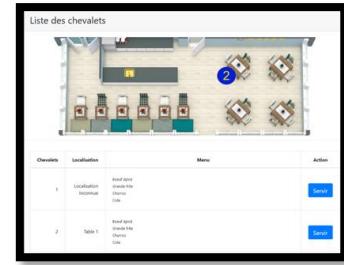
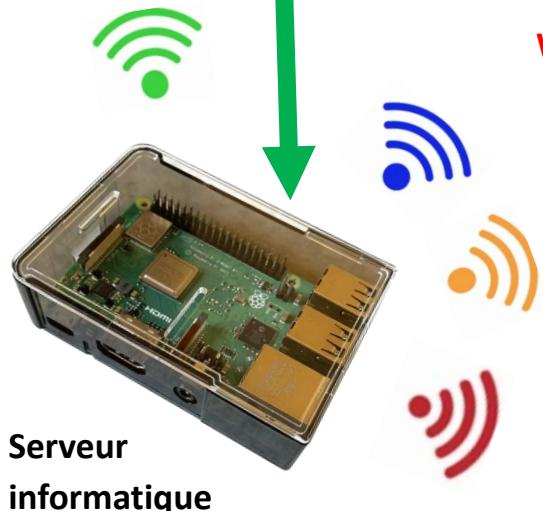
PC d'exploitation



Chevalets avec étiquette RFID



Lecteur /programmateur **RFID**



Terminal de Service,
non fourni



Terminal de
Commande, **non**
fourni

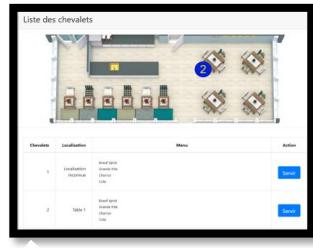
FONCTIONNEMENT DU SYSTÈME DIDACTIQUE EN MODE AUTONOME

- Le lecteur scanne en permanence les chevalets à proximité et envoie leur identifiant au serveur informatique Raspberry Pi.
- Le lecteur est autonome** mais peut également interagir avec l'ordinateur d'exploitation pour réaliser des opérations spécifiques. **Il peut même être reprogrammé ponctuellement (scripts micropython fournis)**, de façon non permanente : **le programme par défaut est exécuté à chaque démarrage et ne peut pas être écrasé**.
- Le serveur informatique basé sur une carte Raspberry Pi exécute un serveur Web «[Restaurant](#)» composé d'une partie «[Utilisateur](#)», permettant de réaliser un processus de passage de commande et d'une partie «[Administration](#)», permettant de gérer les commandes et de localiser les chevalets.
- Les terminaux ne sont pas fournis**

