

---

# Projet électronique embarqué n°1 :

## Robot EDBeagle

---

### Argumentaire condensé :

POUR les passionnés de robotique

QUI SOUHAITENT éveiller les enfants à l'algorithmie et la robotique

NOTRE PRODUIT EST **un robot éducatif piloté par écran tactile**

QUI se déplace dans l'espace (avant, arrière, gauche et droite) en fonction d'un algorithme conçu par l'enfant.

CONTRAIREMENT aux produits du marché

NOTRE PRODUIT est intuitif et robuste dans son utilisation

### Persona :

Enfants, utilisateurs finaux du produit

Responsable développement du produit

### Récit utilisateur :

#### Priorité 1 :

En tant que "Responsable développement du produit", je veux avoir un environnement de développement stable et efficace basé sur : Buildroot (génération du noyau Linux, du *device-tree* et du système de fichiers) et le protocole NFS (partage du système de fichiers). Ceci permettra une reprise ultérieure du développement et un usage pour d'autres gammes de produits.

#### Priorité 2 :

En tant que "Responsable développement du produit", je veux que vous mettiez en place une liaison série RS232 entre la Beaglebone et la carte Teensy qui commande les moteurs du robot et que vous envoyiez les commandes permettant de spécifier les déplacements désirés et leur vitesse (ces commandes sont spécifiées dans le code de la carte Teensy).

#### Priorité 3 :

En tant que "Responsable développement du produit", je veux que vous développiez une interface simple sur l'écran tactile permettant de piloter les moteurs du **robot**.

#### Priorité 4 :

En tant que "Responsable développement du produit", je veux que vous conceviez une interface simple et intuitive permettant de créer un algorithme de commande du robot sur l'écran tactile. Par exemple : AV, AV, DR, AR. Cet algorithme devra être visible, afin que l'enfant puisse le corriger si le robot ne réalise pas le parcours voulu (suivi de chemin, évitement d'obstacles...). Cette interface pourra s'inspirer de l'existant.

#### Priorité 5 :

En tant que "Responsable développement du produit", je veux que vous réalisiez et testiez sur l'écran tactile l'interface conçue à la priorité précédente.

#### Priorité 6 :

En tant que "Responsable développement du produit", je veux que vous amélioriez l'interface précédente, avec des capteurs ou actionneurs externes à la carte : bouton d'arrêt d'urgence...

## **Contraintes techniques matérielles :**

Compte tenu des demandes de l'utilisateur final, je vous fournis le matériel suivant pour le développement du projet :

une platine complète comprenant :

- Une carte BeagleBone Black
- Un robot motorisé avec sa carte d'interface moteurs (Teensy 3.1) accessible via un bus I2C, USB ou RS232
- Un ensemble de capteurs ou actionneurs, disponibles à l'école

Le développement logiciel (gestion des capteurs, traitement des images infra-rouges) se fait sur la Beaglebone, la Teensy ne sert qu'à la commande et à l'asservissement des moteurs.

## **Contraintes techniques logicielles :**

Le développement logiciel doit respecter les règles de l'art en vigueur dans le monde professionnel des systèmes embarqués, à savoir. Il se fera en langage C sur une machine virtuelle Linux Debian dont on vous fournira une version « basique » à configurer (priorité 1).

Voici une liste non exhaustive des éléments à respecter :

- gestion rigoureuse de la mémoire : déclaration (locale, globale) et type des variables, allocation statique/dynamique...
- syntaxe C claire et compréhensible par une tierce personne ;
- respect de la norme C99 ;
- algorithmes pensés pour être efficient en terme de ressources (temps, énergie, mémoire...)
- commentaires du code en utilisant des balises standards pour la génération d'aide automatique (avec Doxygen par exemple) ;
- sauvegarde et suivi du code par une solution dédiée (serveur GitLab de l'école)

