

Fauvarque Florian
RICM4



Rapport de stage 4^{ème} année

**Réalisation d'une application de
visualisation de données sur Android**

Tome principal

Année universitaire 2010 – 2011

7 Juin 2010 – 27 Août 2010

Je tiens à remercier toute les personnes du service SED, qui m'ont accueilli au sein de leur équipe lors de ce stage, et plus spécialement : Christophe Brailon mon tuteur, Guillaume Roche et Sandrine Avakian pour leur aide et leurs conseils, ainsi que mes collègues stagiaires Jocelyn Gate et Eric Chaloin pour leur bonne humeur qu'ils ont partagée avec moi tous les jours dans la halle robotique.

Table des matières

1	Introduction	5
2	Contexte du stage	5
2.1	Présentation de l'INRIA	5
2.1.1	L'INRIA, l'institut français de recherche en informatique et automatique	5
2.1.2	L'INRIA Grenoble – Rhône Alpes	7
2.1.3	Le service SED	7
2.2	Les réseaux de capteurs	8
2.3	Le projet ESPAD	8
2.3.1	Présentation et objectifs du projet	8
2.3.2	Choix scientifiques et techniques	9
2.3.2.1	Plateforme utilisée	9
2.3.2.2	Principe de fonctionnement	9
2.3.2.3	Infrastructure logicielle	10
2.3.3	Le cas d'utilisation de l'aviron	11
2.3.3.1	Scénario	11
2.3.3.2	Affichage	12
2.3.3.3	Equipement	14
2.4	La plateforme Android	14
3	Travail réalisé	15
3.1	Description du sujet	15
3.2	Gestion de projet	15
3.2.1	Outils utilisés	15
3.2.2	Diagramme de Gantt	17
3.2.3	Gestion des risques	18
3.3	Les différentes tâches	19
3.3.1	Mise à jour du téléphone	19
3.3.2	Le protocole de communication	19
3.3.2.1	Le protocole Bluetooth et le profil SPP	19
3.3.2.2	Le protocole Bluetooth dans Android	20
3.3.2.3	La JSR 82 et son implémentation avec la librairie Bluecove	20
3.3.2.4	La communication entre le téléphone Android et le nœud contrôleur ..	22

3.3.3	Etats de l'application et menus dynamiques	23
3.3.4	Visualisation des données sous forme de graphes	24
3.4	Quelques problèmes rencontrés	25
3.5	Ce qu'il reste à faire.....	26
4	Bilan et conclusion	27
5	Annexes.....	28
5.1	Protocole de communication	28
5.2	Captures d'écran de l'application	31
6	Bibliographie / Webographie	33
7	Liste des documents rédigés pendant le stage	34
8	Table des révisions du SVN	35
9	Résumé	36
9.1	Résumé en Français	36
9.2	Abstract in English	36

Table des illustrations

Illustration 1	: Réseau des partenaires académiques et industriels.....	6
Illustration 2	: Architecture du réseau de capteurs	10
Illustration 3	: Parties d'un aviron	11
Illustration 4	: Rapport Sonar au 26/08/2010	16
Illustration 5	: Architecture de la stack Bluetooth de la librairie Bluecove	21
Illustration 6	: Graphe d'état transition de la classe SensorsDatasVisualiser.java.....	23
Illustration 7	: Exemple d'application Android utilisant la librairie AChartEngine	24
Illustration 8	: Exemple d'interface possible pour le cas d'utilisation de l'aviron	26

1 Introduction

Les progrès conjoints de la microélectronique, de la microtechnologie, des technologies de transmission sans fil et des applications logicielles ont permis de produire à coût raisonnable des micro-capteurs de faible volume, susceptibles de fonctionner en réseaux.

Le projet le projet ESPAD consiste ainsi à expérimenter des réseaux de capteurs sans fil pour les sportifs souhaitant avoir une meilleure compréhension de leurs performances.

Durant ce stage, il m'a été demandé de développer une application Android de visualisation de données d'un réseau de capteurs dans le cadre du projet ESPAD.

Dans ce document, je vous présenterai dans un premier temps l'institut de recherche INRIA, puis le projet ESPAD et la plateforme Android. Dans un deuxième temps je présenterai mon projet et son déroulement. Pour finir, je dresserai un bilan de ce stage.

2 Contexte du stage

2.1 Présentation de l'INRIA

2.1.1 L'INRIA, l'institut français de recherche en informatique et automatique

L'INRIA (Institut National de Recherche en Informatique et en Automatique), sous la double tutelle des ministères de la recherche et de l'industrie, est l'institut français de recherche publique en Sciences et Technologies de l'Information et de la Communication.

Fondé en 1967, l'institut accompagne depuis plus de 40 ans les mutations économiques et sociales liées à la diffusion des technologies numériques. Il mène une activité de recherche fondamentale et de développement technologique au rayonnement international, dans les domaines suivants :

- Modélisation, simulation et optimisation
- Programmation
- Communication, information et calculs ubiquitaires
- Interaction avec des mondes réels ou virtuels
- Ingénierie numérique
- Sciences numériques
- Médecine numérique

Le budget de l'institut est de 217 millions d'euros, dont 21% de ressources externes. Environ 4100 personnes dont 3150 scientifiques y travaillent, répartis dans 174 équipes-projets dans les 8 centres de recherche en France situés à Lille, Paris, Nancy, Saclay, Rennes, Bordeaux, Sophia Antipolis et Grenoble.

L'institut a de nombreux partenariats scientifiques, académiques et industriels.



Illustration 1 : Réseau des partenaires académiques et industriels

En effet, quelques équipes-projets sont communes avec des grandes écoles, des universités, ou encore d'autres organismes de recherche. Beaucoup de chercheurs de l'INRIA sont aussi enseignants, et l'institut propose chaque année de nombreux stages et sujets de thèse.

L'INRIA, avec plus de 800 contrats de recherche actifs, a une longue expérience en matière de relations avec l'industrie, que ce soit de grandes entreprises tels que ST Microelectronics ou Orange, ou des PME. Avec son incubateur INRIA Transfert, l'institut favorise et accompagne la création d'entreprise dans le but de permettre le transfert technologique. C'est en tout 98 entreprises innovantes dans le domaine de l'informatique qui ont été ainsi créées, dont plus de 50 encore en activité.

Enfin, l'INRIA entretient aussi de nombreuses relations internationales. En effet, l'institut participe à plus de 180 projets européens dans le cadre des 6^e et 7^e PCRD (*Programme-cadre de Recherche et Développement*). Il est aussi le membre fondateur de l'ERCIM (*European Research Consortium for Informatics and Mathematics*), un consortium de pays d'Europe visant à stimuler la collaboration et la coopération entre les communautés de la recherche et l'industrie européenne.

L'institut accueille plus de 1800 scientifiques étrangers, et a même un laboratoire conjoint avec la Chine, le LIAMA (*Laboratoire franco-chinois d'Informatique, d'Automatique et de Mathématiques Appliquées*).

2.1.2 L'INRIA Grenoble – Rhône Alpes

Créé en 1992 dans l'Inovallée à Montbonnot, c'est dans un contexte régional particulièrement porteur pour les nouvelles technologies que s'est développé le centre INRIA Grenoble - Rhône-Alpes.

Il comporte aujourd'hui près de 650 personnes, dont environ 190 chercheurs et enseignants-chercheurs, 200 ingénieurs, techniciens et administratifs, et 220 doctorants (dont 45% étrangers), répartis entre 33 équipes de recherche et 8 services de support à la recherche.

Son budget en 2008 était de 17 millions d'euros, dont 6,4 millions de ressources contractuelles.

En tout, 19 entreprises innovantes ont été créées à partir des recherches de l'INRIA Grenoble et de ses partenaires, ce qui a généré plus de 600 emplois.

L'INRIA Grenoble - Rhône-Alpes est ainsi leader dans la recherche de plateformes expérimentales en robotique, en réseaux de grille de calcul (GRID5000) ou encore en réalité virtuelle et réalité augmentée.

2.1.3 Le service SED

Les services Support Expérimentation et Développement logiciel, appelés SED, sont présents dans tous les centres de recherche INRIA. Leurs missions sont identiques et se déclinent selon 3 axes :

1. maintenir un réseau d'expertises pour diffuser les bonnes pratiques de développement logiciel et l'utilisation d'outils communautaires au sein des équipes-projets
2. mettre en place, développer et maintenir les plateformes expérimentales avec les équipes-projets
3. participer aux développements logiciels, notamment aux ADT (Actions de Développement Technologique) au sein des équipes-projets

Composé de 12 personnes, le service s'occupe notamment du support des outils utilisés par les chercheurs tel que Gforge, une forge de logiciel propre à l'INRIA, et Pipol, un parc hétérogène de 15 machines pour tester la portabilité des logiciels développés à l'INRIA.

