

**RAPPORT DU PROJET TUTORE**

Traceur GPS/GSM

Tuteur :

Youcef TOUATI

Binôme :

Rida AMZIL

Yasar HERGUNER

Sommaire :

[1. Introduction : 4](#_Toc441012621)

[2. MÉTHODOLOGIE et ORGANISATION DU TRAVAIL: 5](#_Toc441012622)

[2.1. Étude du projet : 5](#_Toc441012623)

[2.1.1. Le projet en deux grandes parties : 6](#_Toc441012624)

[3. Recherche générale : 7](#_Toc441012625)

[3.1. Traceur GPS /GSM 7](#_Toc441012626)

[3.1.1. La partie client : 7](#_Toc441012627)

[3.1.2. La partie serveur : 8](#_Toc441012628)

[3.2. Composant d’un traceur GPS/GSM : 8](#_Toc441012629)

[3.2.1. Localiser la position avec un GPS : 8](#_Toc441012630)

[3.2.2. Le Protocole NMEA : 8](#_Toc441012631)

[3.2.3. Les trames NMEA 0183 : 8](#_Toc441012632)

[3.2.4. L’envoi des SMS avec le module GSM : 10](#_Toc441012633)

[3.2.4.1. Les Commandes AT : 10](#_Toc441012634)

[3.2.4.2. Structure d’une commande AT : 10](#_Toc441012635)

[3.2.4.3. Commandes de base : 11](#_Toc441012636)

[3.2.4.4. Commandes General : 11](#_Toc441012637)

[3.2.4.5. Les commandes utilisées pour envoyé SMS : 11](#_Toc441012638)

[3.2.5. Les microcontrôleurs : 11](#_Toc441012639)

[3.2.5.1. Le microcontrôleur vu global : 11](#_Toc441012640)

[3.2.5.2. Structure interne d'un microcontrôleur : 12](#_Toc441012641)

[3.2.5.3. Les microcontrôleurs PIC24F : 13](#_Toc441012642)

[3.2.5.4. GPIO General Purpose Input/Output : 14](#_Toc441012643)

[3.2.5.5. Les interruptions : 14](#_Toc441012644)

[3.2.6. Protocole de communication des liaisons série : 15](#_Toc441012645)

[3.2.6.1. Protocole de communication  série UART/USART: 15](#_Toc441012646)

[3.2.6.2. Protocole de communication  série SPI : 15](#_Toc441012647)

[3.3. Recherche technique : 16](#_Toc441012648)

[3.3.1. Environnements de développement : 16](#_Toc441012649)

[3.3.1.1. La carte Mikromedia PIC 24 : 16](#_Toc441012650)

[3.3.1.2. Composons du Mikromedia PIC24 : 16](#_Toc441012651)

[3.3.1.3. Préparation de la carte Mikromedia PIC24: 17](#_Toc441012652)

[3.3.1.4. Choix d’outil de développement : 17](#_Toc441012653)

[3.3.1.4.1. MikroC Pro : 17](#_Toc441012654)

[3.3.1.4.2. Mplab IDE X: 18](#_Toc441012655)

[3.3.1.4.3. Sonde de programmation à utiliser : 18](#_Toc441012656)

[3.3.2. Algorithmes : 19](#_Toc441012657)

[3.3.2.1. Diagramme général traceur GPS/GSM 19](#_Toc441012658)

[3.3.2.2. Réception du tram GPS et le découpage : 19](#_Toc441012659)

[3.3.2.3. Traitement de la position et l’envoi par SMS : 20](#_Toc441012660)

[4. Perspective : 20](#_Toc441012661)

[5. Conclusion : 21](#_Toc441012662)

1. Introduction :

Dans le cadre de ce cours qui est intitulé “projet tutoré nous devions fournir un travail en binôme concernant un projet assez conséquent avec l'aide d'un tuteur qui nous accompagnera tout au long du projet.Notre tuteur désigné est Monsieur TOUATI et notre projet et la réalisation d'un traceur GPS/GSM sur une carte.

Notre sujet est donc la programmation d’un traceur GPS/GSM sur la carte Mikromedia PIC24, qui permet la localisation de la position ainsi que l’itinéraire d’une personne ou d’un véhicule. Sa envoie également la position sous lien Google Map par SMS.

Les traceurs GPS/GSM sont de plus en plus utilisés par les particuliers, mais aussi les entreprises. Par exemple on peut s'en servir pour localiser la position d’une personne qui souffre de la maladie d’Alzheimer ou bien dans une entreprise pour localiser ses véhicules de service.