---

## Projet électronique embarqué n° 2 : BeagleDoom

---

### Argumentaire condensé :

POUR les passionnés de rétro-gaming  
QUI SOUHAITENT jouer en 2D ou 3D à un jeu type *doom-like* ou *FPS*  
NOTRE PRODUIT EST **une console de jeu légère, interactive, avec écran tactile**  
QUI permet à l'utilisateur de jouer à son type de jeu préféré, en 2D ou en 3D  
CONTRAIREMENT au produit du marché  
NOTRE PRODUIT est sans restriction de cartes ou de monde disponibles

### Persona :

*Gamer*

Joueur occasionnel

Nostalgique des années 90

### Récit utilisateur :

#### Priorité 1 :

En tant que Responsable développement du produit, je veux avoir un environnement de développement stable et efficace basé sur : Buildroot et NFS. Ceci permettra une reprise ultérieure du développement et un usage pour d'autres gammes de produits.

#### Priorité 2 :

En tant que Responsable développement du produit, je veux pouvoir me déplacer dans un labyrinthe 2D simple, avec une entrée, une sortie et un ensemble de murs, à l'aide de l'écran tactile.

#### Priorité 3 :

En tant que Responsable développement du produit, je veux pouvoir me déplacer dans un labyrinthe 2D simple, avec une entrée, une sortie et un ensemble de murs, **à l'aide d'un dispositif externe simple et intuitif.**

#### Priorité 4 :

En tant que Responsable développement du produit, je veux pouvoir me déplacer dans un labyrinthe 2D généré aléatoirement, à l'aide d'un dispositif externe simple et intuitif. A chaque nouvelle partie, un nouveau labyrinthe apparaît.

#### Priorité 5 :

En tant que "Responsable développement du produit", je veux bénéficier d'une expérience joueur complète : enregistrement du score, classement des plus hauts scores...

#### Priorité 6 :

En tant que Responsable développement du produit, je veux pouvoir me déplacer dans un labyrinthe 3D généré à partir d'un labyrinthe 2D, par la technique du *ray-casting*.

### Priorité 7 :

En tant que Responsable développement du produit, je veux pouvoir me déplacer dans un labyrinthe 3D peuplés de monstres et de pièges.

### **Contraintes techniques matérielles**

Compte tenu des demandes de l'utilisateur final, je vous fournis le matériel suivant pour le développement du projet :

Une carte BeagleBone Black

1 écran tactile 7 pouces

un ensemble de capteurs / actionneurs : accéléromètre, bouton poussoir, joystick...

### **Contraintes techniques logicielles :**

Le développement logiciel doit respecter les règles de l'art en vigueur dans le monde professionnel des systèmes embarqués, à savoir. Il se fera en langage C sur une machine virtuelle Linux Debian dont on vous fournira une version « basique » à configurer (priorité 1).

Voici une liste non exhaustive des éléments à respecter :

- gestion rigoureuse de la mémoire : déclaration (locale, globale) et type des variables, allocation statique/dynamique...
- syntaxe C claire et compréhensible par une tierce personne ;
- respect de la norme C99 ;
- algorithmes pensés pour être efficient en terme de ressources (temps, énergie, mémoire...)
- commentaires du code en utilisant des balises standards pour la génération d'aide automatique (avec Doxygen par exemple) ;
- sauvegarde et suivi du code par une solution dédiée (serveur GitLab de l'école)

---

## Projet n° 3 : Beaglepod

---

### Argumentaire condensé :

POUR les amateurs de musique  
QUI AIMENT regarder les clip vidéos  
NOTRE PRODUIT EST un IPOD nouvelle génération  
QUI va permettre d'écouter et visualiser ses clip préférés  
CONTRAIREMENT à l'IPOD  
NOTRE PRODUIT utilise un écran plus grand !

### Persona :

- Responsable développement produit
- Amateur de musique

### Récit Utilisateur :

Priorité 1 :

En tant que "Responsable développement du produit", je veux avoir un environnement de développement stable et efficace basé sur : Buildroot et NFS.