Les personnes du service participent aussi au développement de Plateformes Expérimentales (PFE) dans des domaines variés tels que :

- Image et Interaction
- Réseaux de capteurs

- Cluster et Grille
- Transport

2.2 Les réseaux de capteurs

Un réseau de capteurs sans fil est un réseau ad-hoc de nœuds capteurs autonomes répartis dans l'espace pour mesurer des données physiques ou environnementales tels que la température, un champ magnétique, une force, une accélération, ou encore un rythme cardiaque.

Chaque nœud intègre :

- Une unité de captage chargée de capter des grandeurs physiques (chaleur, humidité, vibrations, rayonnement...) et de les transformer en grandeurs numériques
- Une unité de traitement informatique
- Une unité de stockage de données
- Un module de transmission sans fil

Ces nœuds capteurs sont ainsi de véritables systèmes embarqués.

Selon un magazine du MIT, le réseau de capteurs sans fil est l'une des dix nouvelles technologies qui bouleverseront le monde et notre manière de vivre et de travailler. Il répond à l'émergence de ces dernières décennies, de l'offre et d'un besoin accru d'observation et de contrôler des phénomènes physiques et biologiques dans différents domaines :

- Industriels, techniques et scientifique (monitoring de la température, de la pression, de l'hygrométrie, de la luminosité...)
- Ecologie et environnement (surveillance des UV, de la radioactivité...)
- Santé (suivi des malades, veille éco-épidémiologique et épidémiologique)
- Sécurité
- Transports (automatisations diverses, prévention des accidents...)
- L'automatisation des bâtiments domotique

2.3 Le projet ESPAD

2.3.1 Présentation et objectifs du projet

Le projet ESPAD vise des personnes souhaitant pratiquer leur sport en toute sécurité et/ou qui recherchent une meilleure compréhension de leur performance. Il s'intègre au marché des équipements sportifs portatifs et aspire au développement d'équipements innovants qui

seront mis en œuvre dans 3 disciplines sportives : l'aviron, les sports mécaniques et les sports d'hiver.

Le résultat attendu est un socle technologique et 3 prototypes opérationnels.

Le système ESPAD repose sur un réseau de capteurs hétérogènes embarqués, communiquant sans fil et facile à déployer. Il s'appuie aussi sur des outils logiciels appropriés pour la configuration et l'exploitation du système afin de répondre aux attentes « loisir, bien-être, santé, sécurité et performance » des nouvelles pratiques sportives dans l'aviron, les sports mécaniques et les sports d'hiver. ESPAD autorisera les traitements en temps réel (traitement de données simple) ou en post-traitement (traitement complexe), selon les cas d'usage. Le système sera non intrusif et ne dérangera pas les sportifs lors de leur entraînement.

2.3.2 Choix scientifiques et techniques

2.3.2.1 Plateforme utilisée

Les nœuds prototypes utilisés pour constituer le réseau de capteurs sont basés sur des WSN430 développés par l'INRIA. Selon les fonctionnalités visées (nœud Bluetooth, nœud capteur jauge de contrainte...), la carte fille adéquate sera rajoutée.

La connectique sur la WSN430 ne permet la connexion que d'une seule carte fille. Celle-ci peut intégrer des capteurs tels que :

- Capteurs de force
- Cadencemètres
- Jauge de contraintes
- Cardio-fréquencemètres
- EMG
- GPS
- Accéléromètres
- Capteur de température / thermo couple
- Cellule photosensible
- Autres

2.3.2.2 Principe de fonctionnement

Le système ESPAD est constitué d'un nœud contrôleur qui configure et met en place le réseau de nœuds capteurs. Il gère également les transmissions de ces derniers pendant la phase de recueil des données.

Les nœuds capteurs acquièrent les données.

Pour augmenter le rendement dans l'utilisation de la bande passante, le nœud contrôleur organise les nœuds capteurs en groupes échantillonnant à la même fréquence. Il récupère les données échantillonnées en interrogeant un groupe de nœuds capteurs à la fois. Les

nœuds capteurs du groupe interrogé envoient leurs données chacun à leur tour, suivant un ordre défini lors de la mise en place du réseau.

Le GPS, qui fait partie des nœuds capteurs, permet de dater précisément les données.

Les nœuds capteurs enregistrent aussi localement les données qu'ils échantillonnent. Le stockage se fait sur la mémoire flash de la carte du nœud capteur dont la taille est de 1Mo.

L'interface de configuration du système est faite grâce à un smartphone du commerce. A ce jour, les smartphones offrant le plus de liberté au niveau du développement applicatif sont les smartphones Google, sous Android.

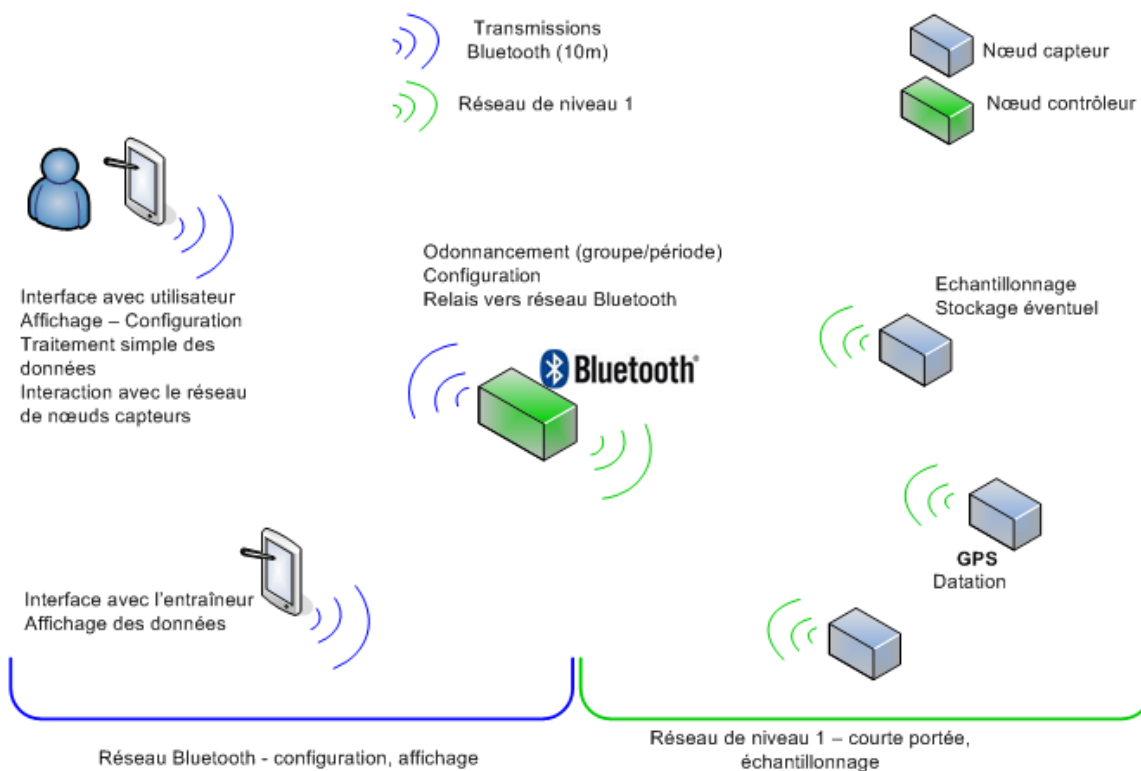


Illustration 2 : Architecture du réseau de capteurs

2.3.2.3 Infrastructure logicielle

La partie logicielle comporte plusieurs briques :

- Le protocole de découverte et sa couche MAC, permettant le déploiement du réseau de capteurs, embarqué sur les nœuds
- Le protocole d'ordonnancement des transmissions et la couche MAC associée, permettant de recueillir les données de tous les capteurs sur un nœud puits (nœud

contrôleur) en réduisant le risque de collisions des paquets envoyés, embarqué sur les nœuds

- La couche applicative embarquée sur les nœuds, qui gère la configuration de l'application et qui contient son profil (types de capteurs à détecter, fréquence d'échantillonnage, stockage local ou remontée de toutes les données vers le nœud puits...). Elle est également en charge de la gestion du stockage local s'il y a lieu et fait l'interface entre l'affichage déporté sur le smartphone Android/PC (communication par Bluetooth) et la configuration du protocole de découverte sur le nœud contrôleur
- La partie visualisation sur le smartphone Android ou le PC portable. Cette partie inclut l'interface de configuration avec le protocole de découverte ainsi que l'affichage en temps réel de certaines données. L'affichage en différé des données traitées sera implémenté uniquement sur PC
- La partie traitement/mise en forme des données, qui peuvent être prise en charge par les nœuds capteurs dans le cadre de traitements simples ou qui seront à la charge du smartphone ou PC
- La fusion de données s'effectue sur PC avec un logiciel complet

2.3.3 Le cas d'utilisation de l'aviron

L'une des applications du projet ESPAD est l'intégration de nœuds capteurs dans la pratique de l'aviron.

Ainsi ils permettent, d'une part d'évaluer la performance sportive en temps réel ou en différé d'un rameur, et/ou d'autre part, de lui offrir un cadre de pratique sportive sécurisé. Ces boîtiers de petite taille ne dérangeront pas les sportifs lors de leur entraînement.

