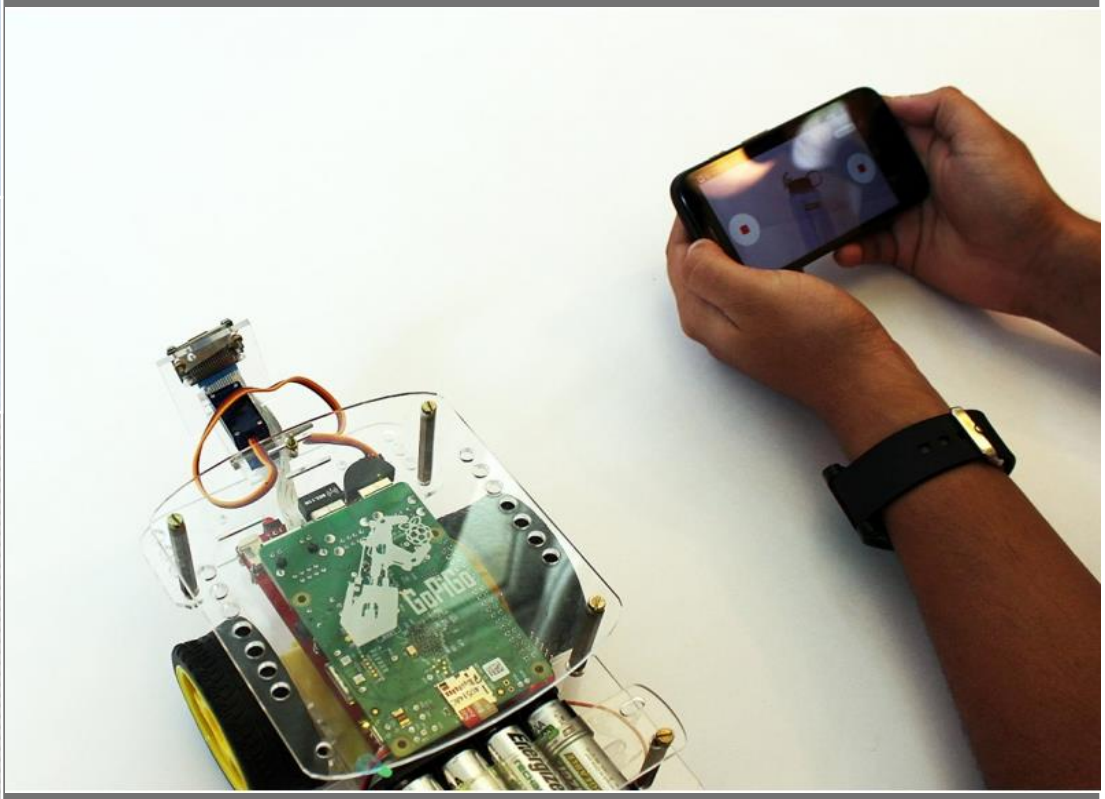
UNIVERSITÉ PARIS 8
VINCENNES-SAINT-DENIS

Rapport du projet GOPIGO

Base de données embarquées



Amel REKKAB & Nadir HAMMOUM
Le 30/12/2016

SOMMAIRE

INTRODUCTION

Ce présent travail s'inscrit dans le cadre du Module « Base de données embarquées ».

Le marché de la téléphonie portable connait actuellement une véritable révolution, d'un simple téléphone portable pour émettre des appels à un téléphone évolué doté de capacités proches d'un véritable ordinateur appelé Smartphone.

Le projet a pour objectif de réaliser une application qui permet de piloter un Robot GoPiGo à distance tout en tirant profit des gadgets que possèdent les Smartphones de nos jours. Mis à part le développement proprement dit de l'application, la première étape consistait choisir les outils conviviaux et envisageables à l'aboutissement du projet. Par la suite, nous avons entamé la modélisation et le développement de l'application.

Ce rapport s'articule autour de 2 axes principaux :

- La première étape était l'étude préalable de l'application, c'est à dire définir le contexte et le choix des outils à utiliser pour le développement.
- La deuxième étape est la présentation des différentes étapes de réalisation de notre application.

