

# Voiture éco-shell : Récupération de données

**RAPPORT DE PROJET TUTORÉ**

***VOUMARD Benoit***  
***RONDOT Maxime***  
***MOTTE Jean-Baptiste***

## **SOMMAIRE :**

### **Introduction**

#### **I. L'envoi des SMS**

- 1) Le module SIM800L
- 2) Les commandes AT

#### **II. Le travail réalisé**

- 1) Le programme
  - C
  - C#
- 2) La carte électronique

#### **III. Les perspectives**

## **Introduction :**

L'éco-marathon est une compétition ouverte aux étudiants sur le thème de la performance énergétique. Le but est réaliser un véhicule permettant de faire un maximum de distance à l'aide de l'apport énergétique d'un litre d'essence (ou son équivalent). Plusieurs type de propulsions sont acceptées au sein de la compétition : le moteur thermique, le moteur électrique, la pile à combustible et le solaire.

Dans notre situation, le véhicule sera propulsé grâce à une pile à combustible. Des mesures de tension, courant et puissance doivent être réalisées et envoyées au stand pour permettre une surveillance en temps réel. Notre projet était donc d'assurer l'envoi et la réception des données via SMS. Pour cela nous avons utilisé un PIC18f4580 et un module SIM800L. Deux cartes électroniques seront produites à la fin de ce projet. L'une sera installé sur le véhicule pour envoyer les données et l'autre sera au stand pour recevoir.

Dans un premier temps, nous expliquerons comment fonctionne le module SIM800L et les commandes AT. Ensuite nous parlerons de la structure de la carte électronique.