Terminal de commande
iPhone, Androïd, PC



Terminal de Service
iPhone, Android, PC



Chevalets avec étiquette RFID

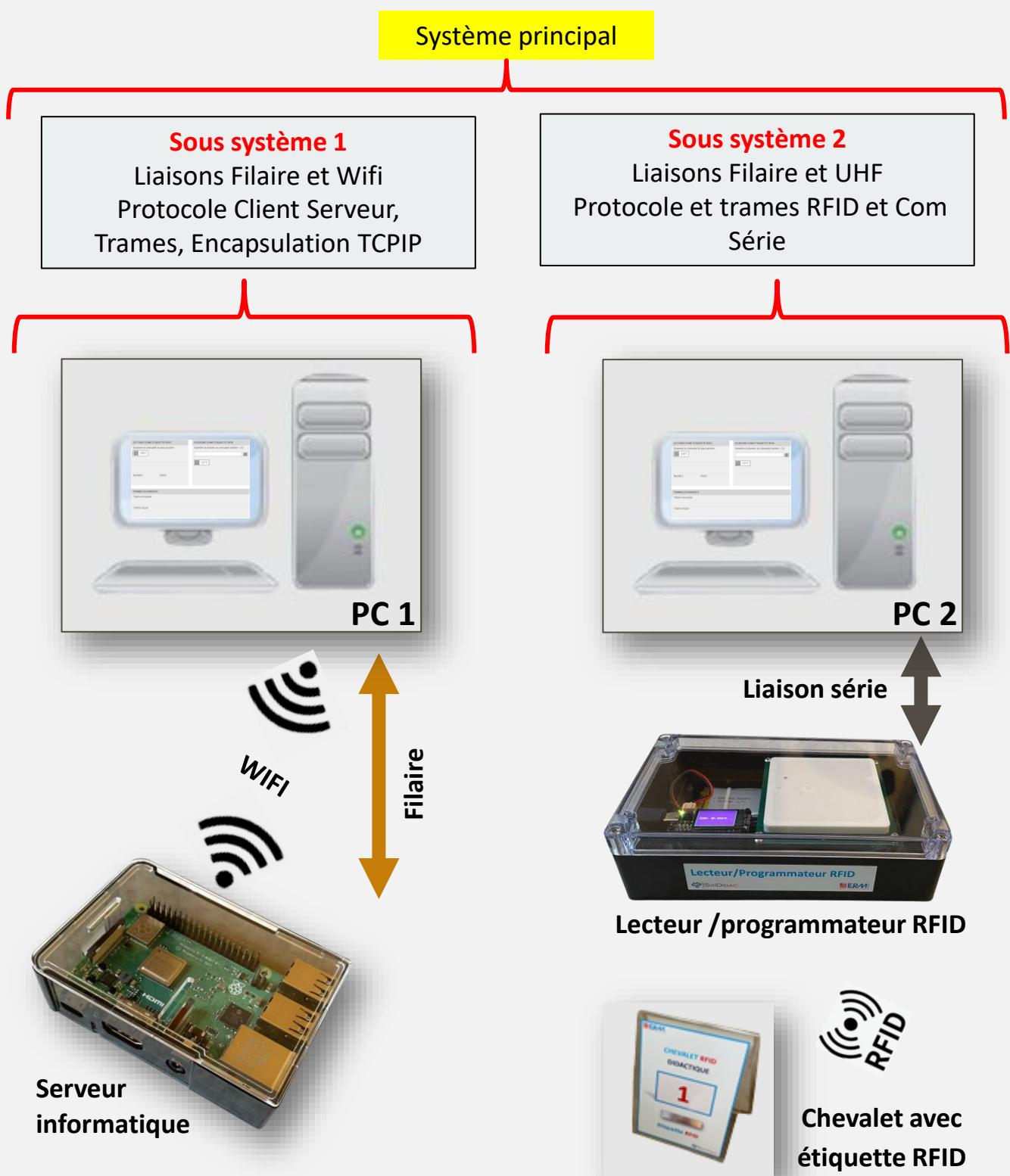


Serveur informatique
Raspberry Pi

Lecteur /programmateur RFID
Microcontrôleur
Afficheur Led
ESP 32-S3 TFT Feather
Module YRM 100 UHF RFID
Batterie

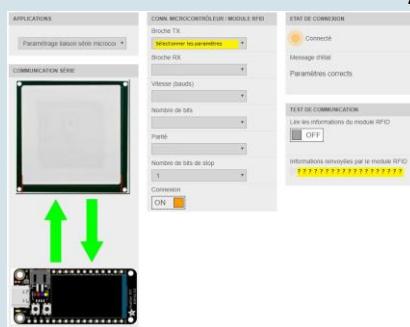
ORGANISATION POUR LA RÉALISATION DES TP

Le système principal **Chevalet RFID** est composé de 2 sous systèmes permettant la mise en œuvre séparée de la réalisation des TP associés. Il est complété par un terminal de commande et un terminal de service pour la mise en œuvre totale du process allant de la commande d'un plat à sa livraison sur table.

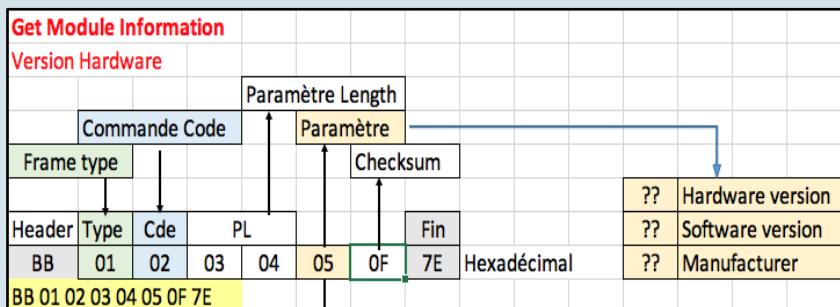
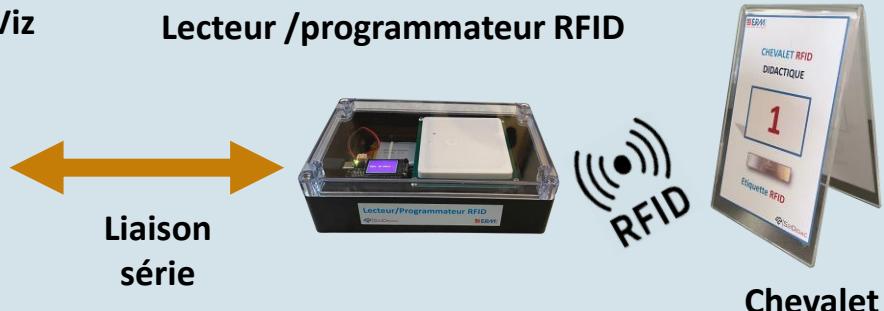