Mais, avant de vous parler plus en détail de notre travail et du développement de l'application, il nous semble important de vous présenter les outils utilisés afin d'aboutir à ce projet.

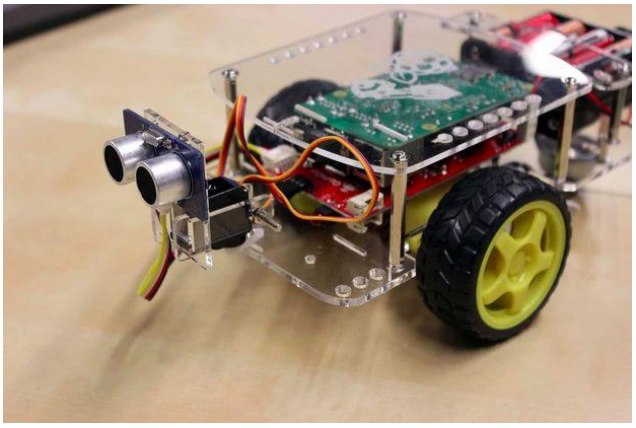
1. PRESENTATION DES OUTILS

1.1 Kit Robotique GoPiGo

Le Kit Robotique GoPiGo est un robot complet pour Raspberry Pi. Le robot possède un châssis robot, des moteurs, des commandes, et une alimentation robuste. Avec tant de langages de programmation et d'accessoires USB disponibles pour le Pi.

Le kit de démarrage comprend :

- Kit GoPiGo2 Base.
- Rasberry Pi 3.
- GoPiGo Servo Package.
- Mini Wifi dongle.
- Capteur a ultrasons.
- Carte micro SD avec un logiciel. personnalisé Dexter Industries.
- Alimentation.
- Câble Ethernet.



1.2 Rasberry PI3

Le **Raspberry Pi** est un nano-ordinateur mono carte à processeur ARM et permet l'exécution de plusieurs variantes de systèmes d'exploitation libre GNU/Linux et de logiciels compatibles.

Le Raspberry Pi 3 est la 3 eme génération Raspberry Pi,

Il est doté de :

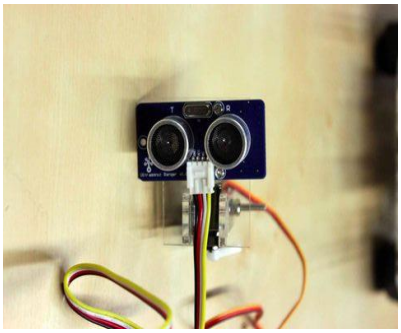
- 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)

Tout comme Raspberry Pi2, il est doté de :

- 1Go de RAM
- 4 ports USB
- 40 broches GPIO
- Port HDMI
- Port Ethernet
- jack 3,5 mm audio combiné et vidéo composite

- Interface de l'appareil photo (CSI)
- Interface d'affichage (DSI)
- fente pour carte Micro SD
- cœur graphique VideoCore IV 3D