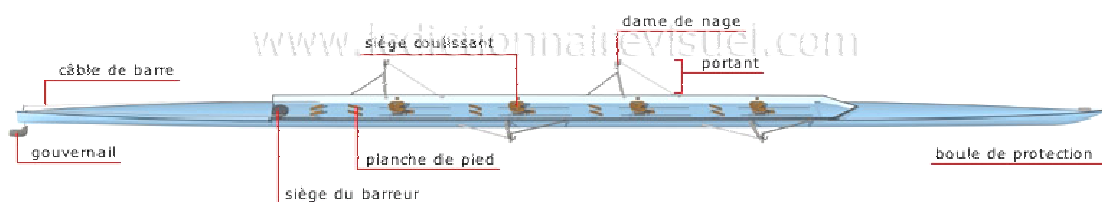


Illustration 3 : Parties d'un aviron

2.3.3.1 Scénario

Avant d'aller sur l'eau, le rameur équipe son bateau avec l'aide de son entraîneur. Ils installent les capteurs de force et d'amplitude au niveau des dames de nage, ainsi que sur la barre de pieds. Chaque boîtier est alors mis en route et le réseau de capteurs s'organise automatiquement. Le nœud contrôleur détecte les nœuds capteurs mis en place ainsi que le smartphone avec l'application ESPAD qui sert d'interface homme machine. Il lui fournit la liste des capteurs détectés avec leur niveau de batterie par Bluetooth.

L'utilisateur peut, s'il le souhaite, activer l'ensemble ou une partie des capteurs. L'utilisateur sélectionne alors les capteurs qu'il souhaite dans la liste, affichée dans le menu de configuration du réseau sur l'écran du smartphone. Ceci lui permet de se focaliser sur un aspect particulier du geste technique en ne sélectionnant que les informations intéressantes données par certains capteurs. La liste des capteurs sélectionnés est alors envoyée par Bluetooth au nœud contrôleur pour vérification.

Si le nombre de capteurs détectés ou choisis est insuffisant (il manque un ou plusieurs capteurs obligatoires), le nœud contrôleur (protocole de découverte) remonte une erreur au smartphone qui l'affiche à l'utilisateur, lui enjoignant de vérifier l'état de ses nœuds capteurs (batterie) ou lui proposant de sélectionner les capteurs obligatoires manquants.

Une fois le choix des capteurs et leur configuration effectués, les capteurs sont étalonnés (mise à zéro des capteurs de forces ou déplacement). Cet étalonnage peut être réalisé de manière indépendante pour chaque capteur.

Lors de la séance, le barreur et l'entraîneur, qui suit à proximité en canot à moteur, sont en mesure de lire l'ensemble des données à tout moment. En effet, le nœud Bluetooth transmet également les données affichées sur le smartphone au terminal de l'entraîneur.

Le nœud contrôleur transmet les données à afficher par Bluetooth au smartphone à une fréquence inférieure (1Hz) à celle utilisée pour l'échantillonnage. L'affichage est alors mis à jour une fois par seconde. Toutes les données échantillonnées sont référencées dans le temps et enregistrées localement sur le nœud capteur. Elles sont récupérables à la fin de la séance à l'aide d'un PC.

A la fin de la séance, l'entraîneur qui possède un pc portable peut facilement extraire les données des différents nœuds capteurs. L'entraîneur lance le logiciel d'exploitation qui détecte tout de suite le matériel branché sur le PC en USB ou par liaison série, et établit la liaison.

L'entraîneur ou le rameur peut alors exploiter les données récupérées sous format CSV sous forme graphique grâce à un logiciel d'exploitation et analyser l'entraînement grâce aux éléments précis reçus.

2.3.3.2 Affichage

L'affichage sur le smartphone comprend la partie configuration du réseau ainsi que l'affichage des données recueillies par le réseau.

La partie configuration comprend :

- Détection du nœud Bluetooth et établissement de la communication,

- Détection de toutes les unités et affichage de leur niveau de batterie,
- Gestion des nœuds capteurs inclus dans le réseau et activés,
- Gestion des erreurs renvoyées par le protocole de découverte demandant une intervention de l'utilisateur,

Les informations disponibles en direct sur l'écran du rameur, de l'entraîneur ou du barreur sont les suivantes :

- La fréquence cardiaque,
- La distance parcourue,
- La vitesse instantanée,
- La vitesse minimale, moyenne, maximale
- La cadence courante, minimale, maximale
- Le nombre de coups effectués
- L'amplitude visuelle et l'angulation,
- L'amplitude et l'angulation efficaces,
- La force maximale développée par coup et l'angle d'application de cette force
- Le temps propulsif.

Les informations sont rafraîchies et affichées périodiquement (1Hz).

Les autres informations (complètes) sont disponibles après téléchargement des informations à l'issue de la séance.

L'affichage de ces informations demande un minimum de traitement, fait d'opérations simples. Cette partie traitement des données brutes se fera principalement sur le nœud capteur.

Plusieurs scénarios d'affichage sont définis, chacun d'entre eux permettant de se focaliser sur un type de données en particulier. Ceci évite de surcharger l'affichage et améliore la lisibilité des informations sur l'écran du smartphone. Dans la mesure où celui-ci reçoit tous les types de données, un système d'onglets sera mis en place, permettant de jongler entre les données à afficher.

Les scénarios d'affichage sont les suivants :

- Vitesse : instantanée, minimale, moyenne, maximale.
- Cadence: courante, minimale, maximale.
- Global : fréquence cardiaque, distance parcourue, vitesse instantanée.
- Amplitude, angulation : Amplitude visuelle, angulation visuelle, amplitude et angulation efficaces.
- Nombre de coups : nombre de coups effectués, force maximale développée par coup et angle d'application de cette force, temps propulsif.
- A définir par l'utilisateur, maximum 4 ou 5 éléments.

2.3.3.3 Equipement

Dans le cadre de ce scénario, le système se compose de :

- Un nœud capteur – fréquence cardiaque, un par rameur
- Un nœud GPS + accéléromètre.

- Nœuds capteurs analogiques :
 - Barre de pieds : force uniquement (1 ou 2 voies, moyenne ou gauche/droite). Une par rameur. L'évaluation de la force appliquée sur la barre de pieds est effectuée à l'aide d'un seul nœud capteur possédant une jauge de contrainte.
 - Dame de nage : force et angle. Ces fonctionnalités peuvent être remplies par un seul nœud capteur, si le nombre d'entrées qu'il possède est suffisant

- Un nœud contrôleur avec une carte Bluetooth.
- Un smartphone permettant de faire l'interface entre l'utilisateur et le réseau de capteurs (affichage).
- Un smartphone pour l'entraîneur.

2.4 La plateforme Android

Initialement le développement de l'application de visualisation des données devait se faire sur iPhone, mais les possibilités de la plateforme iOS sont trop restreintes et contraignantes. Il a été fait alors le choix de passer sur la plateforme Android pour son ouverture et sa communauté très dynamique.

Android est un OS pour téléphone mobile et tablette tactile, promu par Google et l'Open Handset Alliance qui comprend plus de 35 constructeurs, fournisseurs de logiciel, et opérateurs. Il concurrence des plateformes telles que l'iOS d'Apple, Windows Mobile de Microsoft, RIM OS intégré dans les BlackBerry de Research In Motion, WebOS d'HP Bada de Samsung, ou encore Symbian et MeeGo de Nokia.

La plateforme Android est un OS basé sur GNU/Linux entièrement gratuit, sous licence open source Apache 2. Le kit de développement (SDK) et le code source d'Android sont disponibles depuis novembre 2007 en version 1.0. La version actuelle des sources est la 2.2 (Froyo), et la version 3.0 (Gingerbread) devrait sortir normalement en novembre 2010.

Android a une très grande communauté de développeurs qui produisent des applications diverses et variées pour étendre les fonctionnalités du système d'exploitation. Il y a actuellement plus de 100 000 applications sur l'Android Market, ce qui en fait le deuxième environnement de développement le plus populaire, derrière iOS. Les applications sont

écrites pour la plupart en Java, et peuvent utiliser le hardware et les fonctionnalités du système via des bibliothèques Java développées par Google.

Le fait que le code soit complètement ouvert, permet à de nombreux développeurs de le modifier pour y ajouter des fonctionnalités ou corriger des bugs, et ainsi de proposer à la communauté des builds personnalisés, souvent plus avancés que les versions officielles proposées par les constructeurs.

3 Travail réalisé

3.1 Description du sujet

Dans le cadre de mon stage, il m'a été demandé de réaliser l'application Android chargée de la transmission au nœud contrôleur les préférences de l'utilisateur pour la configuration initiale du réseau de capteurs, et de la visualisation en temps réel des informations remontées par celui-ci.

Ces informations comprennent :

- Les informations de configuration du réseau et de ces capteurs, nécessitant ou non une intervention de l'utilisateur,
- Les erreurs éventuelles lors de la configuration,
- L'affichage des performances du (des) sportif(s) lors de la séance.