## I . L'envoi des SMS

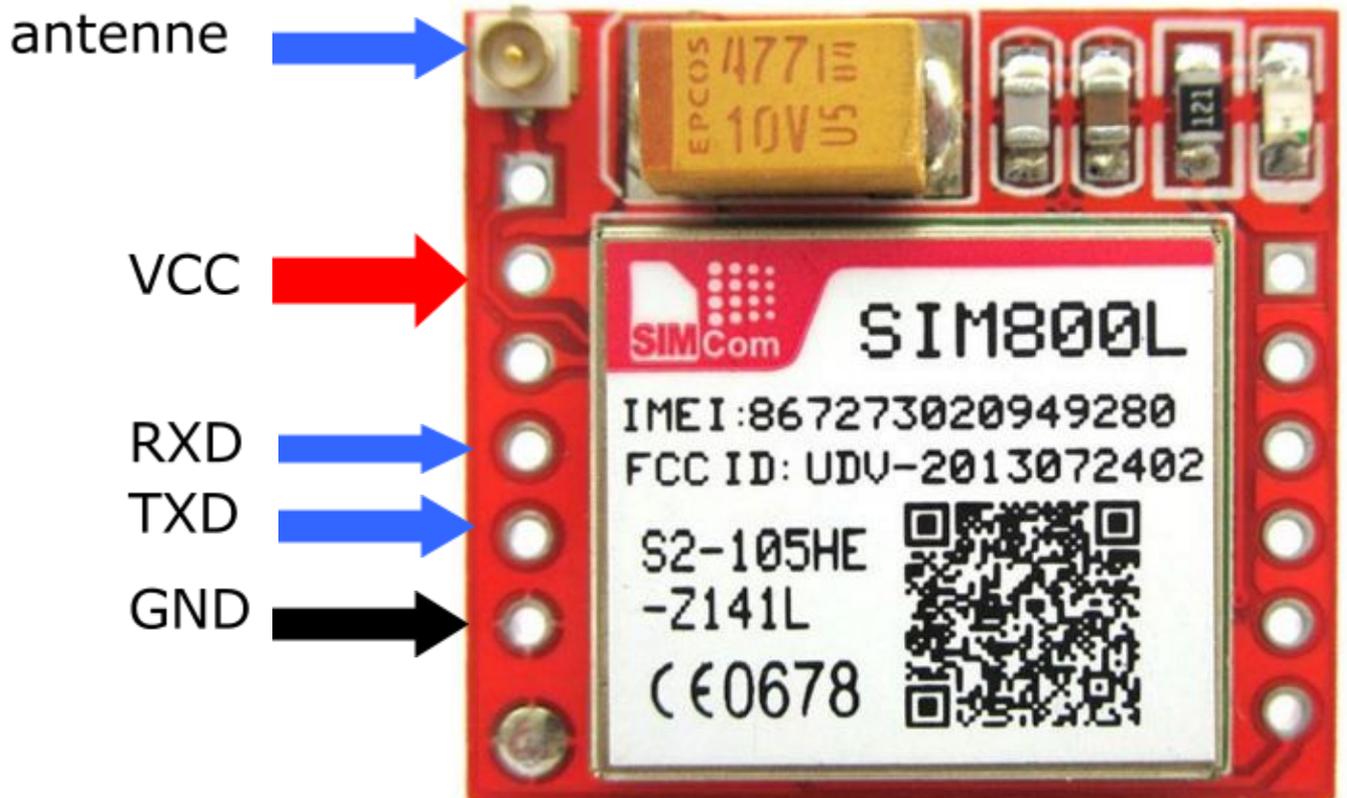
---

Pour envoyer des SMS via le 18F4580, nous avons tout d'abord étudié le fonctionnement du module SIM800L. Ensuite Nous avons appris à utiliser les commandes AT, indispensable pour communiquer avec le module SIM, pour au final les intégrer à notre programme.

### 1) Le module SIM800L

Le module SIM800L est module GSM (Global System for Mobile communication) qui va nous permettre d'envoyer au stand, via SMS, les données du véhicule.

Le module comporte 12 pins mais dans notre cas nous n'utiliserons que 4 pins : VCC, RXD, TXD et GND.



Les pins VCC et GND : ils servent à l'alimentation du module. Celui-ci doit avoir une tension d'alimentation entre 3,3 V et 4,4 volt.

Le pin RXD : il sert à la réception des bits de données.

Le pin TXD : il sert à la transmission des bits de données.

Il reçoit les informations via une liaison série soit directement d'un ordinateur, soit d'un microcontrôleur. Dans notre cas, nous avons tout d'abord commencé à communiquer avec le module depuis un ordinateur (à l'aide du logiciel *Putty*) pour nous familiariser avec son fonctionnement, puis avons œuvré pour le faire travailler avec notre microcontrôleur.

Pour fonctionner, il doit recevoir des lignes de commandes spécifiques : les commandes AT (ou commande Hayes).

## 2) Les commandes AT

Les commandes AT (ou commandes Hayes) compose un langage de commande qui est utilisé dans le milieu de la communication et des réseaux de communication. Elles permettent entre autre :

- De composer un numéro de téléphone
- De commander le raccordement d'un modem à la ligne
- De connaître l'état de la ligne

Ces lignes de commandes se caractérisent par le fait qu'elles commencent toute par « AT » suivie de l'ordre. Les commandes sont à envoyer dans un certain ordre pour envoyer ou recevoir un SMS.

- ***L'envoi de « AT » permet de vérifier la communication entre le module et l'ordinateur/ microcontrôleur. La réponse « OK » du module signifie que la liaison est bonne.***
- ***« AT+CPIN=xxxx » permet de rentrer le code PIN de la carte SIM. On attend ensuite les réponses « OK », « SMS ready » et « call ready »***
- ***« AT+CMGF=1;&w » permet d'initialiser le mode SMS. Cette commande doit être utilisée à chaque envoi. Si la commande est***

**envoyer sous la forme « AT+CMGF= 'numéro' » cela nous permet de lire le sms à l'emplacement indiqué.**

**« AT+CMGS= 'xxxxxxxxxx' » nous permet d'entrer le numéro de téléphone du destinataire. Il faut ensuite écrire son message.**

```
AT
OK
AT+CPIN=1234
OK

+CPIN: READY

Call Ready

SMS Ready
AT+CMGF=1;&w
OK
```

*Moniteur de Putty lors de l'envoi des commandes AT*

```
AT+CMGS="0643212319"
> coucou
+CMGS: 45

OK
```

Après nos tests sur *Putty*, nous avons travaillé sur l'envoi des SMS de façon automatisé grâce un microcontrôleur Microship PIC 18F4580.

## II. Le travail réalisé

---

### 1) Le programme en C

Nous avons défini les fonctions du programme C comme suit:

- Collecter les données
- Les encapsuler pour pouvoir les réceptionner facilement
- Envoyer les données

Nous étions parti sur une solution pour lire les SMS avec le PIC18 mais puisque nous ne comptons plus intégrer le lecteur de carte SD cette solution nous était inutile.

Nous nous sommes donc limité à l'envoi de données via SMS.