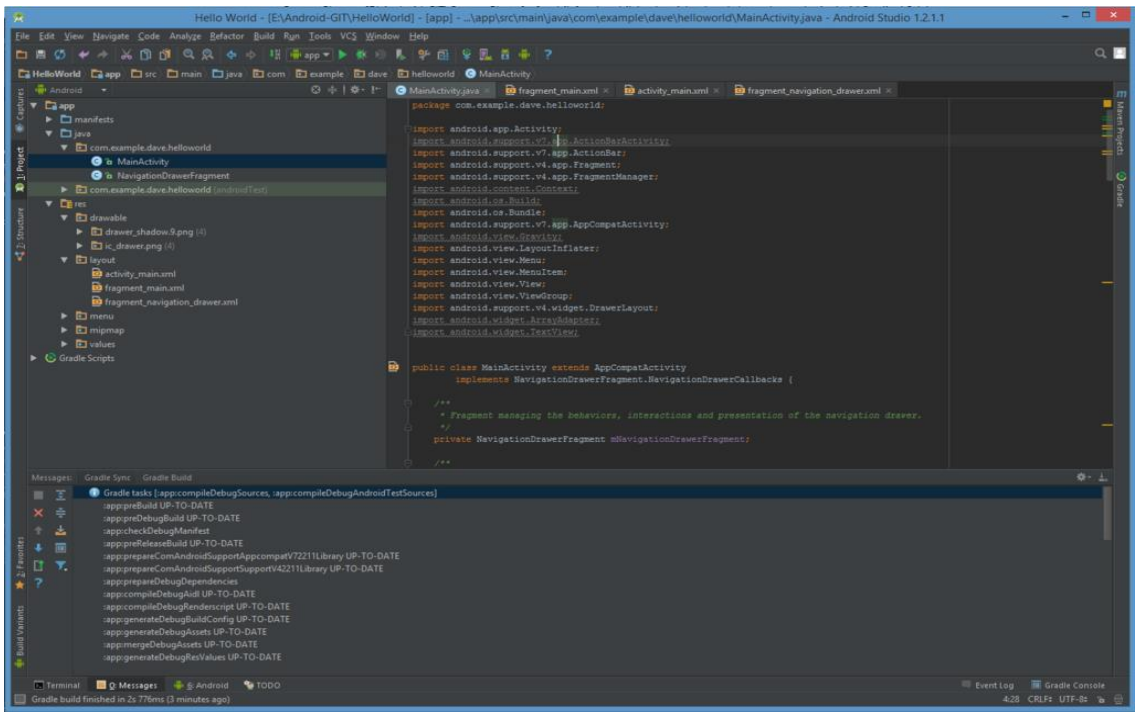
1.3 Capteur Ultra Son



Le capteur Ultrason est le capteur le plus utilisé en robotique, il émet de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son.
Elles se réfléchissent par l'objet à détecter et reviennent sous forme d'écho au capteur.
Celui-ci calcule alors la distance le séparant de la cible sur la base du temps nécessaire à l'impulsion pour aller du capteur à l'objet et revenir.

