

**Epreuve E6 – Projet 2025**

**Kit d’arbitrage et suivi des performances des joueurs**

Antoine DONNART spécialité Informatique et Réseaux

BTS CIEL IR

La Croix Rouge La Salle – Brest

Donneur d’ordre : Carole PAUREAU

P2025: Kit d’arbitrage

*DONNART Antoine*

*IR*

Dossier technique du projet - partie individuelle

[1 - Situation dans le projet 2](#_Toc1830946174)

[1.1 - Synoptique de la réalisation 2](#_Toc954267580)

[1.2 - Description de la partie personnelle 2](#_Toc287244141)

[2 - Réalisation de la fonction et/ou cas d'utilisation xxx 2](#_Toc1565573444)

[2.1 - Conception détaillée 3](#_Toc6176967)

[2.2 - Tests unitaires 3](#_Toc1577043585)

[2.2.1 - Test unitaire du module matériel/logiciel xxx 3](#_Toc1396512167)

[2.2.2 - Problèmes rencontrés 4](#_Toc141572033)

[3 - Bilan de la réalisation personnelle 4](#_Toc1984807830)

1. Situation dans le projet
   1. Synoptique de la réalisation

Le Tennis Club Brestois, fondé en 1950, était à l'origine un club privé réservé aux membres par parrainage. Son essor a été marqué par la victoire de Yannick Noah en 1983, la participation aux Internationaux de Brest dans les années 1990 et l’organisation du dernier Open 2024.

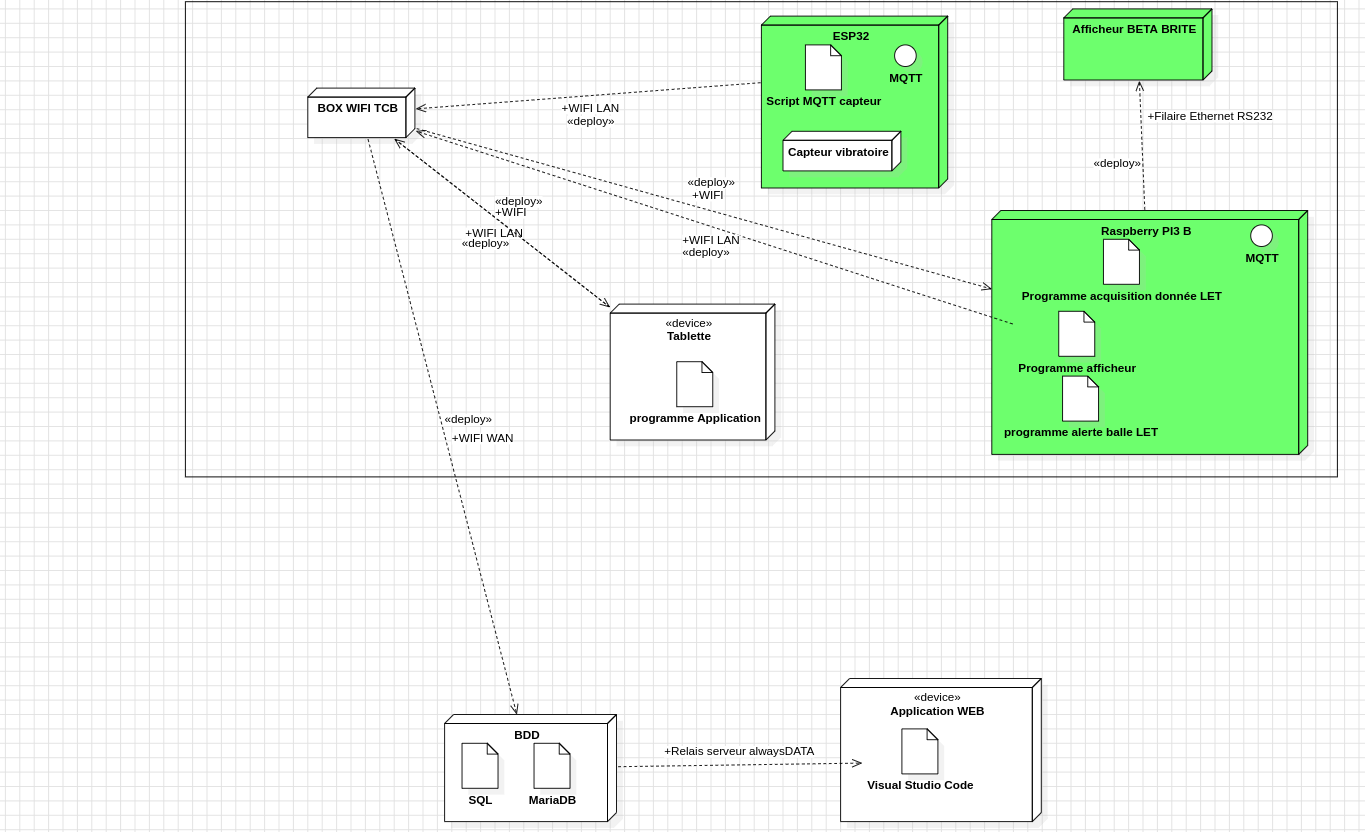
Dans le cadre de notre BTS CIEL, nous devons développer une application de suivi des performances des joueurs ainsi qu’une solution d’arbitrage portative intégrant un affichage et une détection de certains paramètres du jeu, modernisant ainsi les outils d’analyse et d’arbitrage du club.

