

**CoBox**

Github :

Site web : [cobox.paulitow.fr](http://cobox.paulitow.fr)

# CAHIER DES CHARGES



**Projet de système d'exploitation**  
**Moyse – Desmarets – Ilhe - Legeas**

**ISEN** | école  
d'ingénieurs  
ALL IS DIGITAL!

<b>1. Présentation .....</b>	<b>2</b>
<b>1.1 Besoins .....</b>	<b>2</b>
<b>1.2. Contrainte .....</b>	<b>3</b>
<b>1.3. Roadmap .....</b>	<b>4</b>
<b>2. Solutions .....</b>	<b>5</b>
<b>2.1. Matériel .....</b>	<b>5</b>
2.1.1. Centrale .....	5
2.1.2. Actionneur .....	5
<b>2.2. Logiciel .....</b>	<b>6</b>
2.2.1. Serveur central .....	6
2.2.2. Base De Données .....	6
2.2.3. IHM d’affichage des données .....	7
<b>3. Glossaire .....</b>	<b>8</b>

## I.1 Besoins

Le but du projet Cobox est de concevoir une solution de domotique pour particuliers, permettant le contrôle et la supervision de différents éléments (tout équipement pouvant être commandé par un relais) au sein d'un foyer.

Cette solution se décomposera en une centrale domotique, et X clients. Les clients étant les actionneurs / capteurs.

A ce jours les besoins concernant la centrale sont :

- Echanger des informations à ses actionneurs en mode serveur via Wifi directe (mode hotspot)
- Afficher de manière claire les mesures des différents éléments supervisés par la centrale

Et concernant la partie client :

- Echanger des informations à la centrale en mode client via Wifi directe
- Piloter les actionneurs via des GPIO\*
- Acquisition de données via des capteurs (Température / humidité...)

## I.2. Contrainte

Les contraintes de ce projet ont été définies en accord avec notre client et sont susceptibles d'être modifiées avec l'évolution du présent document.

Contraintes	
CENTRALE	La centrale doit comporter un système d'exploitation linux
	La centrale doit utiliser un programme développé en C pouvant dialoguer avec les différents actionneurs
ACTIONNEUR	L'actionneur doit être capable de dialoguer via Wifi
	L'actionneur doit être capable de piloter jusqu'à 8 relais
	L'actionneur doit être capable de monitorer un capteur et de transmettre sa valeur à la centrale

