# Manuel d’utilisation R2D2

## Installation

La démarche ainsi que les fichiers d’installation aussi bien pour la beaglebone que pour le téléphone sont présentés dans le Readme du dépôt github : https://github.com/ISENRobotics/R2D2-in-the-sky

## Beaglebone

Les mots de passe et identifiants de la Beaglebone sont les suivants :

Utilisateur : root

Mdp : pas de mot de passe

Utilisateur : debian

Mdp : temppwd

## Réseau

La Beaglebone est configurée pour prendre l’adresse IP 172.16.0.2 au démarrage. Connectez-vous au réseau TP-Link\_9D54 pour pouvoir communiquer avec la Beaglebone.

De base sur le réseau, seule l’adresse LAN 172.17.0.2 peut communiquer avec la Beaglebone au travers du réseau. Les autres adresses IP disponibles (172.17.0.3-6) sont bloquées via l’administration réseau, elles ne peuvent essayer d’accéder qu’à l’administration réseau disponible à l’adresse 172.17.0.1.

La clé de connexion au réseau wifi du routeur est : 35899266

### Identifiants actuels :

Id : admin

Mdp : ej9pwKolm29

### Identifiants en cas de reset du routeur :

Id : admin

Mdp : admin

## Texte de l’écran LCD

Les textes défilants sur l’écran LCD sont disponibles dans le fichier texte\_lcd.txt présent dans le dossier classe. Pour rajouter des phrases dans ce texte, la manœuvre est simple :

Ouvrez le fichier texte\_lcd.txt. Toute les lignes commencent par ^^^^. Ce point est très important ! Commencez toutes vos lignes par la suite de caractères ^^^^. Tout manquement à cette règle entrainera un mauvais affichage !

Les retours à la ligne dans une même phrase s’effectuent avec l’écriture \n. Exemple : ISEN \n rules affichera :

ISEN

rules

## Ajout de fonctionnalités au programme existant

L’architecture mise en place permet d’ajouter (ou de retirer) des fonctionnalités au programme Python sans affecter la totalité du programme. Des templates ont été développés afin de permettre une insertion facile de nouvelles fonctionnalités.

La façon d’ajouter de nouvelles fonctionnalités est simple (cas d’une classe simple) :

1. Copiez-collez le contenu du fichier template.py contenu dans le dossier classes.
2. Dans la boucle while not self.stoprequest.isSet(): de la fonction run, écrivez le code que vous souhaitez voir exécuter.
3. Dans le fichier controleur.py, effectuez l’importation de votre nouvelle classe (lignes 10 à 16), décommentez l’instanciation de votre classe (lignes 55 à 56), le passage de votre classe en mode non-bloquant (lignes 58 à 59) et enfin, le démarrage et l’arrêt de votre classe (Lignes 79 à 80 pour le démarrage, lignes 97-98, 115-116 et 133-134 pour l’arrêt).
4. Vous venez d’ajouter votre fonctionnalité au programme. Félicitations !

Dans le cas où vous souhaitez également pouvoir effectuer une surveillance sur une classe utilisée dans votre nouvelle fonctionnalité, un template de surveillance avec logging des informations importantes a également été développé. L’étape supplémentaire à effectuer dans ce cas consiste simplement à instancier la classe de surveillance avant la classe de traitement principal et à partager la classe à surveiller entre votre classe de traitement principal et votre classe de surveillance à l’aide du controleur (cf fonction mise\_a\_jour template dans controleur, et fonctions kill et stop dans template\_surveillance).