Priorité 2 :

En tant que "Amateur de musique", je veux pouvoir écouter un morceau de musique en affichant la photo de l'album sur l'écran

Priorité 3 :

En tant que "Responsable développement du produit" je veux pouvoir visualiser une vidéo sur l'écran

Priorité 4 :

En tant que "Amateur de musique", je veux pouvoir visualiser un clip vidéo avec sa bande sonore sur ma Beaglepod

Priorité 5 :

En tant que "Amateur de musique", je veux pouvoir tirer au hasard un clip vidéo parmi 5 et le visualiser avec sa bande sonore sur ma Beaglepod.

### Contraintes techniques matérielles

Compte tenu des demandes de l'utilisateur final, je vous fournis le matériel suivant pour le développement du projet :

Une platine complète comprenant :

Une carte BeagleBone Black

1 écran tactile 7 pouces.

1 audio cape

## **Contraintes techniques logicielles :**

Le développement logiciel doit respecter les règles de l'art en vigueur dans le monde professionnel des systèmes embarqués, à savoir. Il se fera en langage C sur une machine virtuelle Linux Debian dont on vous fournira une version « basique » à configurer (priorité 1).

Voici une liste non exhaustive des éléments à respecter :

- gestion rigoureuse de la mémoire : déclaration (locale, globale) et type des variables, allocation statique/dynamique...
- syntaxe C claire et compréhensible par une tierce personne ;
- algorithmes pensés pour être efficient en terme de ressources (temps, énergie, mémoire...)
- commentaires du code en utilisant des balises standards pour la génération d'aide automatique (avec Doxygen par exemple) ;
- sauvegarde et suivi du code par une solution dédiée (serveur GitLab de l'école, par exemple)

---

# Projet électronique embarqué n° 4

## Beagle-drone

---

### Argumentaire condensé :

POUR les pilotes en herbe

QUI SOUHAITENT « hacker » n'importe quel drone

NOTRE PRODUIT EST un auto-pilote *user-friendly* pour drone

QUI permet de contrôler un drone type quadrimoteur, en vol stationnaire

CONTRAIREMENT aux produits du marché, il s'adapte à toutes les structures mécaniques et est *open-source*

NOTRE PRODUIT utilise les capacités de communication, de calcul et d'affichage de la carte BeagleBone pour s'adapter à toutes les situations

### Persona :

Pilote en herbe

Passionné d'électronique

Fan de rétro-ingénierie

### Récit utilisateur :

#### Priorité 1 :

En tant que "Responsable développement du produit", je veux avoir un environnement de développement stable et efficace basé sur : Buildroot et NFS. Ceci permettra une reprise ultérieure du développement et un usage pour d'autres gammes de produits.

#### Priorité 2 :

En tant que "Responsable développement du produit", je veux pouvoir commander en vitesse, indépendamment, les 4 moteurs d'un drone quadricoptère.

#### Priorité 3 :

En tant que "Responsable développement du produit", je veux commander en vitesse, à l'aide d'un **écran tactile**, indépendamment, les 4 moteurs d'un drone quadricoptère.

#### Priorité 4:

En tant que "Responsable développement du produit", je veux pouvoir mesurer et afficher sur un écran les angles de roulis et de tangage d'un drone quadricoptère.

#### Priorité 5 :

En tant que "Responsable développement du produit", je veux pouvoir asservir mon drone en vol stationnaire.

#### Priorité 6 :

En tant que "Responsable développement du produit", je veux pouvoir avoir une estimation de l'altitude de mon drone, à l'aide de signaux GPS et/ou d'une centrale inertielle.