L'application communique avec le nœud contrôleur du réseau de capteurs grâce au protocole Bluetooth selon le profil SPP (*Serial Port Profile*), et au niveau de la couche applicative grâce à un protocole spécialement rédigé pour le projet. Ce protocole permet de faire de la validation des paquets transmis et spécifie des codes opérations qui permettent un traitement approprié du message une fois celui-ci reçu. Ce protocole définit aussi sans ambiguïté les formats des structures de données encapsulées dans les messages pour les lire et les traiter sans erreur.

3.2 Gestion de projet

3.2.1 Outils utilisés

La gestion de projet a une part très importante dans le projet ESPAD. En effet, pour chaque cas d'utilisation (aviron, sports d'hivers, sports mécaniques) Sandrine, la chef de projet, produit un document qui est ensuite approuvé par les partenaires concernés. Cela permet de définir sans ambiguïtés le contexte et les fonctionnalités qui seront développées. Un

document a aussi été produit pour le protocole de découverte du réseau de capteurs, et un autre pour le protocole de communication entre le nœud contrôleur et le téléphone Android.

Nous faisons régulièrement des réunions avec les membres de l'équipe pour faire le point sur l'avancement du projet et discuter des spécifications.

Pour le logiciel de visualisation de données, nous utilisons le service Gforge fourni par l'INRIA ce qui nous permet d'avoir dans une interface web un dépôt SVN, un tracker de bugs, des forums, un dépôt des documents, des listes de diffusion, et un dépôt de fichier pour télécharger les dernières versions de l'application.

Les applications Android étant codées en Java, pour améliorer la qualité de mon code j'utilise l'outil open source Sonar. En effet, Sonar analyse en profondeur le code source et vérifie le respect de règles selon 7 grands axes : l'architecture et le design, la duplication de code, les tests unitaires, la complexité du code, les bugs potentiels, les règles de programmation et les commentaires, avec 3 outils : Findbugs, Checkstyle et PMD. En tout c'est environ 600 règles qui sont activées pour analyser mon code, et l'objectif est que 100% de ces règles soient respectées.

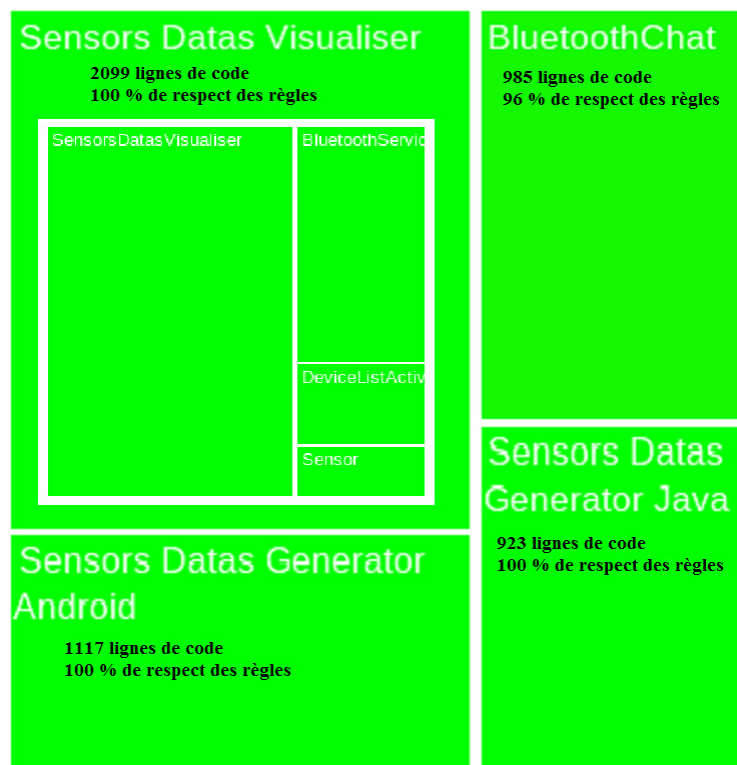
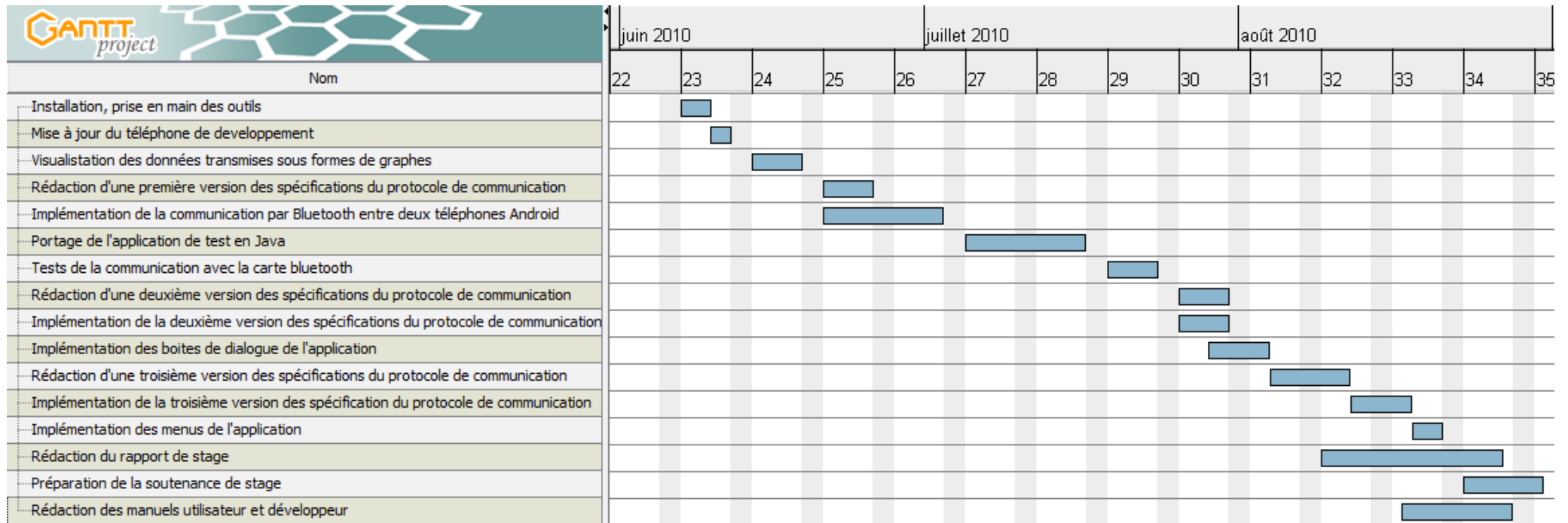


Illustration 4: Rapport Sonar au 26/08/2010

3.2.2 Diagramme de Gantt



3.2.3 Gestion des risques

Risque	Action curative				Action préventive				Probabilité	Impact	Total
	Action	Responsable	Date cible	Date clôture	Action	Responsable	Date cible	Date clôture			
Perte (casse, vol, perte) d'une machine de travail	Remplacer le pc par un de rechange	INRIA	N'importe quand durant le projet	Fin du stage	Faire attention à ses affaires	Florian Fauvarque	Tout au long du stage	Fin du stage	2	4	8
Maladie	Travailler depuis chez moi si possible	Christophe Braillon	N'importe quand durant le projet	Fin du stage	Faire attention à soi	Florian Fauvarque	Tout au long du stage	Fin du stage	1	2	2
Difficulté technologique	Demander de l'aide aux personnes compétentes	Christophe Braillon	N'importe quand durant le projet	Fin du stage	Effectuer différents tutoriaux	Florian Fauvarque	Tout au long du stage	Fin du stage	3	2	6
Panne du serveur hébergeant le SVN	Passer les données par clé USB et les centraliser sur un PC	INRIA	N'importe quand durant le projet	Fin du stage	Faire des sauvegardes régulières sur plusieurs supports	Florian Fauvarque	Tout au long du stage	Fin du stage	1	1	1

3.3 Les différentes tâches

3.3.1 Mise à jour du téléphone

Android est un système d'exploitation régulièrement mis à jour par Google, chaque version apportant de nouvelles fonctionnalités. Ainsi le profil SSP du protocole Bluetooth et la librairie OpenGL ES 1.0 n'ont été ajoutés qu'à partir d'Android 2.0.

Le téléphone de développement qui m'a été fourni est un Google Dev Phone 2, qui correspond à un HTC Magic (aussi appelé Sapphire ou MyTouch 3G aux US) avec les droits administrateurs. Ce téléphone est livré avec une version d'Android 1.6, et la mise à jour vers la dernière version d'Android n'était pas encore disponible sur le site officiel de HTC pendant la durée de mon stage.

Je me suis donc renseigné pour installer une version non officielle d'Android 2.1, et après quelques recherches je me suis tourné vers la ROM CyanogenMod 5. En effet, celle-ci a une grande communauté de développeurs qui la supportent. De plus, la version 5.0.8 pour HTC Magic basée sur Android 2.1 est très stable et apporte les fonctionnalités qu'il me manquait. La version 6.0 de la ROM CyanogenMod, basée sur Android 2.2, est encore en phase de tests et il n'est pas conseillé de l'utiliser en production.