TP4 STI2D 1 ^{ere}		PROTOTYPER ET INTEGRER REQUÊTES RFID (Script Python)
Compétences développées		C06.5 Interpréter les résultats d'une simulation et conclure sur la performance de la solution. C05.8 Rechercher et écrire l'algorithme de fonctionnement puis programmer la réponse logicielle relative au traitement d'une problématique posée.
Connaissances		6.2 Expérimentations de constituants de la chaîne d'information. 6.3 Intégration des éléments prototypés du produit. 2.4.3 Encodage de l'information : binaire, hexadécimal, ASCII.
Ressources		Sous Système avec lecteur et dossier technique, programmes Python
Problématique		Géolocaliser chevalet RFID d'un restaurant fastfood
Activités du TP		<p>A partir du document M100 Communication Protocol et du tableau de bord MyViz: Paramétriser, tester la communication entre le microcontrôleur et le module RFID.</p> <p>Lire les informations du module</p> <ul style="list-style-type: none"> Dans le tableau d'assistance Excel, saisir les arguments de la trame de commande «Get Module Information» pour lire les données du module RFID Copier la trame finale dans MyViz (trames brutes) pour tester cette requête Convertir la réponse brute en caractères ASCII pour rendre la lecture possible. Compléter le programme Python donné assurant l'intégration des requêtes prototypées, l'exécuter et le valider pour lire les informations du module RFID

PC avec Tableau de Bord MyViz



Lecteur /programmateur RFID



```

# Fabricant
trame_commande = "????????????????"
print("Fabricant")
print(ecriture_lecture(trame_commande))

# Version de hardware
trame_commande = "????????????????"
print("Version de hardware")
print(ecriture_lecture(trame_commande))

# Version de software
trame_commande = "????????????????"
print("Version de software")
print(ecriture_lecture(trame_commande))

# Scan des lecteurs à proximité
trame_commande = "????????????????"
print("Scan")
print(ecriture_lecture(trame_commande))
  
```

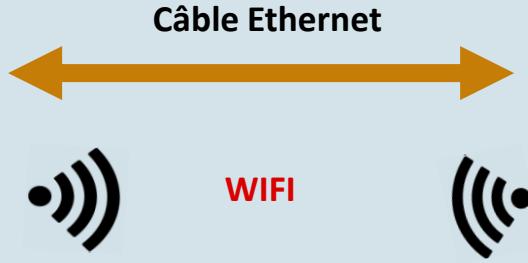
Prototypage trame RFID

Intégration

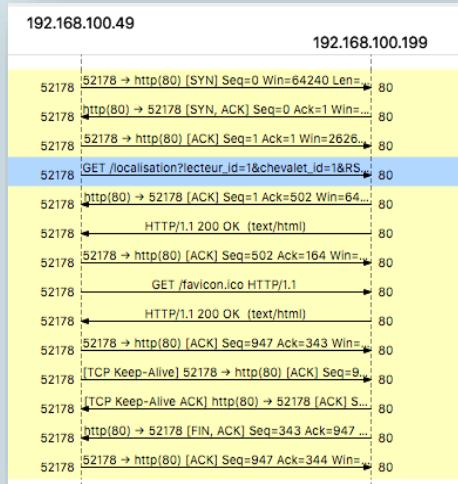
Programme python à commenter, compléter, exécuter, valider

