1. **Android**
2. Le choix d’Android

Il existe aujourd’hui sur le marché des systèmes d’exploitation pour Smartphones cinq grands acteurs :

* Nokia avec son système d’exploitation Symbian
* RIM avec BlackBerry OS
* Microsoft avec Windows Phone 7
* Apple avec IOS4
* Google avec Android

Ces dernières années, Android a eu une croissance considérable (+886 % en un an) et cela devrait continuer, les différentes études montrent qu’Android devrait être leader des systèmes d’exploitation mobiles en 2012. Cela nous permet de croire à un avenir pour ce projet.

De plus, Android est basé sur un noyau linux, ainsi, Android est libre et ouvert. Cela permet un développement aisé et peu onéreux.

Enfin, Android utilise le couple de langage XML – Java pour le développement. Cela nous permet de mettre en œuvre des langages utilisés au cours de notre DUT tout en utilisant un nouveau SDK. Nous avons décidé d’utiliser la version 1.6 d’Android (appelée Donut) pour notre développement. Ce choix a été fait en raison du matériel disponible et par la volonté de diffuser notre application à un maximum de personnes possible. En effet, Android 1.6 est aujourd’hui encore très utilisé et les outils de développement associés permettent une conception aisée.

1. L’histoire d’Android

En juin 2005, Google rachète une start-up du nom d’Android, spécialisée dans  le développement d’application mobile. C’est là que commence le développement de leur propre système d’exploitation mobile. Google présente au public ce système pour la première fois en 2006 mais il faudra attendre 2007 pour avoir l’annonce officielle.

C’est seulement fin 2008 que le premier téléphone utilisant le système d’exploitation Android arrive sur le marché américain : c’est le G1 sorti le 22 octobre. Ce téléphone utilise la version 1.1 d’Android. C’est aussi en 2008 que Google présente l’Android Market : un magasin d’application pour téléphone sous Android tel l’AppStore pour l’Iphone (sorti en 2007). Ce magasin permet de télécharger directement de petites applications afin de décupler les possibilités du téléphone en utilisant toutes les ressources des appareils (GPS, accéléromètre etc.…)

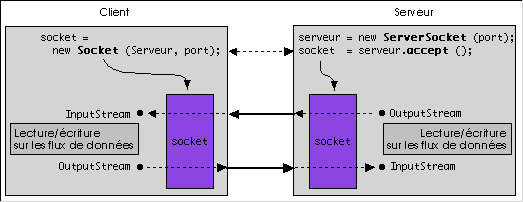
Google continuera à publier assez fréquemment de nouvelles versions d’Android (quatre en deux ans et qui portent toutes le nom d’un dessert et suivent l’alphabet) et les terminaux mobiles se multiplient. En 2009, Google a décidé de distribuer son propre téléphone, le Nexus One afin de stimuler le développement d’Android dans le monde. Il récidivera en décembre 2010 avec le Nexus S puis en 2011 avec le Galaxy Nexus.

Aujourd’hui, Android est en version 2.3 et Google prévoit de sortir la version 3.0 pour le début de l’année 2011.  On compte 160 000 téléphones sous Android activés chaque jour dans le monde et la barre des 100 000 applications a été atteinte dans l’Android Market fin juillet 2010.

1. Le principe de l’application Android

Pour notre projet, nous devons permettre à notre téléphone de communiquer avec le serveur gérant les XBee. Pour ce faire nous avons choisi d’utiliser une connexion par socket. Un socket est une sorte de tube, si celui-ci relie un serveur à un client, cela permettrait de communiquer en mettant au point un protocole assez simple. L’utilisation des sockets étant similaire à celle de flux, nous n’aurons qu’à mettre nos instructions dans ces flux pour que le serveur les reçoive. Nous aurons donc besoin de deux processus : un pour « l’écriture » qui permettra à notre application d’envoyer des informations au serveur et un pour la « lecture » qui lui, tournera en permanence.

Sur l’exemple suivant, on peut voir comment se passe la connexion entre le client et le serveur. Nous créons d’abord notre serveur qui est relié à un port du réseau puis on lui autorise les connexions par socket. Notre application Android crée ensuite un socket relié au serveur. Nous créons ensuite nos flux d’entrée et de sortie afin de pouvoir communiquer avec le serveur. Nous appellerons une méthode « *write(String message)* » qui permettra d’écrire dans le flux de sortie le message à destination du serveur. Pour l’écoute, nous créerons un processus qui lui, écoutera en permanence sur le flux d’entrée tout message en provenance du serveur.