Lorsque la version d'Android 2.2 distribuée Over The Air par HTC sera disponible officiellement pour le Google Dev Phone 2, il conviendra alors peut-être de l'installer pour être sûr que l'OS comporte le moins de bugs possible.

3.3.2 Le protocole de communication

3.3.2.1 Le protocole Bluetooth et le profil SPP

Le protocole Bluetooth est un protocole de communication sans fil pour échanger des données entre périphériques sur une courte distance avec une faible consommation électrique.

Afin de standardiser les communications Bluetooth en fonction de leur utilisation (oreillette sans fil, synchronisation, transfert de fichier, etc...), le Bluetooth SIG (*Special Interest Group*) a défini une vingtaine de profils. Ceux-ci décrivent chacun un ensemble de fonctionnalités et le comportement qu'auront les périphériques afin qu'ils puissent communiquer entre eux.

En particulier, le profil SPP (*Serial Port Profile*) interagit directement avec la couche de transport RFCOMM, et est utilisé pour créer des ports série virtuels sur le périphérique

Bluetooth. Cela permet de transmettre ainsi un flot de données qui seront ensuite lues et traitées par la couche applicative.

Le Service Discovery Protocol (SDP) est un service qui permet aux périphériques Bluetooth de découvrir les services supportés par les autres périphériques à portée et leurs paramètres associés. Chaque service est ainsi identifié par un UUID (*Universal Unique Identifier*). Pour la carte Bluetooth du nœud contrôleur, nous avons utilisé l'UUID standard "00001101-0000-1000-8000-00805F9B34FB".

3.3.2.2 Le protocole Bluetooth dans Android

La plateforme Android inclut le support du protocole Bluetooth et du profil SPP depuis la version 2.0. En effet, grâce à la librairie android.bluetooth les applications peuvent :

- Rechercher des périphériques Bluetooth à proximité
- Interroger l'adaptateur Bluetooth local pour connaître les périphériques appairés
- Etablir un canal RFCOMM
- Se connecter à d'autres périphériques par l'intermédiaire du service de découverte
- Transférer et recevoir des données à d'autres périphériques
- Gérer des connexions multiples

Afin de tester l'établissement de la connexion et le transfert de données entre deux téléphones, j'ai installé sur le Google Dev Phone fourni et sur mon téléphone personnel (un HTC Magic) l'exemple de démonstration BluetoothChat fourni dans le SDK. Cet exemple permet dans une interface assez sommaire de faire un salon de chat, très similaire à une messagerie instantanée de type MSN.

Par la suite, cet exemple a servi de base pour mon application, mais j'ai conservé la version initiale pour envoyer et recevoir des chaînes de caractères au nœud contrôleur afin de tester la connexion.

3.3.2.3 La JSR 82 et son implémentation avec la librairie Bluecove

Afin de ne plus être obligé d'avoir deux téléphones Android pour faire mes tests, j'ai porté mon application de test en java pour pouvoir la lancer à partir d'un simple PC muni d'une puce ou d'un dongle Bluetooth.

La JSR 82 (Java Specification Requests) est une spécification pour les librairies JavaME et J2SE qui veulent implémenter le support du Bluetooth sur les périphériques compatibles. Ainsi, une librairie qui souhaite implémenter la JSR 82 doit inclure les composants suivants :

- Host Controller Interface (HCI) : interface pour accéder aux capacités de la bande de base, et aux paramètres de configuration

- Logical Link Control and Adaptation Protocol (L2CAP) : protocole permettant de multiplexer, segmenter et réassembler les paquets, et transmettre des informations sur la qualité du service
- Service Discovery Protocol (SDP) : service de découverte des périphériques
- RCOMM : protocole émulant un port série RS-232

Les profils suivants sont aussi requis :

- Generic Access Profile : profil le plus commun, utilisé par tous les autres profils pour établir une connexion
- Service Discovery Application Profile : profil qui interagit directement avec le service SDP, afin de trouver des services sur des périphériques Bluetooth à portée
- Serial Port Profile (SPP) : profil utilisé pour créer une liaison série
- Generic Object Exchange Profile : profil générique utilisé par tous les autres profils pour se servir du protocole OBEX

Après quelques recherches, j'ai choisi d'utiliser dans mon application Java la librairie BlueCove. Open Source, celle-ci implémente la JSR-82 et est compatible Windows, Mac et Linux.

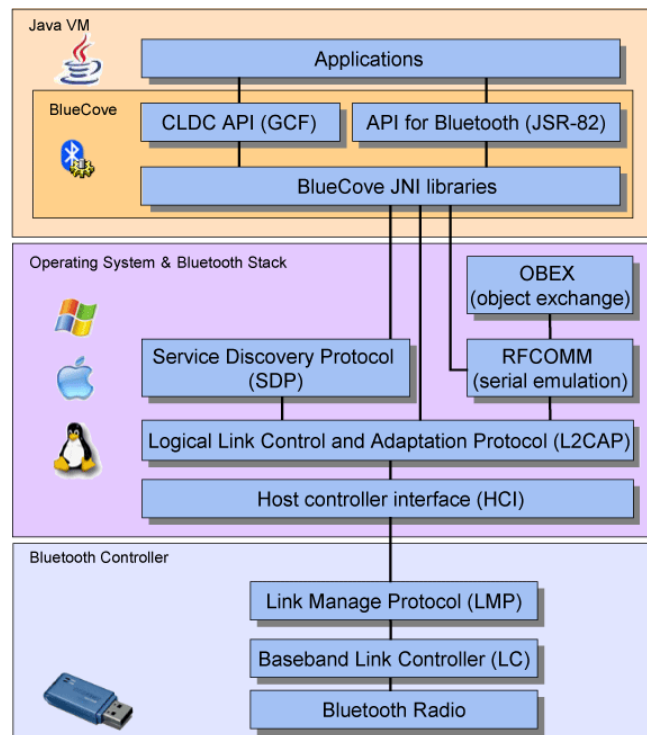


Illustration 5 : Architecture de la stack Bluetooth de la librairie BlueCove

3.3.2.4 La communication entre le téléphone Android et le nœud contrôleur

Afin de constituer un référentiel commun et documenté pour standardiser la forme des paquets échangés entre le téléphone Android et le nœud contrôleur, avec les personnes de mon équipe travaillant sur le projet ESPAD, nous avons rédigé les spécifications du protocole de communication entre le téléphone Android et le nœud contrôleur.

Les données transmises sont ainsi encapsulées avec un entête sous la forme suivante, ce qui permet de valider que le paquet reçu est complet, sinon dans le cas il n'est pas traité :

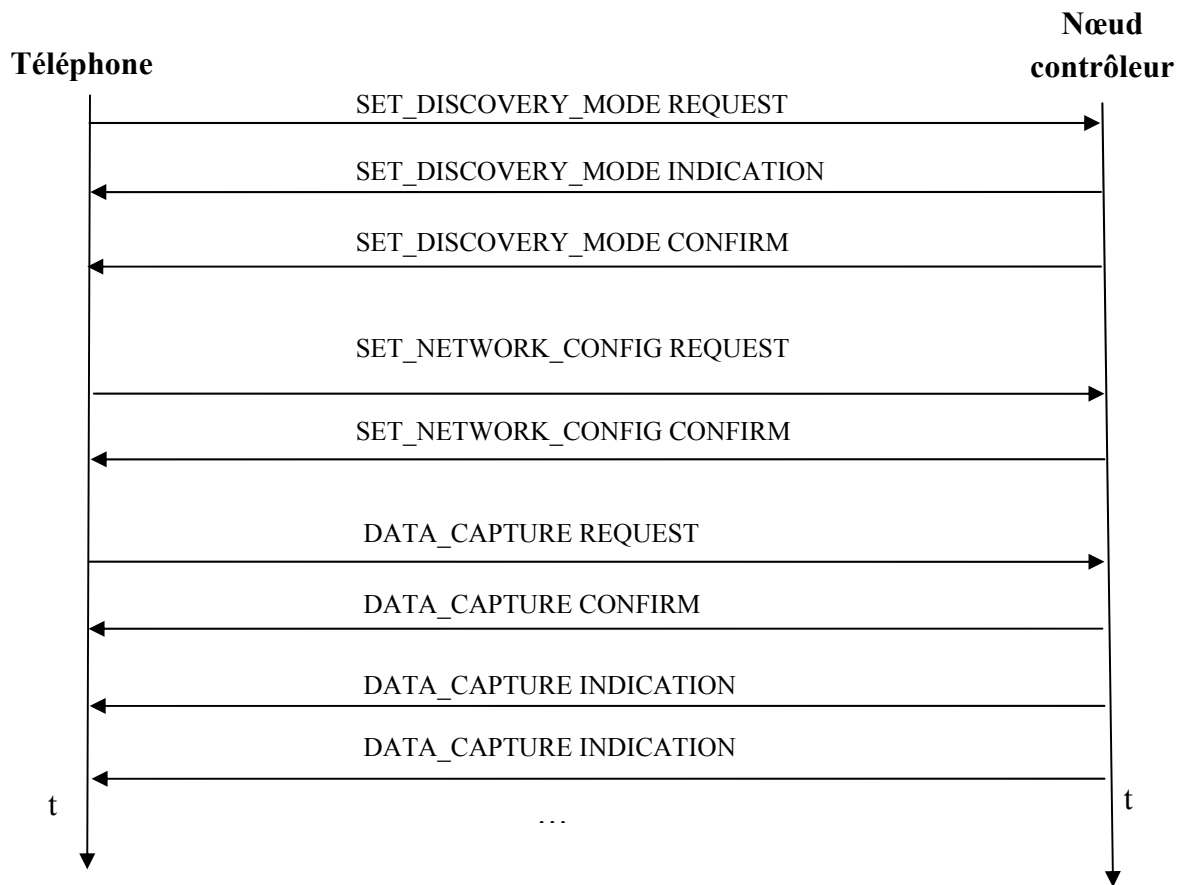
<i>Start</i>	<i>Type</i>	<i>Opcode</i>	<i>Length</i>	<i>Data</i>	<i>End</i>
1 octet	2 bits	6 bits	1 octet	Length octets	1 octet