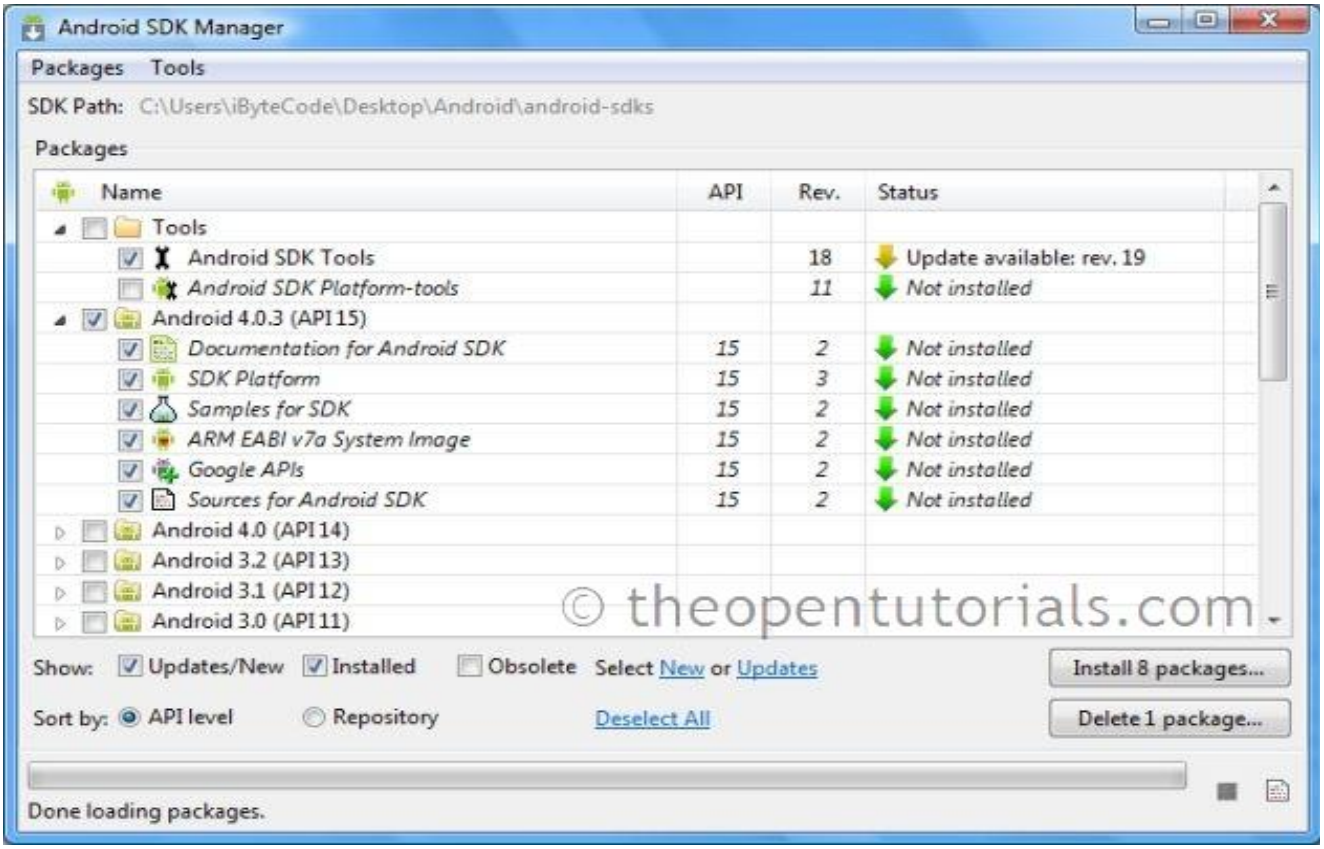
1.4 Android Studio

Android Studio est un environnement de développement qui permet de réaliser des projets sur différents types de support,tablette ou smartphone. il s'articule autour d'un éditeur reposant sur la technologie Open Source de développement Java IntelliJ IDEA.
Il intègre notamment refactoring, colorisation syntaxique, moteur d'analyse de code, et aide à la gestion des appels d'APIs Android.



1.5 SDK

Software Development Kit est un ensemble d'outils logiciels permettant de faciliter le développement d'un logiciel sur une plateforme donnée (par exemple, il existe un SDK pour Android, un pour iOS, etc.)
Le SDK d'Android est multi- plateforme, permettant de tester son application sur plusieurs versions différentes d'Android, plusieurs tailles d'écran, etc. et ce, même si on n'a pas d'appareil physique.



1.6 BDD SQLite

Android dispose d'une base de données relationnelle basée sur SQLite qui fournit un moyen efficace de gérer une petite quantité de données.

Pour réaliser des écritures ou lectures, on utilise les méthodes getWritableDatabase() et getReadableDatabase() qui renvoient une instance de SQLiteDatabase. Il existe plusieurs méthodes pour traiter la réponse (lecture/écriture), on trouve :

- **getCount() :** nombre de lignes de la réponse
- **moveToFirst() :** déplace le curseur de réponse à la première ligne
- **getInt(int columnIndex):** retourne la valeur (int) de la colonne passée en paramètre
- **getString(int columnIndex)**: retourne la valeur (String) de la colonne passée en paramètre

- **moveToNext() :** avance à la ligne suivante
- **getColumnName(int) :** donne le nom de la colonne désignée par l'index

1.7 LANGAGES DE DEVELOPPEMENT

1.7.1 JAVA

Java est un langage de programmation moderne orienté objet qui permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation. Java donne aussi la possibilité de développer des programmes pour smartphones et assistants personnels. Enfin, le langage java peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur.

1.7.2 XML

La mise en page des données est assurée par un langage de mise en page tiers. Il existe trois solutions pour mettre en forme un document XML :
CSS : la solution la plus utilisée actuellement, étant donné qu'il s'agit d'un standard qui a déjà fait ses preuves avec HTML
XSL : un langage de feuilles de style extensible développé spécialement pour XML. Toutefois, ce nouveau langage n'est pas reconnu pour l'instant comme un standard officiel
XSLT : permet de transformer un document XML en document HTML accompagné de feuilles de style.

2. REALISATION

2.1 COMPOSANTS D'UNE APPLICATION ANDROID

Activities(android.app.Activity) : Une partie de l'application représentant une vue a l'utilisateur.
Service (android.app.Service) : permet l'exécution d'un algorithme sur un temps indéfini. Il ne s'arrêtera que lorsque la tâche est finie ou que son exécution est arrêtée.
Il peut être lancé à différents moments :

- Au démarrage du téléphone.
- Au moment d'un événement
- Lancement de votre application.
- Action particulière dans votre application.