TP5 STI2D 1 ^{ère}	DIALOGUE CLIENT SERVEUR
Compétences développées	C07.6 Expérimenter des architectures matérielles et logiciels en réponse à une problématique posée
Connaissances	<p>2.4.4 Transmission de l'information</p> <ul style="list-style-type: none"> Architecture d'un réseau informatique, architecture Client/serveur, protocole, encapsulation des données, adresse et physique et adresse logique IPV4 <p>3.4.3 Interopérabilité des produits</p> <ul style="list-style-type: none"> Configuration d'un réseau : adressage statique, dynamique, communication au sein d'un réseau : trames TCP/IP, protocoles HTTP.
Ressources	Sous Système Client/Serveur et dossier technique
Problématique	Transmettre la localisation des chevalets au serveur informatique
Activités du TP	<p>A partir de MyViz, établir connexion WIFI entre PC et serveur HTTP et valider la communication par échange de messages.</p> <ul style="list-style-type: none"> Justifier l'adresse IP du client et le type d'adressage Envoyer une requête HTTP au serveur pour lui indiquer la position d'un chevalet et la valeur du RSSI associée. Valider la réponse du serveur Analyse trames (données ou capturées): Ethernet, protocole TCPIP, HTTP requêtes du client

PC avec Tableaux de Bord



Serveur informatique



Time	Source	Destination	Protocol	Length	Port	Mac
0.000000	192.168.100.49	192.168.100.199	TCP	66	52178	c8:a3:62:12:d1:23
0.001530	192.168.100.199	192.168.100.49	TCP	66	http	HewlettP_05:08:03
0.001684	192.168.100.49	192.168.100.199	TCP	54	52178	c8:a3:62:12:d1:23
0.004603	192.168.100.49	192.168.100.199	TCP	66	52179	c8:a3:62:12:d1:23
0.007163	192.168.100.199	192.168.100.49	TCP	66	http	HewlettP_05:08:03
0.008087	192.168.100.49	192.168.100.199	HTTP	555	52178	c8:a3:62:12:d1:23
0.0087163	192.168.100.199	192.168.100.49	TCP	66	http	HewlettP_05:08:03
0.007259	192.168.100.49	192.168.100.199	TCP	54	52179	c8:a3:62:12:d1:23
0.037866	192.168.100.199	192.168.100.49	HTTP	217	http	HewlettP_05:08:03
0.0877212	192.168.100.49	192.168.100.199	TCP	54	52178	c8:a3:62:12:d1:23
0.266222	192.168.100.49	192.168.100.199	HTTP	499	52178	c8:a3:62:12:d1:23
0.278379	192.168.100.199	192.168.100.49	HTTP	233	http	HewlettP_05:08:03
0.328375	192.168.100.49	192.168.100.199	TCP	54	52178	c8:a3:62:12:d1:23
45.008891	192.168.100.49	192.168.100.199	TCP	55	52179	c8:a3:62:12:d1:23
45.010414	192.168.100.199	192.168.100.49	TCP	66	http	HewlettP_05:08:03
45.289359	192.168.100.49	192.168.100.199	TCP	55	52178	c8:a3:62:12:d1:23
45.291615	192.168.100.199	192.168.100.49	TCP	66	http	HewlettP_05:08:03
60.005417	192.168.100.199	192.168.100.49	TCP	60	http	HewlettP_05:08:03
60.005467	192.168.100.49	192.168.100.199	TCP	54	52179	c8:a3:62:12:d1:23
75.279706	192.168.100.199	192.168.100.49	TCP	60	http	HewlettP_05:08:03
75.279750	192.168.100.49	192.168.100.199	TCP	54	52178	c8:a3:62:12:d1:23
105.017397	192.168.100.49	192.168.100.199	TCP	55	52179	c8:a3:62:12:d1:23
105.019751	192.168.100.199	192.168.100.49	TCP	60	http	HewlettP_05:08:03

Diagramme des flux

Liste
paquets

TP6 STI2D T^{ale}		MODIFIER DONNEES DANS ETIQUETTE RFID (numéro chevalet)
Compétences développées		C06.5 Interpréter les résultats d'une simulation et conclure sur la performance de la solution. C07.6 Expérimenter des architectures matérielles et logicielles en réponse à une problématique posée
Connaissances		6.2 Expérimentations de constituants de la chaîne d'information. 6.3 Intégration des éléments prototypés du produit.
Ressources		Sous système avec lecteur et dossier technique
Problématique		Renuméroter les chevalets RFID
Activités du TP		<p>A partir du TB MyViz, et des tableaux d'assistance Excel (saisie trame, calcul checksum, conversion Hexa/ASCII) et outil calcul CRC 16 bits:</p> <ul style="list-style-type: none"> Créer et tester les 4 trames RFID pour: <ul style="list-style-type: none"> Faire inventaire des chevalets présents et justifier sur quel critère l'ordre d'affichage de ceux-ci est fait dans la trame de réponse. Sélectionner le chevalet à modifier avec filtrage EPC Écrire le numéro demandé dans la zone mémoire utilisateur de l'étiquette, Vérifier l'écriture et valider le résultat Commenter le programme Python donné assurant l'intégration des requêtes prototypées (1 à 4) en nommant chaque fonction Exécuter le programme