#### Priorité 7 :

En tant que "Responsable développement du produit", je veux pouvoir asservir mon drone en vol stationnaire, **à une altitude donnée.**

## **Contraintes techniques matérielles**

Compte tenu des demandes de l'utilisateur final, je vous fournis le matériel suivant pour le développement du projet :

- une carte BeagleBone Black
- un écran LCD 4" ou 7"
- une centrale inertielle 10-DOF
- un capteur GPS Garmin 10LVX

## **Contraintes techniques logicielles :**

Le développement logiciel doit respecter les règles de l'art en vigueur dans le monde professionnel des systèmes embarqués, à savoir. Il se fera en langage C sur une machine virtuelle Linux Debian dont on vous fournira une version « basique » à configurer (priorité 1).

Voici une liste non exhaustive des éléments à respecter :

- gestion rigoureuse de la mémoire : déclaration (locale, globale) et type des variables, allocation statique/dynamique...
- syntaxe C claire et compréhensible par une tierce personne ;
- respect de la norme C99 ;
- algorithmes pensés pour être efficient en terme de ressources (temps, énergie, mémoire...)
- commentaires du code en utilisant des balises standards pour la génération d'aide automatique (avec Doxygen par exemple) ;
- sauvegarde et suivi du code par une solution dédiée (serveur GitLab de l'école)

---

# Projet électronique embarqué n° 5

## Système d'apprentissage musical

---

### Argumentaire condensé :

POUR les amateurs de musique et les joueurs

QUI SOUHAITENT retrouver des sons et les placer sur une partition

NOTRE PRODUIT EST un générateur graphique de sons

QUI va permettre de produire des sons à la demande

CONTRAIREMENT aux produits habituels qui n'existent pas !

NOTRE PRODUIT utilise les capacités de stockage et de traitement des systèmes embarqués pour permettre une gestion des sons joués associé à un écran de contrôle.

### Persona :

- Responsable développement produit
- Musicien
- Joueur

### Récit utilisateur :

#### Priorité 1 :

En tant que "Responsable développement du produit", je veux avoir un environnement de développement stable et efficace basé sur : Buildroot et NFS. Ceci permettra une reprise ultérieure du développement et un usage pour d'autres gammes de produits.

#### Priorité 2 :

En tant que "Responsable développement du produit", je veux pouvoir afficher une portée musicale avec les 8 notes (do, ré, mi, fa, sol, la, si et do) sur un écran LCD 7 pouces tactile.

#### Priorité 3 :

En tant que "Responsable développement du produit", je veux pouvoir récupérer la position d'un click sur une note de la portée et l'afficher sur une fenêtre console.

#### Priorité 4 :

En tant que "Responsable développement du produit", je veux pouvoir écouter le son d'une note préenregistrée.

#### Priorité 5 :

En tant que "Musicien", je veux pouvoir « clicker » sur une note de la portée affichée sur l'écran LCD et entendre le son associé

#### Priorité 6 :

En tant que "Joueur", je veux pouvoir tirer au hasard un son, l'écouter et le reproduire en « cliquant » sur une des 8 notes de la portée.

#### Priorité 7 :

En tant que "Joueur", je veux pouvoir tirer au hasard un son, l'écouter et le reproduire en « cliquant » au bon endroit sur une portée vide.

#### Priorité 8 :

En tant que "Joueur", je veux pouvoir écouter une séquence paramétrables de sons tirés au hasard et la reproduire sur une portée vide. Un affichage du score sera à prévoir.

### **Contraintes techniques matérielles:**

Compte tenu des demandes de l'utilisateur final, je vous fournis le matériel suivant pour le développement du projet :

Deux platines complètes comprenant :

Une carte BeagleBone Black

1 écran tactile 7 pouces.

1 audio cape

### **Contraintes techniques logicielles:**

Le développement logiciel doit respecter les règles de l'art en vigueur dans le monde professionnel des systèmes embarqués, à savoir. Il se fera en langage C sur une machine virtuelle Linux Debian dont on vous fournira une version « basique » à configurer (priorité 1).