Broadcast and Intent Receivers :
Permet d'écouter ce qui se passe sur l'application et déclencher une action que vous aurez prédéfinie. C'est souvent par ce mécanisme que les services sont lancés.
Content providers
Servent à accéder à des données depuis l'application. Comme :

- Aux contacts stockés dans le téléphone.
- A l'agenda.
- Aux photos.

2.2 CYCLE DE VIE D'UNE APPLICATION ANDROID

Les Smartphones diffèrent des ordinateurs classiques par cycle de vie d'une application. Une application Android est composée d'une ou plusieurs activités qui sont la base d'un composant pour la création d'interfaces utilisateur.

onCreate() : Cette méthode est appelée à la création de l'activité. Elle sert à initialiser l'activité ainsi que toutes les données nécessaires à cette dernière.

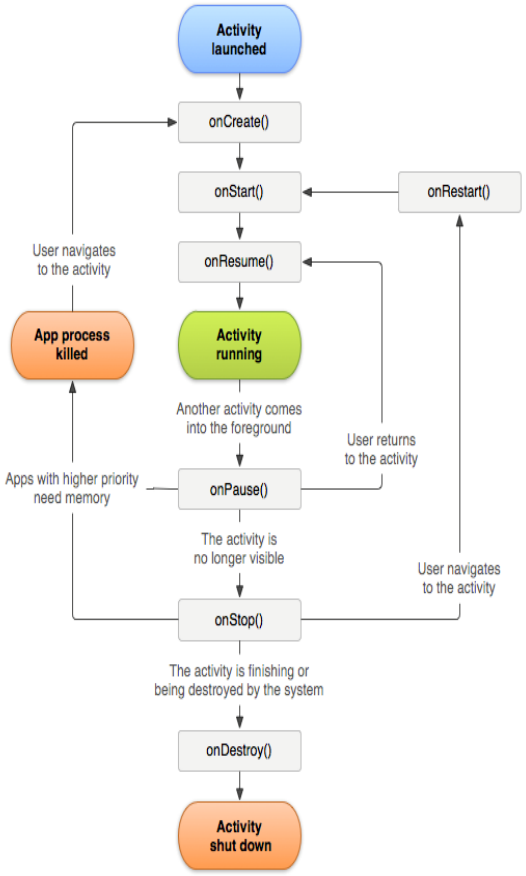
onStart() : Cette méthode est exécuté après chaque onCreate() ou onRestart() et elle fait le chargement des données sauvegardées durant le dernier arrêt.

onStop() : Cette méthode est exécuté avant chaque mise en sommeil et est exécuté avant chaque onDestroy, celle-ci permet la libération des ressources.