Nous verrons, dans un premier temps, les méthodologies et l'organisation de travail, mais également les recherches effectuées de manière générale sur le projet.

La seconde partie portera sur les recherches techniques faites notamment sur l’environnement de développement avec la carte Mikromedia PIC24.

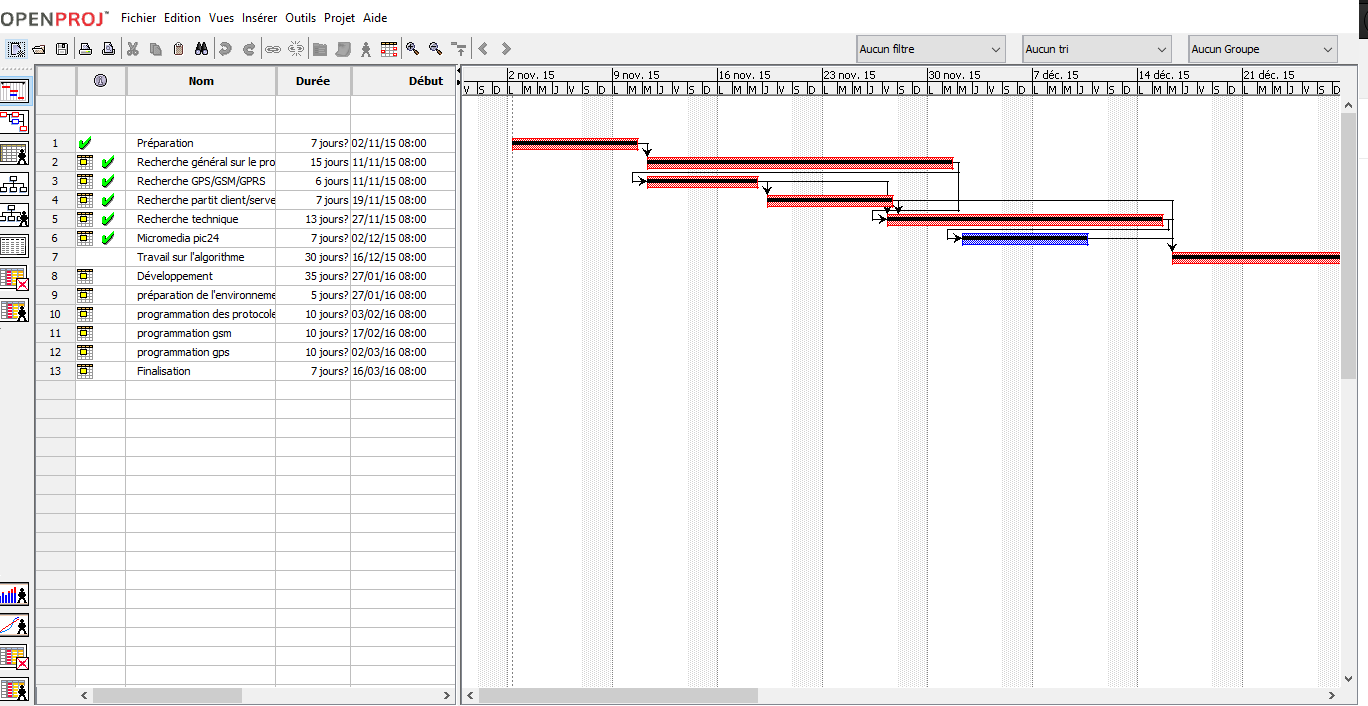
Enfin nous verrons dans un troisième point, la préparation des algorithmes, mais aussi les perspectives envisager pour la deuxième partie du projet.

