

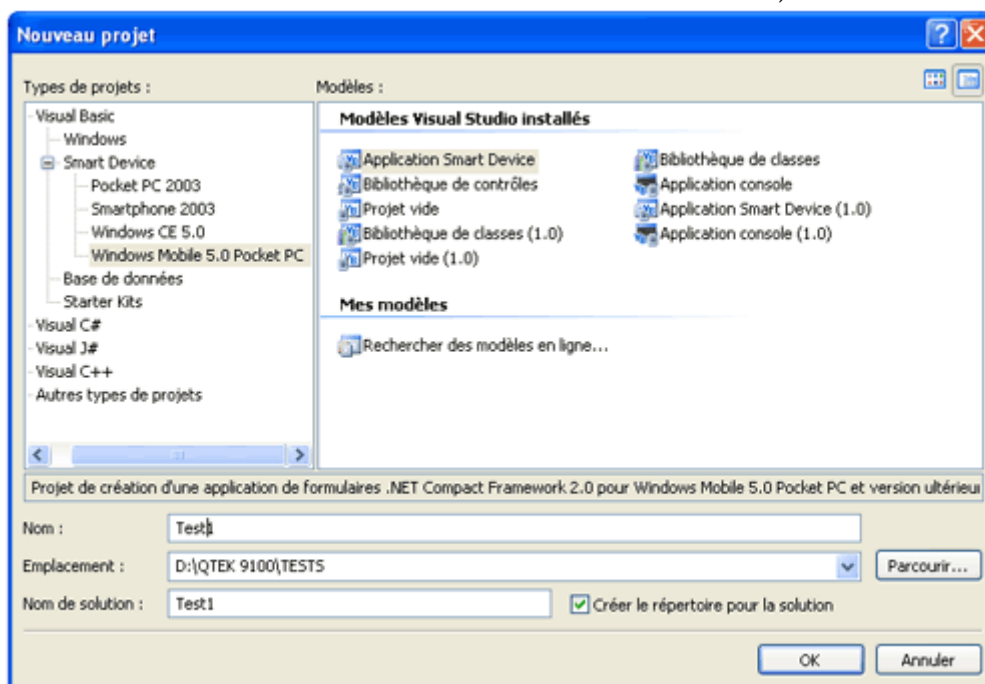
TD n° 2 Compact .Net Framework avancé, par J.-Y Tigli

1 Introduction au SDK Windows Mobile 6.x

Télécharger le Windows Mobile 6.o SDK ou suivant. Ce SDK supporte à la fois un simulateur de réseau cellulaire et de GPS.

- Le simulateur de GPS vous permet de tester vos applications géolocalisées sans avoir d'accès à un dispositif GPS,
- Un nouvel Émulateur de réseau cellulaire vous permet de tester le comportement de votre application dans une variété de scénarios comme des appels téléphoniques entrants, des messages de SMS entrants, des appels téléphoniques perdus et la connectivité de données perdues,
- Ces émulateurs de dispositifs vous permettent de tester vos applications sur une large gamme de dispositifs sans avoir besoin de les acheter.

L'utilisation du SDK se fait au travers Visual Studio avec Menu Fichier > Nouveau Projet :



On choisit un projet "Smart Device" > "Windows Mobile 5.o Pocket PC" > "Application Smart Device", on indique le nom du projet (ici Test1) et le répertoire cible et zou... on retrouve notre environnement standard de développement avec un Form1 relooké :)