Voici une liste non exhaustive des éléments à respecter :

- gestion rigoureuse de la mémoire : déclaration (locale, globale) et type des variables, allocation statique/dynamique...
- syntaxe C claire et compréhensible par une tierce personne ;
- algorithmes pensés pour être efficient en terme de ressources (temps, énergie, mémoire...)
- commentaires du code en utilisant des balises standards pour la génération d'aide automatique (avec Doxygen par exemple) ;
- sauvegarde et suivi du code par une solution dédiée (serveur GitLab de l'école, par exemple)

# Projet électronique embarqué n° 6

## Streaming musical

### Argumentaire condensé :

POUR les mélomanes  
QUI SOUHAITENT écouter de la musique partout  
NOTRE PRODUIT EST un système de streaming  
QUI va permettre de diffuser de la musique d'un point A à un point B de leur habitation  
CONTRAIREMENT aux produits habituels qui nécessitent une connectique Bluetooth ou WIFI  
NOTRE PRODUIT utilise les capacités de conversion Analogique/Numérique et Numérique/Analogique des systèmes embarqués pour permettre s'adapter à de nombreuses interfaces anciennes ou nouvelles.

### Persona :

- Responsable développement produit
- Responsable technique
- Mélomane averti

### Récit Utilisateur :

#### Priorité 1 :

En tant que "Responsable développement du produit", je veux avoir un environnement de développement stable et efficace basé sur : Buildroot (génération du noyau Linux, du *device-tree* et du système de fichiers) et le protocole NFS (partage du système de fichiers). Ceci permettra une reprise ultérieure du développement et un usage pour d'autres gammes de produits.

#### Priorité 2 :

En tant que "Responsable technique", je veux une analyse détaillée de la bibliothèque ALSA et de son utilisation, en ligne de commande ou dans un programme C

#### Priorité 3 :

En tant que "Responsable technique", je veux pouvoir utiliser la carte beaglebone pour faire l'acquisition d'un signal audio

#### Priorité 4 :

En tant que "Responsable technique", je veux pouvoir utiliser la carte beaglebone pour un signal audio et le transmettre à un système d'enceintes

#### Priorité 5 :

En tant que "Responsable technique", je veux une analyse détaillée des protocoles de communication entre 2 cartes beaglebone, avec une attention particulière sur la notion de *socket*

#### Priorité 6 :

En tant que „Responsable développement produit“, je veux la démonstration de la communication d'un morceau de musique acquis à partir d'une carte A vers une carte B (qui le lira)

#### Priorité 7 :

En tant que "Mélomane averti", je veux une qualité de son supérieur et respectant les normes en vigueur

Priorité 8 :

En tant que "Mélomane averti", je veux une interface utilisateur simple et conviviale, regroupant les fonctions suivantes, côté émetteur :

- Lecture du morceau
- Arrêt du morceau
- Reglage du volume sonore
- Spectre du morceau, sur une plage de quelques secondes

**Contraintes techniques matérielles:**

Compte tenu des demandes de l'utilisateur final, je vous fournis le matériel suivant pour le développement du projet :

une platine complète comprenant :

- une carte BeagleBone Black
- une carte audioCape
- un écran 7 pouces résistif

**Contraintes techniques logicielles :**

Le développement logiciel doit respecter les règles de l'art en vigueur dans le monde professionnel des systèmes embarqués, à savoir. Il se fera en langage C sur une machine virtuelle Linux Debian dont on vous fournira une version « basique » à configurer (priorité 1).

Voici une liste non exhaustive des éléments à respecter :

- gestion rigoureuse de la mémoire : déclaration (locale, globale) et type des variables, allocation statique/dynamique...
- syntaxe C claire et compréhensible par une tierce personne ;
- algorithmes pensés pour être efficient en terme de ressources (temps, énergie, mémoire...)
- commentaires du code en utilisant des balises standards pour la génération d'aide automatique (avec Doxygen par exemple) ;
- sauvegarde et suivi du code par une solution dédiée (serveur GitLab de l'école, par exemple)