1. MÉTHODOLOGIE et ORGANISATION DU TRAVAIL:
   1. Étude du projet :

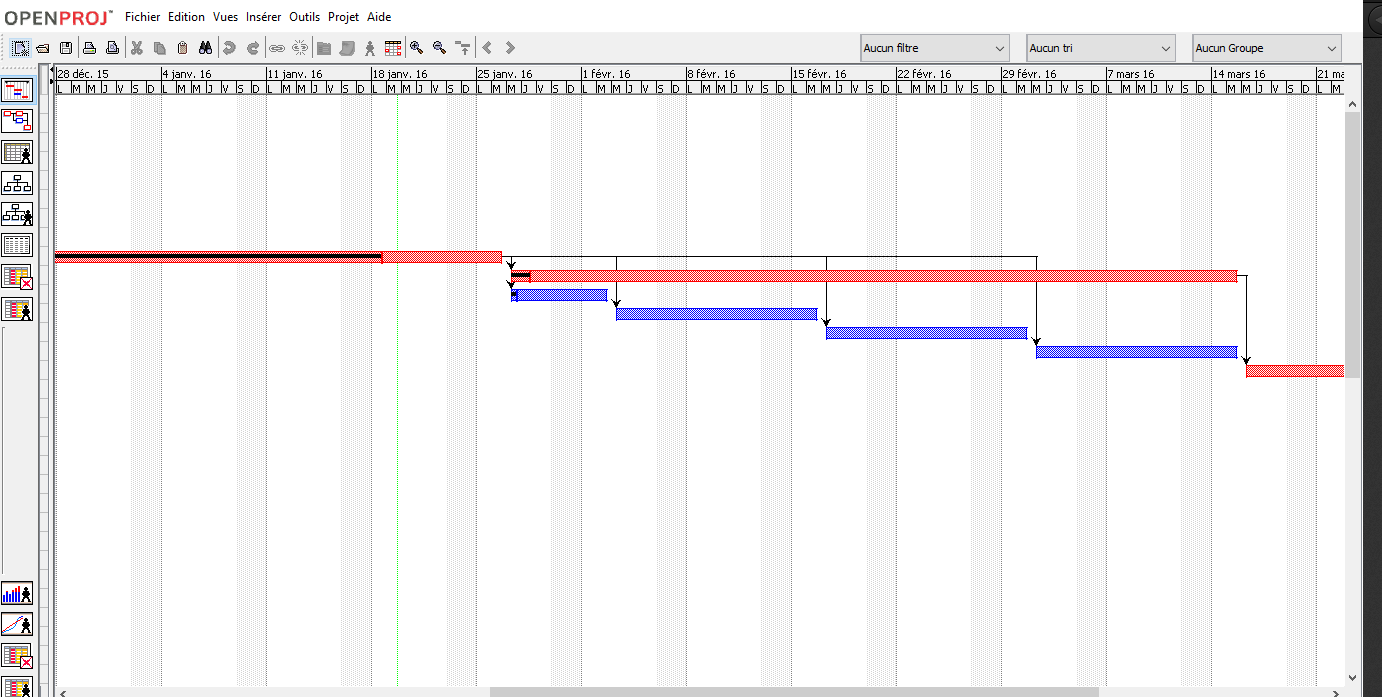
Sur ce projet effectuer en binôme, l'important était de bien gérer le projet et distribuer les tâches pour être efficace.Pour ce faire nous avons utilisé Openproject. Cet outil permet la gestion des taches ainsi que la gestion du projet facilement.Ainsi, nous avons pu établir un calendrier prévisionnel dans laquelle nous avons plusieurs étape détermine (recherche, développement, finalisation etc..) qui nous permet de nous diriger et d'être dans les délais.

Nous nous sommes donc distribuer les taches en première partit. Yasar est parti dans la recherche général du fonctionnement des traceurs GPS/GSM, mais aussi des microcontrôleurs PIC24 et des protocoles de communication. D’autre part Rida a fait des recherches générales sur les protocoles de NMEA (Protocole de localisation GPS), ainsi que sur la partie GSM.Mais Aussi la recherche et la préparation de l’environnement de développement, compatible avec la carte Mikromedia PIC24.

Enfin il y avait une partie pratique, plus hardware, dont on a travaillé tous les deux. Cette partie consiste à souder les différentes composons de la carte.



**Figure 1 : Organisation du projet avec Open Project**



**Figure 2 : Organisation du projet avec Open Project**

* + 1. Le projet en deux grandes parties :

Le sujet nous a demandé tout d'abord un temps de recherche important afin de comprendre le fonctionnement des traceurs GPS ainsi que la carte Mikromedia pic24 proposer. Les microcontrôleurs et les périphériques qui lui seront interfacés. Notre travail sur le projet comprend deux parties. Une première recherche générale sur les traceurs GPS/GSM, sur les microcontrôleurs, les protocoles de communication, le protocole NMEA (GPS), ainsi que la partie GSM.

On a également essayé en parallèle aussi de préparer l’environnement de développement pour être on avance sur le projet.

Pour la 2ème partie de ce projet, on va consacrer tout notre temps dans le développement et le codage. De plus, on va tenter de trouver une solution pour le passage du programme vers la carte pour exécuter le code compilé ainsi que le débogage des fonctions exécuté.