PC avec Tableaux de bord



Liaison
série

Lecteur /programmateur RFID

Chevalets



Trame réponse chevalet 1								Protocole Control										PC+EPC					
header	Type	Cde	Para Length	RSSI	PC		EPC correspondant au N° de votre chevalet (96 bits)										CRC		Check	Fin			
BB	02	22	00	11	C2	34	00	E2	00	47	0A	81	A0	68	21	07	96	01	0D	98	11	5C	7E
	02	34	00	17	194	52	00	226	00	71	10	129	160	104	33	07	150	01	13	152	17		1372
02 22 00 11 C2 34 00 E2 00 47 0A 81 A0 68 21 07 96 01 0D 98 11 5C																							

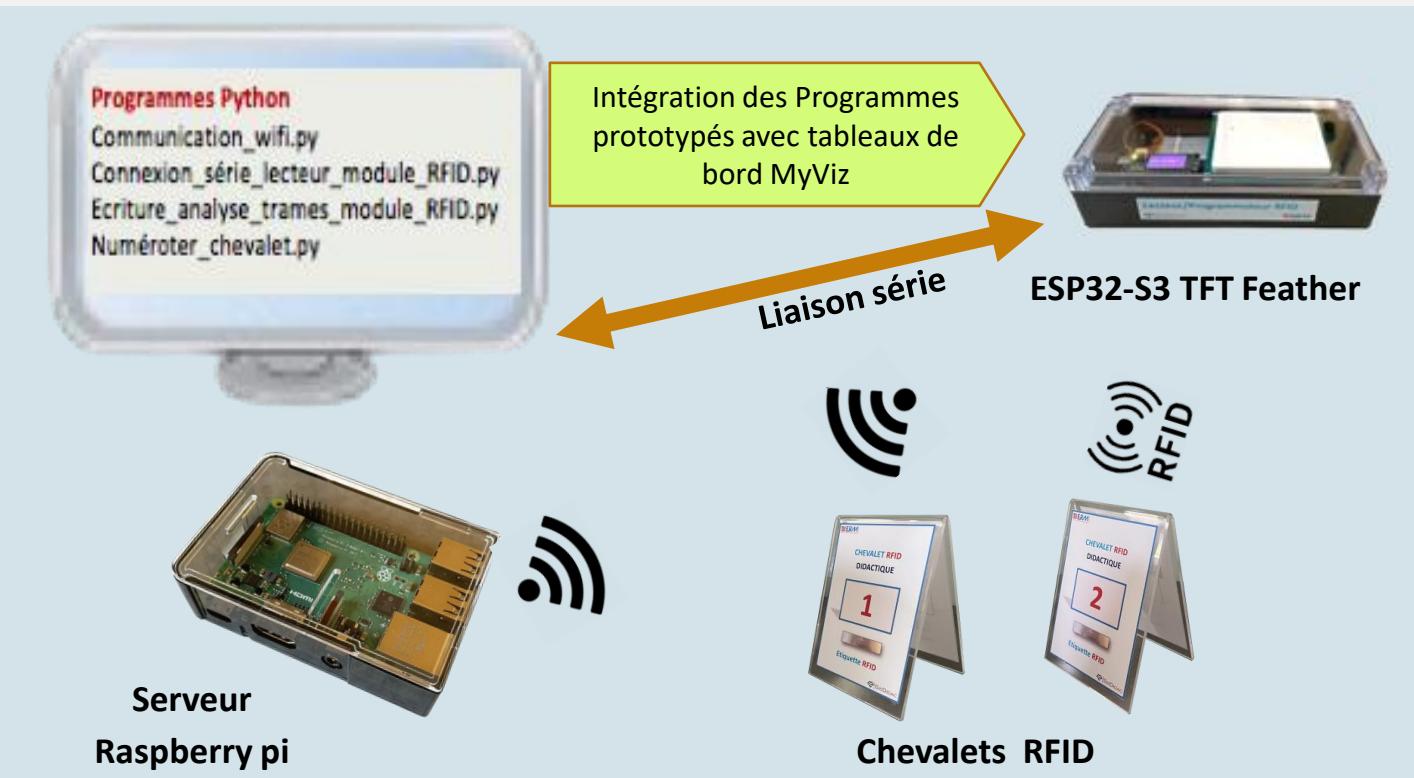
Prototypage Trame RFID

Intégration trames prototypées
dans programme Python

Intégration

```
# Scan des lecteurs à proximité
print("Scan")
tags_list = lecteur.inventory()
for tag in tags_list:
    print(tag)
```