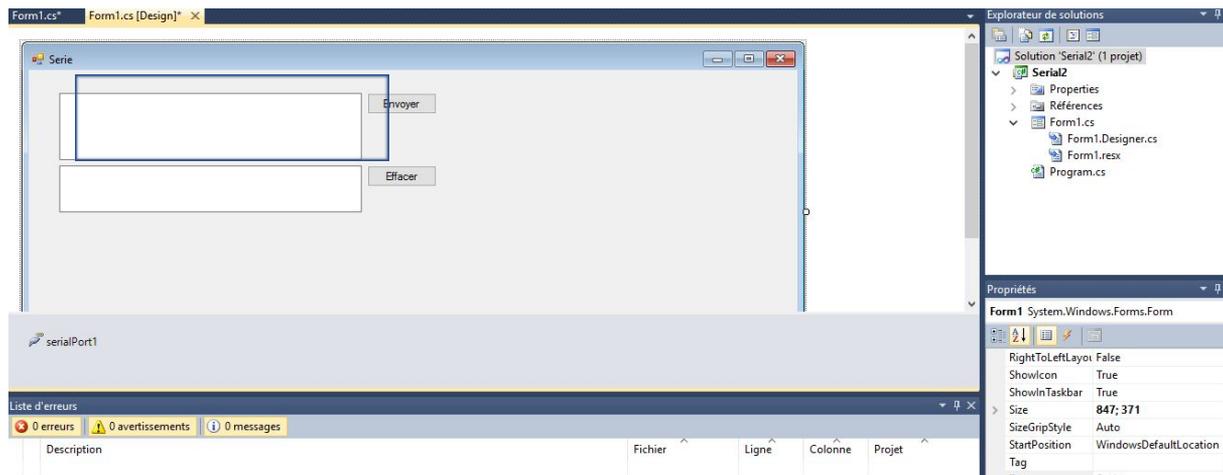
Pour étoffer cette partie il aurait fallu choisir les capteurs adapter à la voiture écoshell et ainsi encapsuler ces données spécifique.

### 2) Le programme en C#

Nous avons décidé de créer un programme ressemblant au logiciel PUTTY mais sans passer par une configuration et une interface graphique plus jolie, simplifiée mais permettant de faire beaucoup de choses. Nous avons choisi d'utiliser le langage C# comme langage de programmation. Ce langage a été

créée par Microsoft en 2002. Il permet de coder sur une interface graphique pour ajouter des boutons, des fenêtres etc. Le logiciel va automatiquement générer le code en lien avec la partie graphique dans lequel on peut y écrire des fonctions.

Nous avons trouvé un exemple sur internet de communication série grâce au C# que voici :



Explication de son comportement :

La zone du dessus contient les commandes à envoyer (AT). Nous avons donc le bouton envoyer en dessous. La zone du bas va nous informer à propos de l'ouverture du port au début. Puis elle va nous transmettre les réponses du module à nos commandes.

La partie « serialPort1 » définit le port série à utiliser. Nous avons donc configuré celui-ci en fonction du port COM sur laquelle il était, dans notre cas le port COM4. Nous devons définir aussi la vitesse. Pour communiquer via SMS nous avons trouvé qu'il fallait choisir 115200 Bauds par secondes.

Le bouton effacer permet d'effacer toutes les commandes de retour situées dans la zone du bas.

Penchons-nous ensuite sur la partie code. Il se génère automatiquement.

Le Logiciel utilisé est Visual Studio dans sa version 2017. Outil mis à disposition gratuitement par Microsoft et très complet.

Les lignes comportant « Using. System » permettent d'utiliser les types de fonctions en lien avec cette ligne. Comme par exemple la ligne « Using System.Text » permet d'utiliser des fonctions de types textes.