Contributeurs : J.-Y. Tigli

Polytech'Nice – Sophia

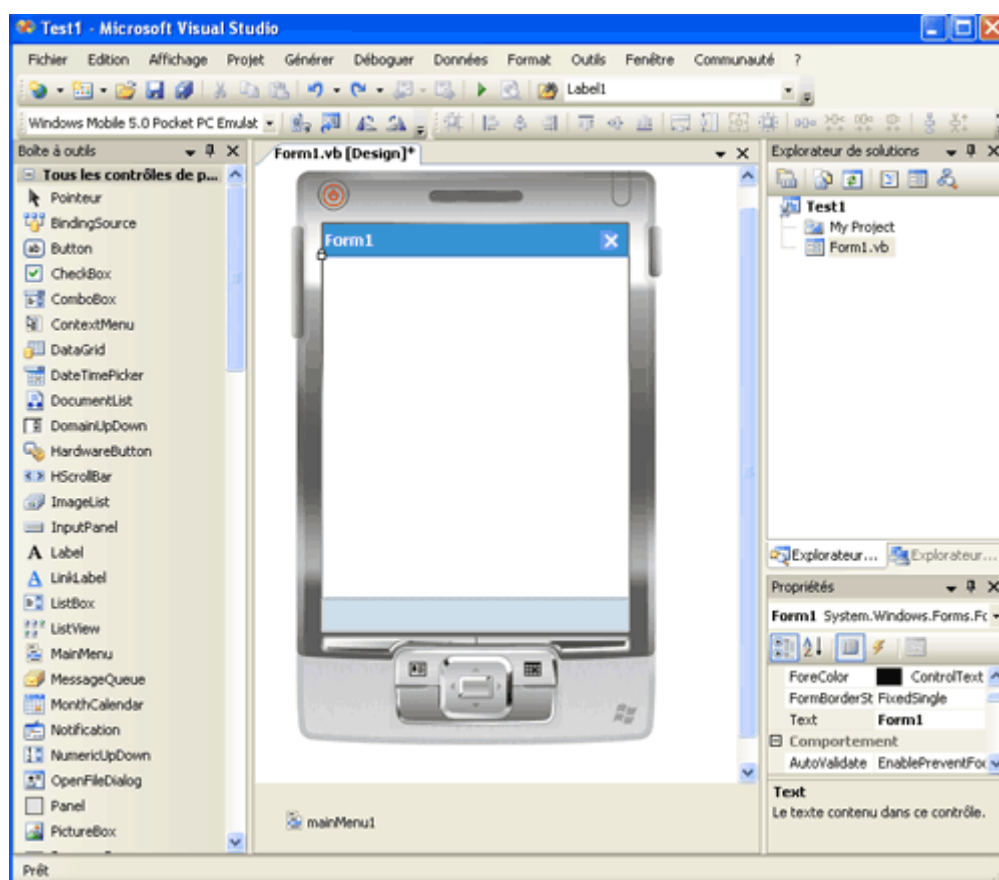
Université de Nice – Sophia Antipolis

930, Route des Colles – B.P. 145 - 06903 Sophia Antipolis Cedex – France

Tél : +33 (0)4 92 96 50 50 – Fax : +33 (0)4 92 96 50 55

<http://www.polytech.unice.fr/>

TD n° 2 Compact .Net Framework avancé, par J.-Y Tigli



2 GPS : Global Positioning System

2.1 Introduction

Le GPS (Global Positioning System) est un système de localisation terrestre. A la fin des années 70, le principal souci du département de la défense des US est de créer un système permettant à tous les éléments de l'armée américaine de se positionner de manière précise et rapide n'importe où à la surface de la terre.

Le système GPS est composé d'une constellation de satellites (24 au total) positionnées de telle manière qu'au moins 3 d'entre eux soient visibles de n'importe quel point du globe. Chaque satellite émet un message codé donnant à ceux qui le reçoivent des informations sur sa position. Le récepteur GPS fait alors de savants calculs pour déterminer, grâce au décalage entre le moment et l'envoi et de la réception du signal, la position en termes de latitude, longitude et altitude (d'autres informations telles que la vitesse sont aussi calculées).

Grâce à un ingénieux système de codage, les données sont très précises pour l'armée américaine, et beaucoup moins précises pour les civiles. Nous devons nous contenter des données civiles, mais généralement ça suffit pour se positionner. La réception des données provenant des satellites et les calculs complexes qui doivent être réalisés pour déterminer la position sont effectués par des appareils spécialisés, de plus en plus bon marché, que sont les GPS.

