

## Projet « Prise Connectée Intelligente Wifi »

### Contexte

**Le travail sera réalisé en groupe de 4 (sinon 5) étudiants. Vous avez environ 54 heures.**

**Vous pourrez trouver l'essentiel des informations sur le net.**

**Lisez attentivement tout le sujet et vous constaterez qu'il est possible de mener des tâches en parallèle !**

Il s'agit de réaliser une prise électrique connectée en Wifi comme celle proposée par la marque CHACON :



<https://www.chacon.be/fr/objets-connectes/46-chacon-prise-intelligente-wifi-5411478530121.html>

<p>Temps alloué <b>encadré</b> :</p> <p>CM : 1.5 heure</p> <p>TD : <math>2 \times 1.5\text{h} = 3</math> heures</p> <p>TP : <math>5 \times 3.5\text{h} = 17.5</math> heures</p> <p><b>TOTAL : 22 heures</b></p>	<p>Temps alloué en <b>autonomie</b> :</p> <p>TP : <math>10 \times 4.0\text{h} = \mathbf{40 \text{ heures}}</math></p>
---	---

**Temps global : 62 heures.**

### [CdC1] Cahier des charges

**Dans un premier temps, il vous est demandé :**

1. De concevoir la carte électronique de cette prise connectée en Wifi (cf. EasyEDA : <https://easyeda.com/fr> ).  
Les composants électroniques doivent répondre aux critères suivants (liste non exhaustive et sans ordre de préférence) :
    - a. Taille de la carte électronique
    - b. Taille de la prise électrique (estimation)
    - c. Poids (estimation)
    - d. Prix global de fabrication
    - e. Wifi intégré
    - f. Consommation
    - g. Etc.
- Vous devez justifier votre solution, notamment en énumérant vos critères par ordre de priorité !
2. L'allumage ou l'extinction de l'appareil branché sur la prise peut se faire depuis l'unique bouton poussoir (cf. le côté latéral gauche de la photo). Une LED indiquera l'état de l'interrupteur commandé. Noter que l'action sur le bouton poussoir doit être visible sur les interfaces web et Android demandées dans les parties suivantes ⓘ.
  3. De réaliser un serveur WEB hébergé par un Raspberry Pi.  
La page WEB doit pouvoir :
    - a. Commander au moins 2 prises en ON/OFF avec retour de l'état ON ou OFF de la prise.

- b. Activer au moins deux plages horaires journalières sur une des deux prises. Les plages horaires journalières seront programmées en « dur » côté Raspberry (donc pas depuis la page WEB, pour l'instant 😊). La page WEB doit indiquer si la programmation est activée ou non.

Le Raspberry PI peut indiquer, par des LEDs, pour chacune des prises :

- a. L'état ON ou OFF
- b. L'activation des plages horaires.

Les parties 1, 2 et 3 peuvent être menées en parallèle !

#### [CdC2] Dans un deuxième temps, on vous demande :

1. De réaliser une application Android reprenant les fonctionnalités proposées par la page WEB précédente.
2. De permettre la programmation d'au moins deux plages horaires journalières depuis la page WEB.
3. De sécuriser l'accès à la page WEB.
4. De sécuriser le transfert des données côté internet.
5. Utiliser un broker privé.

#### [CdC3] Dans un troisième temps, on vous demande :

1. D'ajouter sur la page WEB et l'application Android, un bouton qui permet d'allumer ou d'éteindre toutes les LEDs. Si jamais, une de ces deux conditions est satisfaite par des actions individuelles sur chacune des prises, ce bouton doit prendre l'état correspondant !
2. D'indiquer, sur la page WEB et l'application Android, la température mesurée par la prise à partir du capteur DS18B20.
3. De choisir, lors de la première installation de la prise, le réseau auquel elle doit se connecter. Pour se faire, il faudra qu'elle soit en mode AP (Acces point) avant d'être en STA (STand Alone).
4. De sécuriser l'accès à l'application Android.
5. De permettre la programmation d'au moins deux plages horaires journalières depuis l'application Android.

#### [CdC4] Dans un quatrième temps, on vous demande :

1. Faire une notification de changement d'état de la prise.
2. Faire une notification de seuil de température dépassée.
3. Envoyer par mail et SMS un message indiquant que la température a dépassé le seuil préconisé.

**Rappel : Certaines parties peuvent être menées en parallèle !**

**N'hésitez pas à sauvegarder à chaque fois que nécessaire vos différentes versions ou évolutions de vos programmes.**

#### Quelques mots clés :

- Broker public :
  - [test.mosquitto.org](http://test.mosquitto.org)
  - [broker.hivemq.com](http://broker.hivemq.com)
- ESP8266 :
  - <https://learn.adafruit.com/adafruit-feather-huzzah-esp8266>
  - <https://www.mouser.fr/ProductDetail/Espressif-Systems/ESP8266-DevKitC-02D-F?qs=qSfuJ%252Bfl%2Fd64058n5BJabA%3D%3D>
- IDE Arduino :
  - Bibliothèque MQTT : « PubSubClient » de Nick O'LEARY. S'inspirer fortement de l'exemple « mqtt\_esp8266 » !
  - Bibliothèque pour le DS18B20 : « MAX31850 DallasTemp » qui a besoin de la « MAX31850 OneWire » de Adafruit. S'inspirer fortement de l'exemple « simple » !
  - Bibliothèque WiFiManager de tzapu pour le point 3 du [CdC3]. S'inspirer fortement de l'exemple « basic » !
- Alimentation : recherche « convertisseur a montage pcb (ac/dc) » sur Farnell.com
- Commande relai :
  - <http://riton-duino.blogspot.com/2018/08/alimenter-un-relais-transistor.html>
  - <https://www.youtube.com/watch?v=0F17HFWdTEs>
  - <https://www.youtube.com/watch?v=no38Ov5gtIc>
  - [https://wiki.mdl29.net/lib/exe/fetch.php?media=elec:cours\\_transistor.pdf](https://wiki.mdl29.net/lib/exe/fetch.php?media=elec:cours_transistor.pdf)
- Notification et SMS : pushbullet
- SMS : gammu (pensez à enlever/désactiver le code PIN depuis votre smartphone !)