Ma partie du projet TCB se nomme : *Kit arbitrage – acquisition et affichage, j’avais pour objectif de :*

* Réaliser une solution d’affichage des résultats en temps réel pour les joueurs et le public
* Choisir l’afficheur
* Mesurer la vitesse de la balle au service ou détecter une balle “let”
* Configurer le broker

Après beaucoup de discussion avec mes camarades de projet et avec l’enseignante, gérant principalement ce projet, il a été convenu que pour ma partie je dois :

* Détecter une balle “let”
* Configurer le broker
* Réaliser une solution d’affichage des résultats en temps réel pour les joueurs et le public
* Choisir matériel (capteur LET, Afficheur, BROKER)



* 1. Description de la partie personnelle

L’objectif principal de ma partie du projet est de permettre à l’arbitre d’afficher correctement le score du match. Un objectif secondaire consiste à détecter les balles LET et à envoyer une alerte sur l’application tablette de l’arbitre, lui permettant ainsi de valider ou non l’action.

Afin d’assurer une bonne compréhension du projet et de ma contribution, j’ai préparé ce dossier individuel et élaboré plusieurs diagrammes pour structurer la documentation.

Concernant mon environnement de travail, j’utilise principalement Python sur une Raspberry Pi et C++ sur le système arduino IDE, Le matériel disponible au sein de la Croix-Rouge La Salle répondait déjà aux besoins du projet. En effet, elle disposait d’un Raspberry Pi 3B ainsi que d’un afficheur LED BETA Brite, paramétrable et compatible avec notre usage, cependant le centre de formation ne disposait pas du matériel pour répondre à la demande pour capter les balles LET, une liste de matériel à donc été mis en place.