Intitulé	Valeurs	Description
Start	0x53 ('S')	Indicateur de début de message
Type	01 _b 10 _b 11 _b	Requête : Demande effectuée par le téléphone Confirmation : Réponse à une requête du téléphone. Indication : Indication comportant des données à traiter
Opcode	de 00 0001 _b à 00 1000 _b	Identifiant spécifiant l'opération à effectuer
Length		Taille du champ de données « Data » en octets
Data		Champ de données
End	0x45 ('E')	Indicateur de fin de message

Les Opcodes peuvent être les suivants (la différence entre une opération SET et GET étant que lors du SET on demande au nœud contrôleur d'effectuer une action sur le réseau de capteurs, alors que lors du GET on ne fait que récupérer les informations) :

Intitulé	Valeurs	Description
SET_DISCOVERY_MODE	00 0001 _b	Initialise la découverte des nœuds du réseau et demande au nœud contrôleur de donner les informations concernant les nœuds capteurs
GET_DISCOVERY_MODE	00 0010 _b	Demande la liste des nœuds capteurs disponibles
SET_NETWORK_CONFIG	00 0011 _b	Envoie la nouvelle configuration du réseau
GET_NETWORK_CONFIG	00 0100 _b	Demande l'état de la configuration du réseau
DATA_CAPTURE	00 0101 _b	Recueil des données échantillonnées issues des nœuds capteurs
CRITICAL_ERROR	00 0110 _b	Erreur critique
USER_ERROR	00 0111 _b	Erreur demandant une confirmation de l'utilisateur
WARNING	00 1000 _b	Avertissement

Ainsi un exemple de communication entre le téléphone Android et le nœud capteur pourrait être celui-ci :



A tout moment lors de la configuration, des erreurs peuvent survenir et sont transmises au téléphone sous forme d'indication, qui affiche alors une popup pour signaler à l'utilisateur l'erreur et effectuer l'action appropriée.

3.3.3 Etats de l'application et menus dynamiques

L'application Android de visualisation des données peut être dans les états ci-dessous, qui permettent de déterminer quels sont les menus à afficher pour envoyer des requêtes au nœud contrôleur :

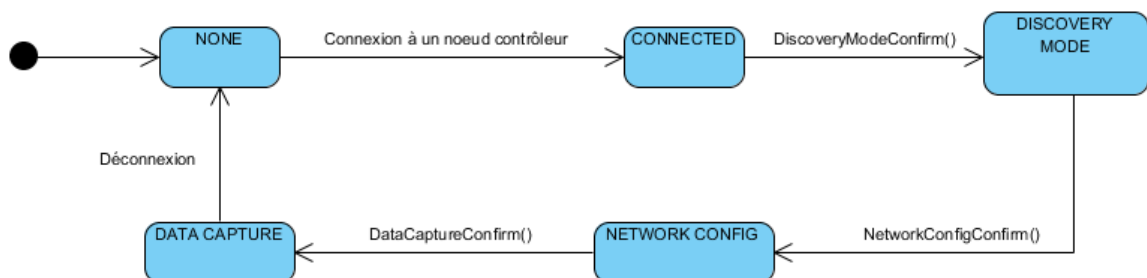


Illustration 6 : Graphe d'état transition de la classe SensorsDataVisualiser.java

Quand l'application vient juste d'être lancée, elle est dans l'état NONE et il est alors possible de lancer la découverte des nœuds contrôleurs à proximité pour s'y appairer puis s'y connecter.

Dans l'état CONNECTED, l'application vient de se connecter à un nœud contrôleur, et il est alors possible de faire un SET_DISCOVERY_MODE pour initialiser la découverte des nœuds ou un GET_DISCOVERY_MODE pour juste demander les informations des nœuds capteurs.

Dans l'état DISCOVERY MODE, l'application affiche une boîte de dialogue pour que l'utilisateur puisse choisir quels sont les nœuds capteurs dont il souhaite récupérer les données, puis une boîte de dialogue pour que l'utilisateur puisse associer à chaque capteur sa position.

Dans l'état NETWORK CONFIG, l'application configure les données locales relatives à la configuration du réseau de capteurs.

Enfin, dans l'état DATA CAPTURE, l'application reçoit en Bluetooth les données des nœuds capteurs, puis les affiche. Il est alors possible de changer par un menu la courbe affichée à l'écran correspondant à un nœud capteur.

3.3.4 Visualisation des données sous forme de graphes

Il n'existe pas de librairie pour tracer des graphes en standard dans l'API du SDK Android développé par Google. Après de nombreuses recherches sur le sujet et quelques essais infructueux, j'ai fini par utiliser la librairie open source AChartEngine. Celle-ci bien que peu esthétique a le mérite de faire ce qu'on lui demande et d'être entièrement gratuite, contrairement à d'autres solutions que j'ai pu trouver.

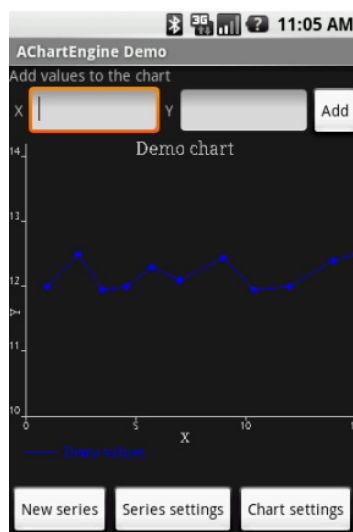


Illustration 7 : Exemple d'application Android utilisant la librairie AChartEngine

Une nouvelle vue est créée pour afficher le graphe, qui est mis à jour en temps réel lorsqu'une donnée d'un nœud capteur est reçue et traitée. Une constante est définie pour choisir le nombre de points à afficher à l'écran, afin de décaler la courbe vers la gauche à chaque nouvelle valeur si le nombre de points maximum est atteint.

3.4 Quelques problèmes rencontrés

Initialement, le sujet de mon stage devait être la visualisation de données 3D issues de cartes de capture de mouvement dans le cadre de la reconstruction de mouvement humains, et j'ai ainsi passé beaucoup de temps à me former sur OpenGL et sur son implémentation dans Android.

Cependant la priorité a été donnée au cas d'utilisation de l'aviron, où les données sont affichées à l'écran sous forme de texte ou de graphe. Ainsi je n'ai pas pu mettre à profit cette formation lors de mon stage, même si c'est certain qu'elle me servira dans ma vie professionnelle plus tard.

Au début les données des capteurs devaient être échantillonnées et transmises à une fréquence de 60Hz, et j'ai eu des problèmes de paquets qui n'étaient pas affichés, car pas lus dans la socket Bluetooth, à cause de la faible taille des paquets reçus à une fréquence trop élevée. Il se trouve que c'est en fait un bug de l'implémentation de la méthode `read(byte[] buffer)` du protocole Bluetooth dans Android, lorsque la taille du buffer de lecture est très supérieure à la taille des paquets dans la socket. Pour le résoudre, j'ai du passer par une méthode détournée en lisant la socket octet par octet, ce qui est moins efficace.

Suite à l'évolution des spécifications du protocole de communication entre la carte Bluetooth et le téléphone Android, les données ne sont maintenant transmises plus qu'à 1Hz, et grâce à l'entête des paquets je connais à tout moment le nombre d'octets à lire. J'ai donc adapté dynamiquement la taille de mon buffer de lecture à la taille réelle des données à lire, ce qui est plus propre au niveau de l'implémentation et plus efficace en termes de performances.