TP7 STI2D T^{ale}	MODIFIER, COMMENTER PROGRAMME PYTON
Compétences développées	CO3.2 Identifier et caractériser l'agencement logiciel d'un produit. CO6.5 Interpréter les résultats d'une simulation et conclure sur la performance de la solution.
Connaissances	4.3.5. Conception informationnelle des produits, structures de programmation. <i>Fonctions logicielles. Méthodes et propriétés utiles en lien avec les bibliothèques logicielles choisies.</i> <i>Codage dans un langage spécifique. Règles d'écriture (organisation du code, commentaires, documentation, etc.).</i> 6.3 Intégration des éléments prototypés du produit.
Ressources	c et lecteur connectés par liaison série, serveur, script Micropython, logiciel Thonny, dossier technique
Problématique	Communiquer avec le module RFID et le serveur sans IHM
Activités du TP	<p>A partir d'un script micro-python partiel « élève », assurant les mêmes fonctionnalités qu'un tableau de bord MyViz correspondant : A l'aide du logiciel Thonny (ou autre) installé sur PC: Suivre les consignes données, éditer, compléter, commenter le fichier source. Exécuter et valider les scripts.</p> <p>Nota: Plusieurs scripts disponibles à choisir par le professeur selon les domaines:</p> <p>Domaine communication</p> <ul style="list-style-type: none"> • Etablir connexion par liaison série: connexion_série_lecteur_moduleRFID.py • Etablir communication wifi avec le serveur: communication_wifi.py • Requêtes communication client/serveur : requête_brute_seveur_wifi.py <p>Domaine module RFID</p> <ul style="list-style-type: none"> • Ecriture et analyse de trames RFID: écriture_analyse_trames_module_RFID • Lecture/écriture dans la zone mémoire de l'étiquette du chevalet: numéroter.py



TP8 STID2 T^{ale}	DIAGNOSTIQUER PANNE DE COMMUNICATION CORRIGER ET VALIDER LE FONCTIONNEMENT
Compétences développées	CO5.2. Identifier et justifier un problème technique à partir de l'analyse globale d'un produit (information). CO5.5. Proposer des solutions à un problème technique identifié en participant à des démarches de créativité, choisir et justifier la solution retenue.
Connaissances	6.2 Expérimentations de constituants de la chaîne d'information; protocole d'essai 3.4.3 Configuration d'un réseau: adressage statique, dynamique.
Ressources	<ul style="list-style-type: none"> Système installé et configuré avec défaut(s) de fonctionnement. Sur PC: logiciel Putty et Angry IP Sur Android: Network Scanner; Dossier technique
Problématique	Diagnostiquer et corriger le défaut communication
Activités du TP	<p>A partir du système en mode dégradé, créé par le professeur à l'aide de programmes exécutables fournis:</p> <ul style="list-style-type: none"> Défaut sur le SSID du serveur: impossibilité de connexion du lecteur au serveur Défaut sur adresse IP du serveur: impossibilité de connexion du lecteur au serveur Défaut sur chevalet: impossibilité de détection du chevalet <ul style="list-style-type: none"> Diagnostiquer les pannes de communication Définir les procédures d'actions correctives. Corriger et valider le fonctionnement

PC avec Putty et Angry IP

scanner



Câble Ethernet



Câble USB



WIFI



Serveur informatique

Raspberry pi

facultatif:

Iphone, IPad,
Android

Lecteur /programmateur RFID



Chevalets RFID

FOURNITURE

Plusieurs configurations possibles

Configuration (pour activités collège & lycée)

- 1 lecteur/programmateur RFID
- 2 Chevalets RFID
- 1 Serveur informatique
- 1 Logiciel de paramétrage et de programmation
- 1 application HTML (Terminal de commande)
- 1 Application HTML (Terminal de service)



Référence S2I//2300

Des chevalets et des lecteurs peuvent être achetés sur demande en complément