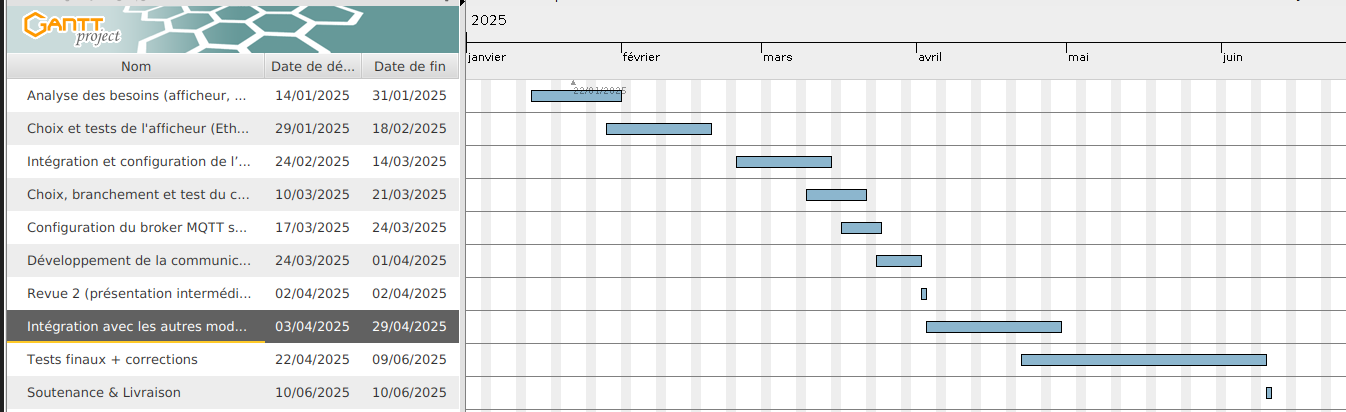
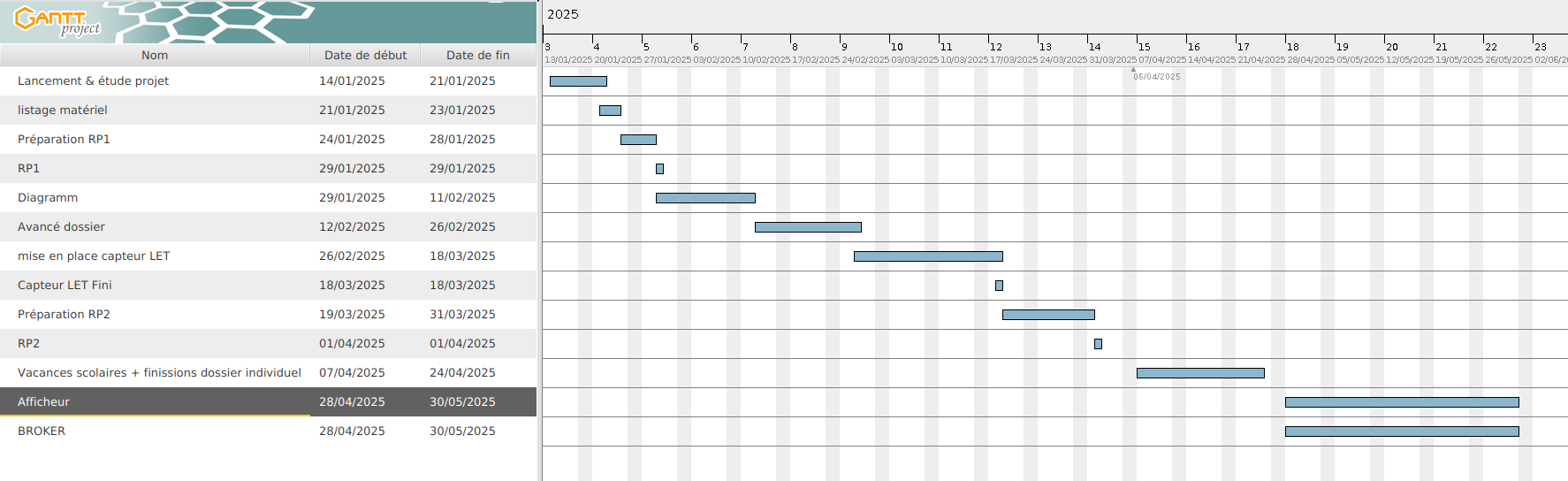
En ce qui concerne le capteur LET, aucun choix précis n’avait été fait au départ. Pour donner suite aux recommandations de mon professeur d’Électronique et Réseau (ER), j’ai opté pour un capteur vibratoire **101020031**, associé à une nappe fonctionnant comme une résistance variable en fonction de sa flexion.

Pour la transmission des données, nous avons décidé d’utiliser un microcontrôleur ESP-12E basé sur l’ESP8266, fonctionnant sur batterie, puis suite à certains détails je suis parti sur un EPS 32-WROOM.

Enfin, bien que l’afficheur BETA Brite puisse être configuré via une application dédiée, un utilisateur GitHub a mis à disposition une documentation et des programmes permettant de le paramétrer directement. La connexion avec la Raspberry Pi se fait via le port Ethernet.

Pour visualiser un ensemble plus précis de ma partie du projet, voici un diagramme de cas d’utilisation et d’exigences (ma partie est surligné en jaune).

Pour suivre notre quotidien sur le projet nous avons mis en place un tableau de bord individuel partagé et j’ai mis en place un diagramme de Qantt prévisonnelle et un diagramme de Qantt actuel



*Pour la suite, on sélectionnera quelques tâches professionnelles significatives de la réalisation.*