Les spécifications du protocole ont en effet bien évoluées depuis le début du stage et j'ai passé beaucoup de temps à implémenter les modifications. Avec du recul, je me suis rendu compte que si j'avais réfléchi en profondeur aux spécifications du protocole avant même de commencer à l'implémenter, j'y aurais au final gagné en efficacité et j'aurais pris moins de retard sur l'interface utilisateur.

3.5 Ce qu'il reste à faire

Pour le cas d'utilisation de l'aviron, il reste ainsi à améliorer l'IHM de l'application en rajoutant un onglet pour visualiser les performances de chaque rameur de l'aviron, et un onglet pour le bateau.

De plus, selon le type de donnée remontée par le nœud capteur, il est parfois pertinent de l'afficher sous forme de texte, sous forme de demi camembert, sous forme de trajet sur une carte, ou encore sous forme de courbe.

Si dessous un exemple d'interface qu'il aurait été possible d'implémenter pour visualiser les performances de l'équipage de l'aviron :

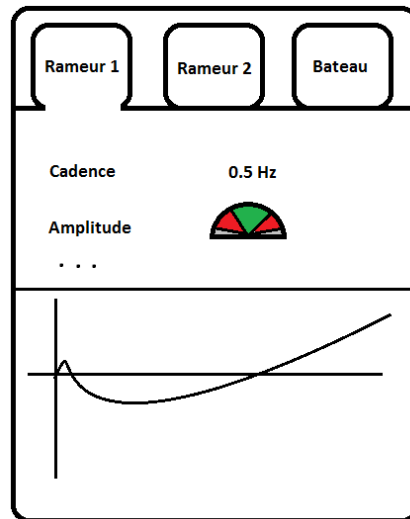


Illustration 8 : Exemple d'interface possible pour le cas d'utilisation de l'aviron

Pour finir, il reste à implémenter les interfaces des cas d'utilisation des sports d'hivers et des sports mécaniques.

4 Bilan et conclusion

Pendant ce stage il m'a été demandé de développer dans le cadre du projet ESPAD l'application Android servant d'interface entre le nœud contrôleur et l'utilisateur.

D'un point de vue professionnel, faire mon stage à l'INRIA fut très instructif et j'y ai beaucoup appris. En effet, ayant déjà développé des applications Android dans le cadre de ma formation à Polytech'Grenoble, j'ai pu me perfectionner dans mes connaissances du SDK d'Android et concevoir une application qui sera véritablement utilisée à l'avenir.

Le protocole Bluetooth étant un élément central de ce stage, je me suis beaucoup formé sur cette technologie, et il est certain que cela me resservira plus tard dans ma vie professionnelle. J'ai aussi appris lors de ce stage à travailler en autonomie, avec rigueur et méthode.

Enfin les difficultés que j'ai rencontrées lors de ce stage m'ont fait réaliser que si j'avais passé plus de temps sur la conception de mon application dès le début, en réfléchissant en amont aux problèmes susceptibles d'arriver, cela m'aurait fait gagner beaucoup de temps au final car je n'aurais pas rencontrés certaines anomalies de mon application.

Sur le plan personnel, mener à bien ce projet fut très gratifiant et m'a permis d'affirmer mon goût pour le développement d'application sur plateforme mobile, et particulièrement pour la plateforme Android.

Pour conclure, on peut vraiment dire que ce stage fut très instructif autant sur un plan professionnel que personnel. Il m'a permis d'avoir un premier aperçu du monde de la recherche et développement, et il m'a donnée envie de poursuivre dans cette voie par la suite.

5 Annexes

5.1 Protocole de communication

Exemples d'OpCodes détaillés du protocole de communication :

<u>SET DISCOVERY MODE REQUEST</u>	
Initialise la découverte du réseau de nœuds capteurs. Envoyé au nœud contrôleur une fois que le téléphone s'est pairé avec le nœud contrôleur.	
Type	Request [01 _b]
Opcode	SET_DISCOVERY_MODE [00 0001 _b]
Length	0 Octet
Data	Pas de donnée

<u>SET DISCOVERY MODE CONFIRM</u>	
Confirmation que le nœud contrôleur a bien envoyé la liste des nœuds capteurs	
Type	Confirm [10 _b]
Opcode	SET_DISCOVERY_MODE [00 0001 _b]
Length	1 Octet
Data	1 Octet BLUETOOTH_ERROR_TYPE (*)

<u>SET DISCOVERY MODE INDICATION</u>		
Envoyée par le nœud contrôleur suite à une requête du téléphone. Donne une liste exhaustive de tous les nœuds capteurs trouvés sur le réseau.		
Type	Indication [11 _b]	
Opcode	SET_DISCOVERY_MODE [00 0001 _b]	
Length	(4 x nombre de nœuds capteur trouvés + 2) Octets	
Data	1 Octet	BLUETOOTH_ERROR_TYPE (*)
	1 Octet	Nombre de nœuds capteur trouvés
	Pour chaque nœud capteur :	
	1 Octet	Id nœud
	1 Octet	Etiquette
	1 Octet	Type de nœud
	1 Octet	Erreur nœud :
	Bit 7	SENSOR_NODE_LVL_BATT_LOW DOUBLE_SENSORS_FOUND NODE_NOT_RESPONDING SENSOR_TYPE_UNKNOWN SENSOR_TYPE_UNUSED
	Bit 6	
	Bit 5	
Bit 4		
Bit 3		
Bit 2		
Bit 1		
Bit 0		

<u>DATA CAPTURE REQUEST</u>			
Demande l'envoi du flux de données provenant du réseau de niveau 1.			
Type	Request [01_b]		
Opcode	DataCapture [00 0101 _b]		
Length	(2 x nombre de nœuds capteur à inclure + 1) Octets		
Data	1 Octet	Nombre de nœuds capteur à inclure	
	Pour chaque nœud capteur :		
	1 Octet	Id nœud	
	1 Octet	Validation de l'inclusion d'un nœud	
		0x00	Non inclut
	0x01	Inclut	

<u>DATA CAPTURE CONFIRM</u>			
Réponse à une requête « DataCapture »			
Type	Confirm [10_b]		
Opcode	DataCapture [00 0101 _b]		
Length	1 Octet		
Data	1 Octet	BLUETOOTH_ERROR_TYPE (*)	

<u>DATA CAPTURE INDICATION</u>			
Flux de données renvoyé par le réseau de niveau 1.			
Chaque trame comprend l'ensemble des paramètres d'un nœud capteur, par exemple un angle, une force ...			
Type	Indication [11_b]		
Opcode	DataCapture [00 0101 _b]		
Length	(Taille des données + 4) Octets		
Data	2 Octet	Numéro de séquence (incrémenté par le nœud contrôleur toute les secondes)	
	1 Octet	Id nœud	
	1 Octet	Taille des données relative au nœud	
	(Taille des données) Octets	Données du nœud	

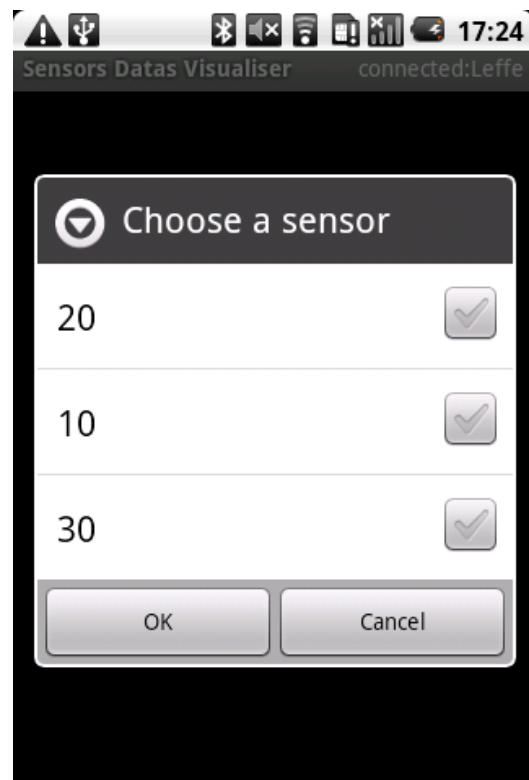
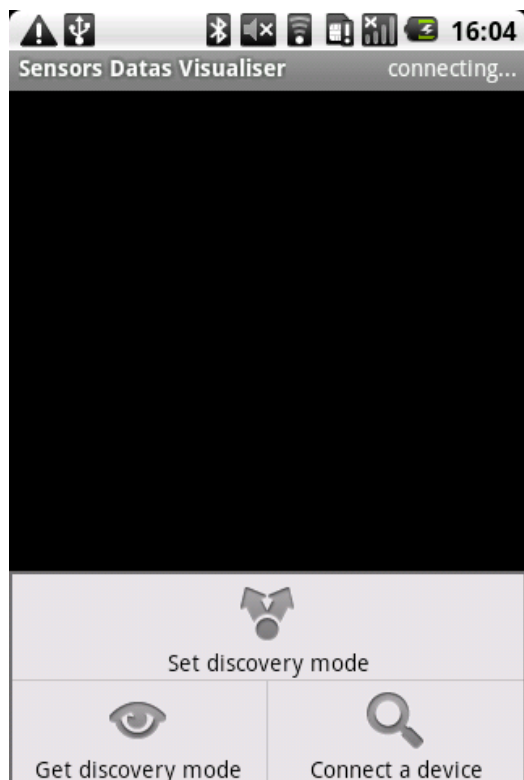
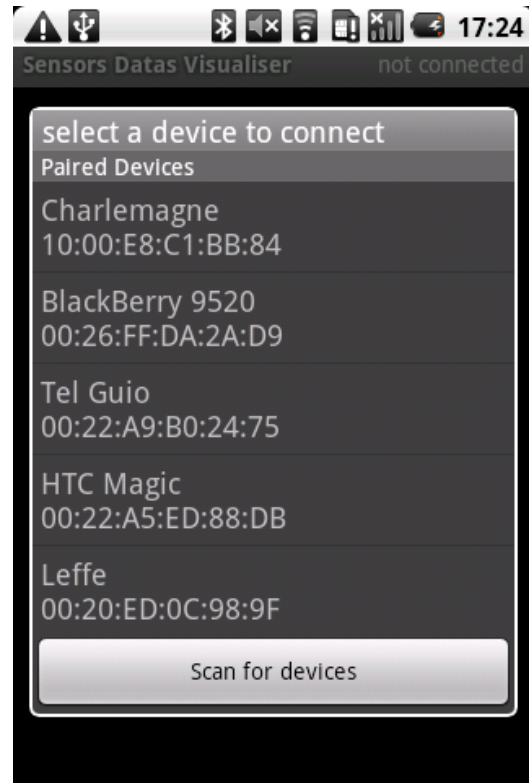
CRITICAL ERROR