Contributeurs : J.-Y. Tigli

Polytech'Nice – Sophia

Université de Nice – Sophia Antipolis

930, Route des Colles – B.P. 145 – 06903 Sophia Antipolis Cedex – France

Tél : +33 (0)4 92 96 50 50 – Fax : +33 (0)4 92 96 50 55

<http://www.polytech.unice.fr/>

TD n° 2

Compact .Net Framework avancé,

par J.-Y Tigli

2.2 Le protocole NMEA

Le GPS est un périphérique qui dialogue avec le reste du monde via une liaison série en utilisant le protocole NMEA (National Marine & Electronics Association). Ce protocole a été tout d'abord utilisé dans la marine et il est devenu au fil du temps la référence dans ce domaine. Le protocole NMEA est une suite de trames ASCII séparées par les caractères CRLF (comme dans un fichier texte). Chaque trame étant elle-même composée de champs séparés par des virgules. A la fin d'une trame, une somme de contrôle permet de vérifier que la trame reçue est valide et qu'elle peut être prise en compte.

Voici un exemple de trame NMEA :

```
$GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68
```

Cette trame a pour identifiant \$GPRMC, ce qui permet de savoir quelles sont les informations que l'on va trouver ensuite et dans quel ordre. Cette trame est donc composée de 12 champs (en comptant \$GPRMC) et sa somme de contrôle est la valeur hexadécimale 68. La somme de contrôle est séparée du reste de la trame par l'étoile. Elle est calculée en effectuant le "ou" exclusif (xor) entre tous les caractères ASCII de la trame jusqu'à l'étoile sans prendre le \$ de début.

Il existe plusieurs type de trames (\$GPGGA, \$GPRMC, \$GPVTG, etc..) car chacune existe pour un type de données bien précis. La trame \$GPVTG donne des informations sur le cap et la vitesse, alors que \$GPGGA sur les satellites disponibles et la précision des mesures.

2.3 La trame \$GPRMC

Le but et tout de même de connaître notre position. Pour cela il existe la trame \$GPRMC qui donne la latitude, la longitude et la vitesse. Pour notre exemple ce sera parfait.

La trame \$GPRMC est composée des informations suivantes :

- 00 \$GPRMC
- 01 Heure du fix
- 02 Alerte (A=OK ; V=WARNIG)
- 03 Latitude au format ddmm.ss
- 04 Sens de la latitude (N=Nord=Positif, S=Sud=Négatif)
- 05 Longitude au format dddmm.ss
- 06 Sens de la longitude (E=Est=Positif, W=Ouest=Négatif)
- 07 Vitesse au sol en Knots (noeuds)
- 08 Cap vrai
- 09 Date du fix
- 10 Déclinaison magnétique
- 11 Sens de la déclinaison magnétique

Vous noterez que les champs 3 et 4 nous donnent la latitude, les champs 5 et 6 la longitude et le champ 7 la vitesse. Le champ 2 est là pour nous indiquer si les données fournies sont fiables ou non. Le GPS délivre à intervalle régulier (généralement chaque seconde) une série trames dont la trame \$GPRMC, et ce même s'il n'est pas encore

Contributeurs : J.-Y. Tigli

Polytech'Nice – Sophia

Université de Nice – Sophia Antipolis

930, Route des Colles – B.P. 145 – 06903 Sophia Antipolis Cedex – France

Tél : +33 (0)4 92 96 50 50 – Fax : +33 (0)4 92 96 50 55

<http://www.polytech.unice.fr/>

TD n° 2 **Compact .Net Framework avancé,** par J.-Y Tigli

dans la possibilité de donner des informations fiables (pas assez de satellites fixés par exemple). Cet indicateur permet de savoir à coup sûr si les données peuvent-être utilisées ou non.

2.4 Communication avec un GPS

La communication s'effectue via une liaison série réelle ou virtuelle. Le GPS envoie périodiquement des informations sous la forme de trames \$GPRMC.