1. Réalisation de la fonction et/ou cas d'utilisation xxx

* *La description des réalisations des tâches professionnelles varie en fonction de la nature de celles-ci (développement logiciel, installation réseau, réalisation de CI, etc.) :*
  + *Présenter succinctement la tâche.*
  + *Justifiez un choix technologique retenu pour la réalisation de la tâche professionnelle (choix des composants électroniques, choix d’un actif réseau, choix d’une bibliothèque logicielle, etc.).*
  1. Conception détaillée
* *Pour une tâche d'électronique, on pourra trouver :*
  + *Un ou plusieurs diagrammes de blocs internes ;*
  + *Un schéma structurel ;*
  + *La justification des choix des composants ;*
  + *Chronogrammes ;*
  + *Routage (documents de fabrication) ;*
  + *etc.*
* *Pour une tâche de développement logiciel, on pourra trouver :*
  + *Un ou plusieurs diagrammes de classe simplifiés où l'on peut voir toutes vos classes (avec les principales méthodes et attributs) et les classes des autres développeurs en relation avec les vôtres (uniquement le nom de la classe).*
  + *Une description brève des classes importantes et la nature des associations avec les autres classes.*
  + *Une description des données traitées :*
    - *les formats, types, plage de validité,*
    - *les structures des fichiers traités (xml, etc.),*
    - *les structures détaillées des bases de données.*
  + *Les IHM*
  + *Les concepts mathématiques utilisés dans les classes développées (FFT, filtrage, ...).*
  + *Une documentation technique de la classe réalisée (algorithme d’une méthode, diagramme de séquence ou diagramme d’état, méthodes de l'API utilisées, etc. ) en fonction de sa complexité.*
  + *Un extrait significatif du code source commenté de la classe (le reste du code source est donné sous format électronique).*
* *Pour une configuration/installation réseau ou de services, on pourra trouver :*
  + *les procédures d’installation des équipements réseaux et/ou des services,*
  + *les fichiers de configuration des équipements du réseau,*
  + *les fichiers de configuration des services,*
  + *les structures des trames échangées ainsi que le protocole de communication utilisé,*
  + *Etc.*

*BROKER*

*CAPTEUR*   
*Afficheur V1 +V2*

*Programme envoie vers application*

sous Linux avec une interface graphique. Une amélioration envisageable serait de passer à un environnement en ligne de commande pour optimiser les performances. Pour exécuter mes programmes, je crée un fichier en **.py**, puis je l’exécute via le terminal avec la commande :

**sudo python3 programme.py**

* 1. Tests unitaires
* *Présentez le récapitulatif chronologique de vos tests unitaires mis en œuvre pour la validation de la tâche professionnelle et précisez les tests unitaires que vous allez détailler par la suite.*
  + 1. Test unitaire du module matériel/logiciel xxx
* *Présentez l'identification du test unitaire (nom, numéro, date, etc.).*
* *Présentez l'objectif du test, éventuellement le cas d'utilisation ainsi que les divers scénarios faisant intervenir le matériel testé et si nécessaire un synoptique décrivant l'environnement matériel et/ou logiciel spécifique nécessaire à la mise en œuvre du test.*
* *Présentez la procédure de test.*
* *Présentez les résultats attendus et le rapport d'exécution des tests unitaires (conformité/non conformité).*
* *On pourra compléter un tableau type :*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Élément testé : | | ESP32 | | | | | | |
| Objectif du test : | | Vérifier l’envoie sur le broker | | | | | | |
| Nom du testeur : | | Teste ESP>BROKER | | | Date :1/04/2025 | |  | |
| Moyens mis en œuvre : | | Logiciel :  Arduino IDE  Mosquitto  Paho MQTT | | Matériel :  ESP 32 + capteur vibratoire + batterie 3,7V Raspberry PI3B | | Outil de développement :  C++ Python | | |
| Procédure du test : | | | | | | | | |
| Id | Description du vecteur de test | | Résultat attendu | | Résultat obtenu | | | Validation (O/N) |
| 1 | Voir si depuis la raspberry nous pouvons voir si etat haut du capteur | | Le programme se lance et nous voyons quand etat haut | | Le programme ne se lance pas | | | N |
| 2 | Après réinstallation de PAHO MQTT et vérification de mosquitto | | Le programme se lance et nous voyons quand état haut | | Le programme ne se lance pas mais affiche un autre problème non lié a PAHO MQTT | | | N |
| 3 | Après avoir changer l’adresse ip pour l’écoute et l’envoi, je réeffectue le test | | Le programme se lance et nous voyons quand etat haut | | Le programme permet bien de voir si état haut même si esp est sur battery | | | O |
|  |  | |  | |  | | |  |
|  |  | |  | |  | | |  |
|  |  | |  | |  | | |  |
| Conclusion du test : | | Il faut faire attention à l’adresse IP fourni par le DHCP et prévoir ce problème en fonction de cela | | | | | | |

* + 1. Problèmes rencontrés
* *Présentez un historique des problèmes rencontrés pendant la phase de tests unitaires,*
* *Présentez les solutions mises en œuvre pour remédier aux problèmes rencontrés.*

1. Bilan de la réalisation personnelle

* *Présentez les points du projet qui ont été validés.*
* *Précisez les parties restant à développer.*
* *Indiquez, si nécessaire, les améliorations possibles.*
* *Proposez une conclusion personnelle.*