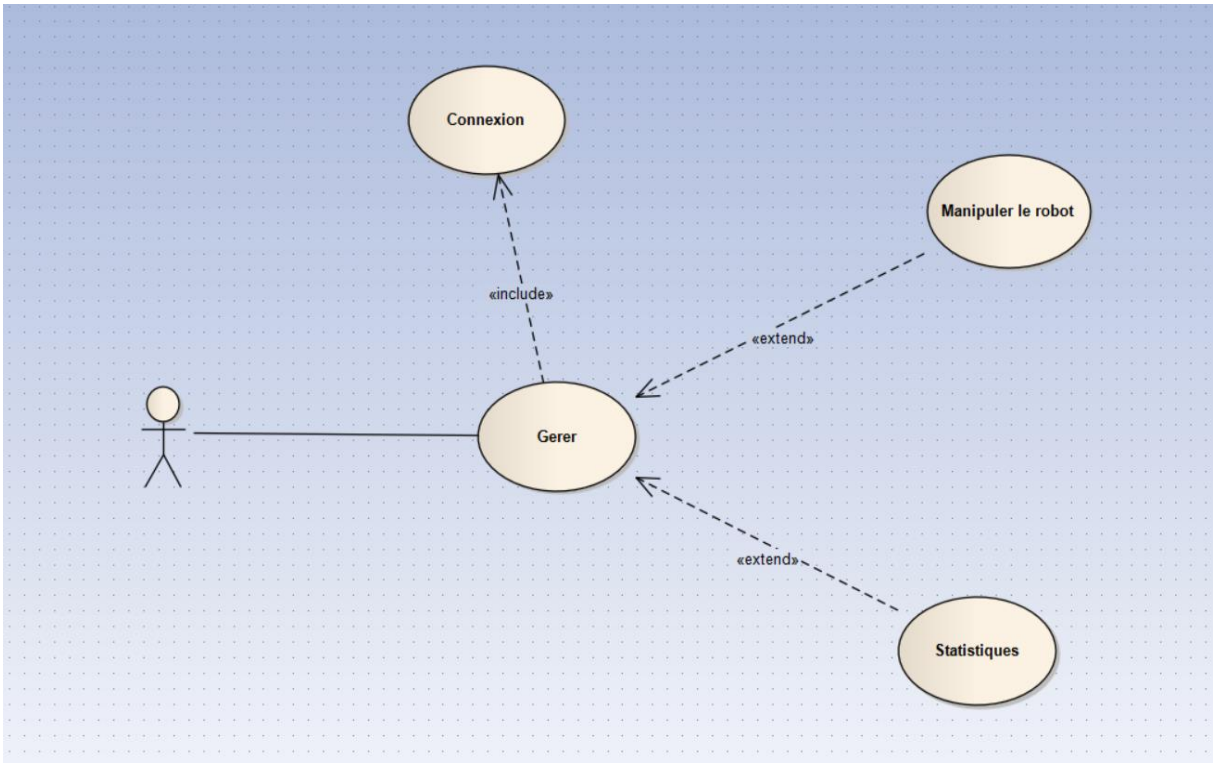
onResume() : Cette méthode est appelé quand l'application passe en arrière-plan.

onPause() : Cette méthode est appelé quand l'application passe en arrière-plan et qu'une autre application se met devant.

onDestroy() : Appelée quand l'application est totalement fermée. Toutes les données non sauvegardées sont perdues.

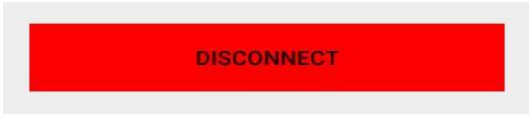


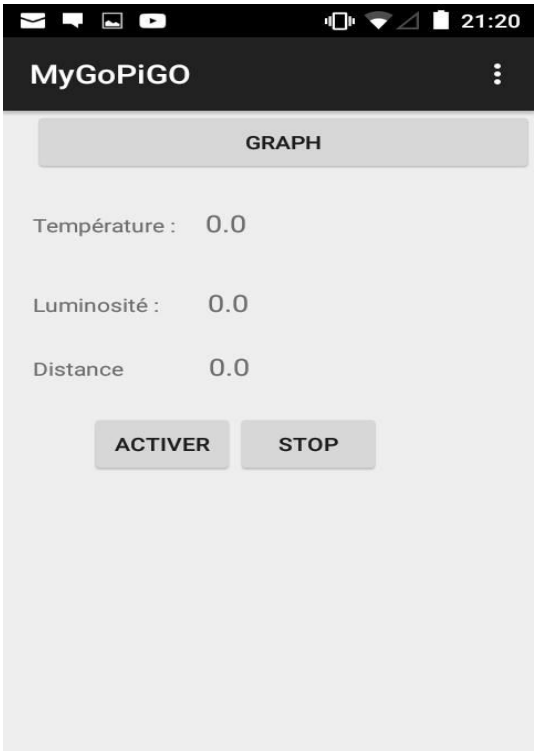
2.3 DIAGRAMME DE CAS D'UTILISATION



2.4 QUELQUES PRISES D'ECRAN

Cette interface représente l'interface d'accueil, Au premier lancement de l'application, cette interface s'affiche avec un menu permettant de surfer dans l'application tel que l'affichage des statistiques, piloter le robot GoPiGo(en le faisant avancer, reculer et tourner vers la gauche ou vers la droite), le bouton de connexion permet d'ouvrir une connexion WIFI avec le robot.
Une fois connecter un bouton de déconnexion apparait qui permettra de se déconnecter du robot.