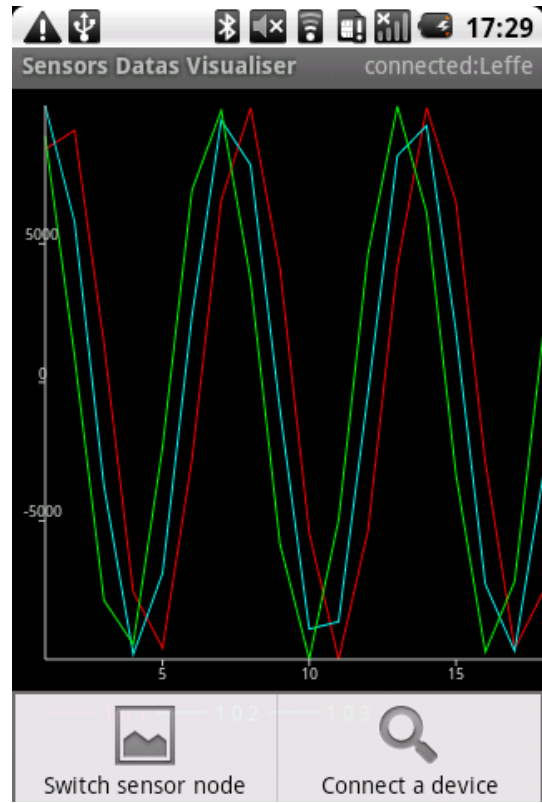
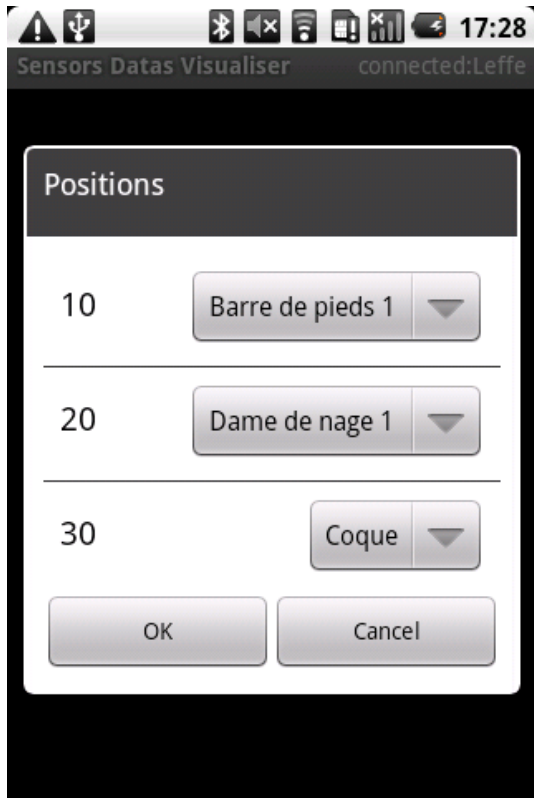
Indication qui peut être renvoyée par le nœud contrôleur une fois que la requête de découverte du réseau a été reçue.

Utilisée pour signaler une ou plusieurs erreurs critique à la fois : dans un seul octet, un bit reflète une erreur particulière.

Type	Indication [11_b]		
Opcode	CRITICAL_ERROR [00 0110 _b]		
Length	1 Octet		
Data	1 Octet	Erreur critique :	
		Bit 7	CTLER_NODE_LVL_BATT_LOW
		Bit 6	
		Bit 5	MISSING_MANDATORY_SENSORS
		Bit 4	
		Bit 3	
		Bit 2	DOUBLE_SENSORS_FOUND
		Bit 1	
		Bit 0	

5.2 Captures d'écran de l'application





6 Bibliographie / Webographie

1. **Guignard Damien, Chable Julien et Robles Emmanuel.** *Programmation Android - De la conception au déploiement avec le SDK Google Android 2.* s.l. : Eyrolles, 2010.
2. **Hopkins Bruce et Ranjith Antony.** *Bluetooth for Java.* s.l. : Apress, 2003.
3. *Android Developers.* [En ligne] <http://developer.android.com/>.
4. *The Official Bluetooth Technology Info Site.* [En ligne] <http://www.bluetooth.com/>.
5. Racine du dépôt SVN. *Gforge.* [En ligne] <https://gforge.inria.fr/scm/viewvc.php/?root=sensvisualiser>.
6. *AChartEngine.* [En ligne] <http://www.achartengine.org/>.
7. *BlueCove JSR-82 project.* [En ligne] <http://bluecove.org/>.

7 Liste des documents rédigés pendant le stage

1. **Avakian Sandrine, Roche Guillaume et Fauvarque Florian.** *Définition du protocole de communication entre le nœud contrôleur et l'application de visualisation des données.* 12/08/2010. Version 3. 15 pages. Français
2. **Fauvarque Florian.** *Documentation développeur.* 26/08/2010. Version 1. 3 pages. Français
3. **Fauvarque Florian.** *Documentation utilisateur.* 26/08/2010. Version 1. 1 page. Français

8 Table des révisions du SVN

Numéro de révision	Auteur	Date	Message
1	fauvarqu	Mon Jul 12 15:54:10 2010 UTC	First commit
2	fauvarqu	Tue Jul 13 13:24:20 2010 UTC	Generation of 3 different types of sensors
3	fauvarqu	Fri Jul 16 14:29:23 2010 UTC	Minor bug corrections
4	fauvarqu	Tue Jul 20 09:40:36 2010 UTC	Adding source code of WSN430 device and minor bug correction
5	fauvarqu	Tue Jul 20 09:44:21 2010 UTC	Tag of version 0.1
6	fauvarqu	Thu Jul 29 07:34:57 2010 UTC	Add of sensors streaming choice dialog
7	fauvarqu	Tue Aug 3 14:33:22 2010 UTC	Add of the sensors position choice dialog
8	fauvarqu	Thu Aug 26 10:14:34 2010 UTC	Add of dynamic menus

9 Résumé

9.1 Résumé en Français

Les progrès conjoints de la microélectronique, de la microtechnologie, des technologies de transmission sans fil et des applications logicielles ont permis de produire à coût raisonnable des micro-capteurs de faible volume, susceptibles de fonctionner en réseaux.

Le projet ESPAD consiste ainsi à expérimenter des réseaux de capteurs sans fil pour les sportifs souhaitant avoir une meilleure compréhension de leurs performances. Il s'intègre au marché des équipements sportifs portatifs et aspire au développement d'équipements innovants qui seront mis en œuvre dans 3 disciplines sportives : l'aviron, les sports mécaniques et les sports d'hiver.

Durant ce stage, il m'a été demandé de développer l'application Android servant d'interface entre le réseau de capteurs et l'utilisateur pour la configuration de celui-ci, et la visualisation en temps réel des informations des capteurs, ainsi que le protocole leur permettant de communiquer.

9.2 Abstract in English

Joint progress in microelectronics, microtechnology, wireless technology and software applications have allowed to produce at reasonable cost small micro-sensors, working in networks.

The project ESPAD consists in experimenting wireless sensor networks to athletes willing a better understanding of their performance. It takes places in portable sports equipment market and aspires to the development of innovative equipment that will be implemented in three sports: rowing, motor sports and winter sports.

During this internship, I developed an Android application to relay information to the user from the controller node, to configure it and display real-time information from sensors. I also developed the protocol allowing them to communicate.