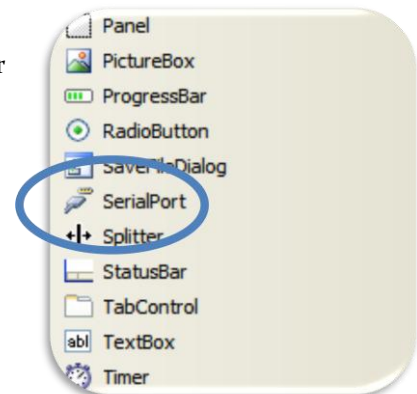
Le récepteur GPS utilise une liaison série pour communiquer.

Il faut donc utiliser les fonctions de gestion des ports série pour communiquer avec un récepteur GPS.

Le Compact Framework, depuis la version 2, propose **le contrôle SerialPort**

Espace de noms : System.IO.Ports

Contrôle : SerialPort



Vous pouvez aussi utiliser les classes suivantes du SDK :

Gps : The interface to the managed GPS API. use this class to open, close, and query the device state, and to query the position data from your GPS hardware.

GpsPosition : Contains the GPS position data received from the GPS hardware.

GpsDevice : Contains the GPS device state data received from the GPS hardware.

Utils : Native memory allocation utilities.

3 Autres éléments du TD :

3.1 Gestion de la mise en veille

Lors d'une communication série, la mise en veille peut se déclencher à tout moment.

Pour assurer une communication de qualité, il faut refuser la mise en veille automatique en appelant à intervalle régulier l'API **SystemIdleTimerReset**

Déclaration C# :

```
[DllImport("coredll.dll")]
static extern void SystemIdleTimerReset();
```

3.2 Sérialisation XML

La sérialisation XML permet de sauvegarder et de restaurer un objet en écrivant les données qui le composent sous la forme d'un flux XML.

Existe dans le Framework .NET depuis le début.

Contributeurs : J.-Y. Tigli

Polytech'Nice – Sophia

Université de Nice – Sophia Antipolis

930, Route des Colles – B.P. 145 – 06903 Sophia Antipolis Cedex – France

Tél : +33 (0)4 92 96 50 50 – Fax : +33 (0)4 92 96 50 55

<http://www.polytech.unice.fr/>

TD n° 2

Compact .Net Framework avancé,

par J.-Y Tigli

Existe dans le Compact Framework .NET que depuis la version 2.

Espace de noms: `System.Xml.Serialization`

Dans votre application, les informations de géo localisation de chaque enregistrement seront stockées dans un fichier XML créé par la sérialisation d'un objet.

4 Sujets de TD avancés :

4.1 Traceur

A partir des informations précédentes, développez une application qui trace et enregistre vos déplacements.

4.2 Création d'un client Web service sous Windows Mobile

A partir de sites comme Xmethods (<http://www.xmethods.net/>) et Salcentral (<http://www.salcentral.com>), choisissez un service et développez un client sur votre smartphone. Vous pouvez par exemple envoyer un SMS avec [SMS Services](#).

4.3 Perspectives

Les plus avancés pourront combiner ces deux développements en invoquant des web services paramétrés par la position GPS, comme ceux de <http://www.cdyne.com/products/demographics.aspx>.

5 Liens utiles

Liens utiles sur le GPS et le NMEA :

<http://www.codeppc.com/dotnetcf2/gps/index.htm>

<http://www.gpspassion.com>

<http://ditwww.epfl.ch/SIC/SA/publications/FI98/fi-5-98/5-98-page1.html>

Lien utile sur la sérialisation :

<http://www.codeppc.com/dotnetcf2/serialization/index.htm>

Contributeurs : J.-Y. Tigli

Polytech'Nice – Sophia

Université de Nice – Sophia Antipolis

930, Route des Colles – B.P. 145 - 06903 Sophia Antipolis Cedex – France

Tél : +33 (0)4 92 96 50 50 – Fax : +33 (0)4 92 96 50 55

<http://www.polytech.unice.fr/>