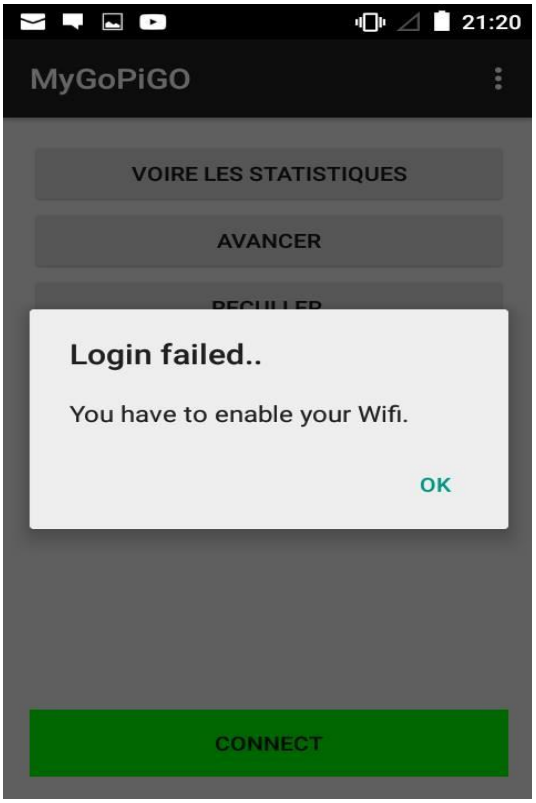




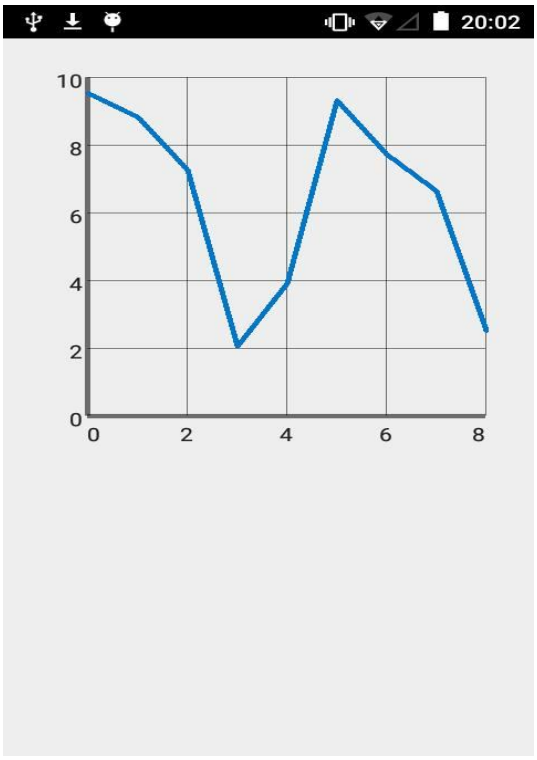
Cette interface apparait lorsqu'on accède aux statistiques, celles-ci permet d'afficher les données récupérées par l'ordinateur a partir de ses capteurs qui sont la température, la luminosité et la distance par rapport à un obstacle,

Deux boutons sont nécessaires pour gérer cet affichage, le bouton « ACTIVER » permet de démarrer l'affichage des données et le bouton « STOP » permet d'arrêter l'affichage.

Un message d'erreur s'affiche sur l'écran si l'utilisateur désire afficher les données des capteurs sans établir une connexion WIFI avec le robot GoPiGo.



Cette interface permet d'afficher à l'utilisateur les données reçues du robot GoPiGo sous forme graphique.



CONCLUSION

A travers ce projet nous avons été chargés de réaliser une application sur une plateforme mobile qui permet de piloter un robot GoPiGo, celle-ci répond en temps réel aux exigences de rapidité et de robustesse des résultats.

Ce travail nous a permis de découvrir une nouvelle plateforme de développement et à enrichir notre savoir et notre expérience dans le domaine de l'embarqué.

Dans la première partie nous avons fait la présentation des outils qui nous ont permis d'aboutir à une application fonctionnelle, et la seconde partie a été consacrée à la réalisation de l'application.