## I.3. Roadmap

Ci-dessous, la Roadmap définissant les étapes cruciales de notre projet

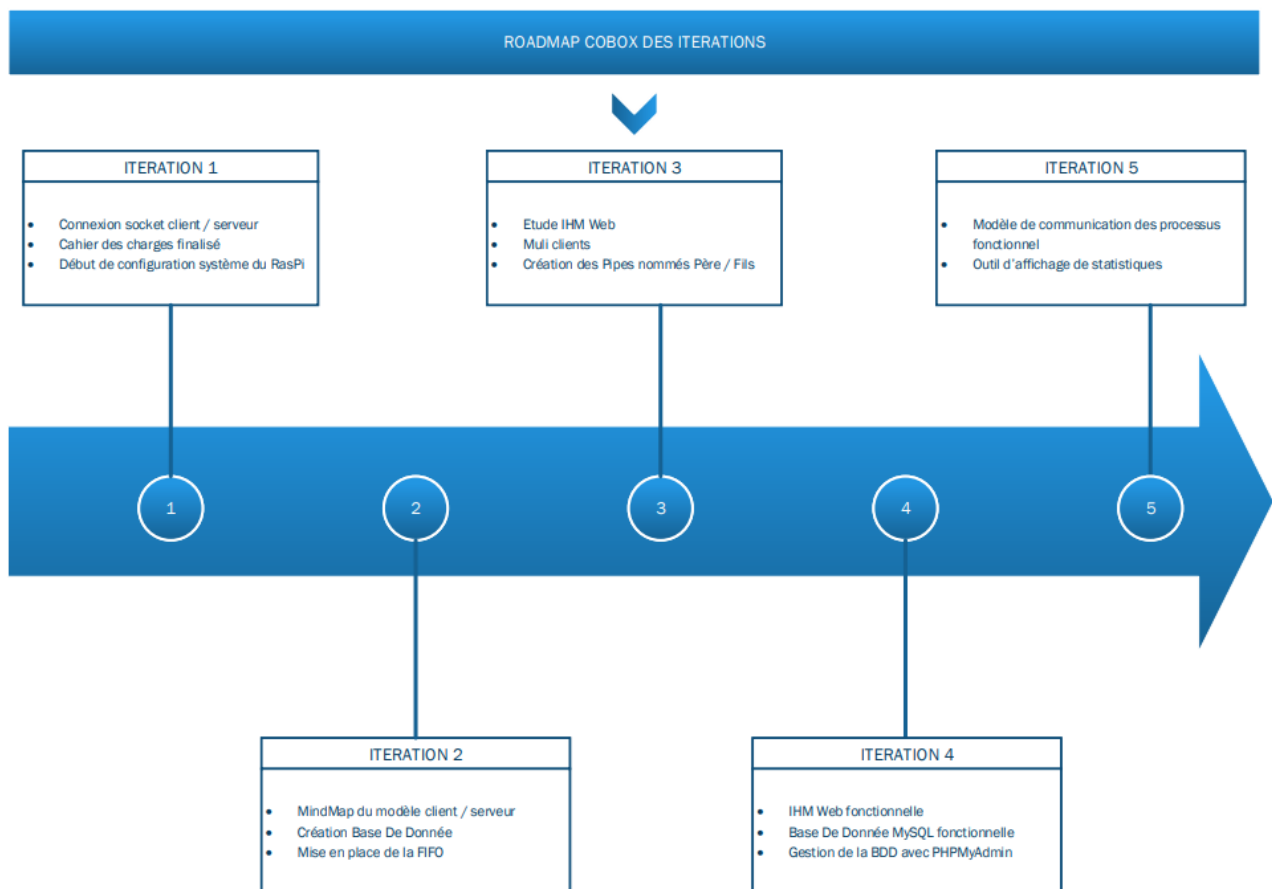


Figure 1 : Roadmap Cobox

### 2.1. Matériel

Ci-dessous sont détaillées les solutions matérielles pour répondre aux besoins

#### 2.1.1. Centrale

Pour répondre à l'ensemble des besoins et dans le but d'améliorer constamment le produit, le choix d'une RaspBerry Pi est retenu pour la centrale domotique.

La version de RaspBerry Pi sélectionnée est la « model 3 B+ ». L'architecture logicielle étant semblable entre les versions, le produit pourra être amené à migrer de plateforme vers une version 4 model B.

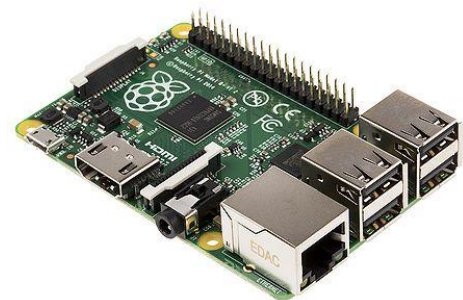


Figure 2 : RaspBerry 3 model B+

#### 2.1.2. Actionneur

Pour répondre aux besoins de l'actionneurs et dans un but de faible consommation, le choix d'un ESP8266 est retenu. Le modèle en question est le Weemos D1 mini.

Le Weemos D1 mini possède 8 entrées-sorties numérique ainsi qu'un port analogique.

Pour répondre aux besoins de communication, cette carte embarque un chipset Wifi ainsi que Bluetooth.



Figure 3 : Weemos D1 mini

## 2.2. Logiciel

### 2.2.1. Serveur central

Le logiciel de centralisation sera programmé en langage C. Il devra répondre aux contraintes décrites ci-dessus.

Le Logiciel permettra de gérer les parties suivantes :

- Communication serveur → fils en Pipe nommé (processus hérités)
- Communication fils → serveur en FIFO
- Communication fils → clients en socket TCP/IP bidirectionnel
- Communication clients → capteurs en GPIO

Le fonctionnement du programme est résumé via le schéma « Modèle de communication COBOX » disponible dans les sources sur GitHub.

### 2.2.2. Base De Données

Le programme archivera les données reçues dans une BDD\* MySQL. Cette base sera pilotée grâce à un interface graphique Web nommée PHP MyAdmin, géré par un serveur HTTP Apache2.

La structure sera déterminée dans un second temps.

Le choix de la BDD se porte sur MySQL car c'est une base certes commune mais étant légère et facile d'administration, donc importable sur une RasPi.

### **2.2.3. IHM d’affichage des données**

L’IHM\* d’affichage des données remontées sera accessible en localhost via navigateur (Web). Ce sera également un serveur HTTP Apache2, connecté à la BDD MySQL pour acquisition de données. Nous réglerons l’affichage sur une journée, mais cela sera paramétrable sur de plus longues durées par la suite.



- ❖ GPIO : General Purpose Input / Output
- ❖ BDD : Base De Données
- ❖ IHM : Interface Homme Machine