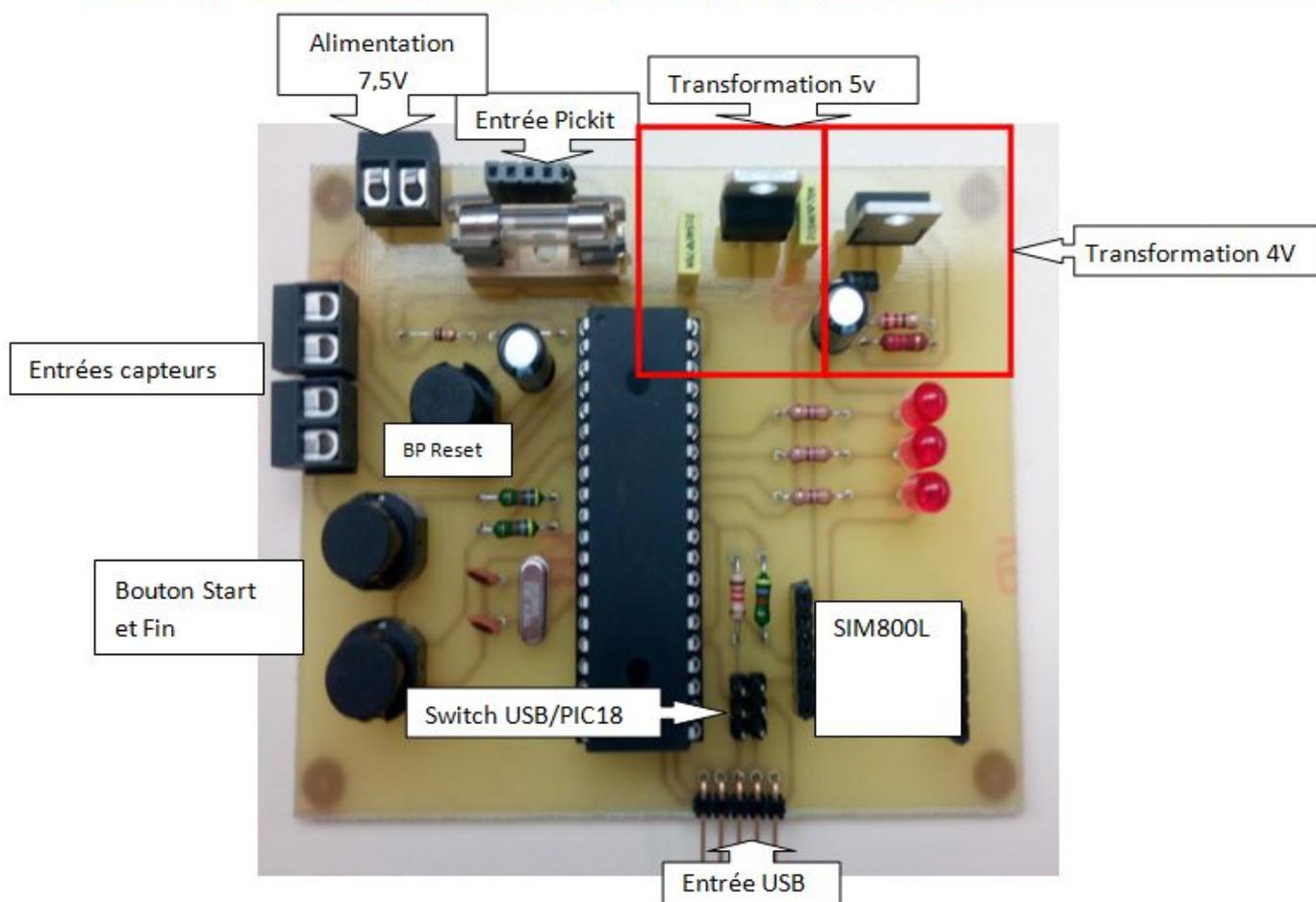
Ensuite vient la partie principale à partir de la ligne 12. Le code qui le suit contient le programme principal. Nous avons donc un test conditionnel d'ouverture du port série.

Passons à la seconde partie du code. A partir de la ligne 41 nous avons le code pour le bouton Effacer puis la fermeture du Port Série

A la ligne 62, le code permet d'envoyer le texte entré dans la zone de texte en tant que commande sur le port série.

Nous avons tenté d'automatiser les commandes d'initialisation de la SIM en arrière-plan au lancement sans réussite à la fin du projet.

### 3) La carte électronique



### **III. La suite du projet**

Le projet nécessite encore des améliorations et des perspectives de projet peuvent être envisagées :

- La carte doit être revue pour devenir totalement fonctionnel. En effet la carte a quelques problèmes au niveau du câblage et n'est pas parfaitement fonctionnel.
- Le programme doit être adapté à la carte. Nous devons adapter le

- programme correctement pour qu'il envoie ou lise les SMS
- L'application graphique doit être terminée. L'application C# fonctionne mais ne permet pas encore de récupérer les informations du SMS pour les classer dans un graphique ou un tableur.
  - L'acquisition des données doit être intégrée au programme. Même si le programme peut envoyer des SMS, il est encore incapable de réaliser des mesures de tension, courant ou puissance pour les envoyer par SMS
  - Les capteurs doivent encore être définie

### **Conclusion:**

Pour conclure, le projet nous a permis de mettre en pratique les connaissances théoriques apprises en cours. Nous avons appris à travailler en groupe et nous organiser. Ce projet nous a permis également de faire face à de problèmes et de trouver des solutions ensemble pour les résoudre.

Remerciements :

Nous tenons a remercier L'IUT de Belfort-Montbéliard pour tout le materiel mis a disposition pour mettre en oeuvre ce projet et de nous proposer des projets variés et interessant.