1. Contexte d’utilisation

Comme nous l’avons vu précédemment, notre projet s’inscrit principalement dans la domotique. Notre application sera tout à fait conforme à cet environnement. En effet, l’intérêt de développer notre application sur Smartphone permettra une utilisation complètement nomade : l’utilisateur pourra contrôler sa maison quel que soit l’endroit où il se trouve (que ce soit depuis son canapé ou depuis son lieu de travail). De plus, l’idée d’un appareil mobile tout-en-un se concrétise un peu plus avec cette application.

1. Le SDK Android

Afin de faciliter le développement sous Android, Google met à disposition gratuitement un SDK. Cette boite à outils, basée sur le langage Java propose tout le nécessaire pour développer des applications sous Android :

* Les librairies nécessaires au développement.
* Une documentation complète.
* Des drivers pour les téléphones afin d’installer rapidement les applications et de les tester sans passer par l’Android Market.
* Un système de gestion d’émulateurs de téléphones sous Android.
* Des outils de débogages.



*L’émulateur Android fourni avec la JDK*

Le SDK, bien que basé sur le langage Java comme dit précédemment, utilise le langage XML pour construire les interfaces des applications.

1. L’environnement de devellopement

NetBeans (appelé à l'origine Xelfi), a vu le jour en tant que projet d'étudiant en République Tchèque, en 1996. Le but était d'écrire un EDI Java semblable à Delphi, mais en Java. Une compagnie fut créée autour de ce projet, nommé NetBeans. Il y a eu deux versions commerciales de NetBeans, appelées Developer 2.0 et Developer 2.1. Aux alentours de mai 1999, NetBeans sortit une version bêta de ce qui aurait dû être Developer 3.0. Quelques mois plus tard, en octobre 1999, NetBeans fut racheté par Sun Microsystems. Après quelques temps de développement supplémentaire, Sun sortit l'EDI Forté Fro Java, Edition Communauté - le même EDI qui avait été présenté en bêta comme NetBeans Developer 3.0.

Il y a toujours eu un intérêt pour l'open-source chez NetBeans, c’est pourquoi, en juin 2000, Sun mis l'EDI NetBeans en open-source.

NetBeans présente aujourd’hui de nombreux atouts justifiant son utilisation au sein de ce projet :

* Il est simple d’utilisation.
* Nous sommes familiarisés avec son interface car nous l’avons déjà utilisé.
* La documentation est riche notamment car NetBeans est développé par Sun, et c’est Sun qui a mis au point Java.
* Le produit est de qualité et a fait ses preuves.
* Les mises à jour son fréquentes et le logiciel ne fait que s’améliorer.

1. Présentation du XML

Lors de la conception de notre application Android, nous aurons besoin d’utiliser le langage XML. Il sera utilisé dans le « manifeste » de l’application et dans le « main.xml ».

XML (Extensible Markup Language, langage extensible de balisage) est un langage permettant la définition d’objets. Il a vu le jour pour sa version 1.0 en 1998 et la version 1.1 a été publiée en 2004. Un document XML peut être vu comme un arbre : le fichier contient un ou plusieurs objets principaux qui eux-mêmes contiennent d’autres objets et ainsi de suite. L’intérêt du XML est que l’utilisateur peut définir lui-même tous ces objets. Nous verrons dans notre application comment se servir du XML pour créer nos objets. La syntaxe est très précise, un saut de ligne avant la déclaration du document retournera une erreur lors du passage du code au parseur.

Voici une anecdote concernant le XML : le groupe de travail XML ne s'est jamais rencontré face-à-face, la conception a été réalisée en utilisant courriers électroniques et téléconférences hebdomadaires. Les principales décisions de conception ont été prises en une vingtaine de semaines de travail intense entre juillet et novembre 1996, lorsque le premier travail de spécification XML a été publié. D'autres travaux de conception sont poursuivis jusqu'en 1997, puis le XML 1.0 est devenu une recommandation W3C le 10 février 1998.