1. Recherche générale :
   1. Traceur GPS /GSM

Un traceur GPS/GSM permet de localiser rapidement la position d’une personne ou d’un véhicule dans lequel il sera installé, ou suivre l’itinéraire de l’objective viser. Il permet aussi d’envoyer la position géographique en cas d’urgence pas SMS, cette action est déclenchée sois par l’utilisateur ou même par l’administrateur en envoyant un message SMS a la destination, et ce dernier répond avec un message SMS qui contient le lien google map. En même temps le traceur envoie la position à un serveur à chaque durée prédéfini. Et le serveur à son tour enregistre chaque position et avec il pourra crée l’itinéraire de l’utilisateur.

**Figure 3 : Schéma expliquant le principe d'un traceur GPS/GSM**

* + 1. La partie client :

Cette partie représente la partie émotteuse qui va envoyer la position sois par SMS ou internet GRPS.

Dans cette partie en peux avoir plusieurs fonctions :

* Envoyer la position GPS à un serveur Web qui va enregistrer la position à chaque durée prédéterminé.
* SMS alerte, déclenché si l’utilisateur a besoin d’aide.

On peut avoir encore plus de fonctions complémentaires :

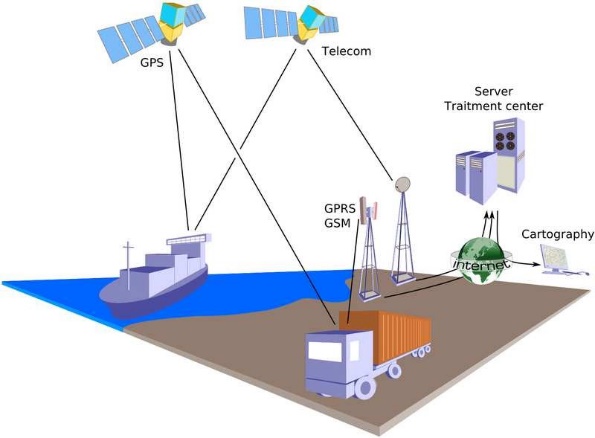
* Déterminer un diamètre à ne pas dépasser, sinon l’administrateur reçoit une alerte.
* Appeler directement la personne en cas d’urgence.
  + 1. La partie serveur :

Cette partie représente la partie administration (serveur). Dans cette partie en va traitais les informations reçu par le traceur (GPS / GSM). Ces informations seront envoyées par internet (GPRS / 3G) a un serveur web qui va les stocker afin de les utiliser pour dessiner l’itinéraire.

* 1. Composant d’un traceur GPS/GSM :

Un traceur GPS/GSM, permet de transmettre une position GPS via une liaison GSM dans notre exemple. Pour cela on peut dire qu’un traceur contiens 3 composons principale. Un **module GPS** pour avoir la position, un **module GSM** pour envoyer la position sois pas SMS ou par connexion internet GPRS/3G et en fin un **microcontrôleur** pour gérer le programme principale.

* + 1. Localiser la position avec un GPS :

Le Global Positioning System (GPS) – que l'on peut traduire en français par « système de localisation mondial » ou plus proche du sigle d'origine, « Guidage par Satellite » – est un système de géolocalisation fonctionnant au niveau mondial. Cela permet de positionner un objet (une personne, etc.) sur un plan ou une carte à l'aide de ses coordonnées géographiques.

* + 1. Le Protocole NMEA :

La norme NMEA 0183 est une spécification pour la communication entre équipements marins, dont les équipements GPS. Elle est définie et contrôlée par la National Marine Electronics Association (NMEA), association américaine de fabricants d'appareils électroniques maritimes, basée à Severna Park en Maryland (Etats-Unis d'Amérique).

La norme 0183 utilise une simple communication série pour transmettre une "phrase" à un ou plusieurs écoutants. Une trame NMEA utilise tous les caractères ASCII.

* + 1. Les trames NMEA 0183 :

Il existe plus d'une trentaine de trames GPS différentes. Le type d'équipement est défini par les deux caractères qui suivent le $. Le type de trame est défini par les caractères suivants jusqu'à la virgule.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| $GPGGA,064036.289,4836.5375,N,00740.9373,E,1,04,3.2,200.2,M,,,,0000\*0E | | | | |
| $ | GP | GGA | 064036.289,4836.5375,N,00740.9373,E,1,04,3.2,200.2,M,,,,0000\* | 0E |

La trame se terminer par une somme de contrôle qui permet de vérifier que la trame n'a pas été endommagée avant sa réception. Dans ce cas c'est à dire (0E).

Tableau des trames supporté par le module GPS GTPA013 utilisé dans le développement du programme traitement et découpage des trames reçu.

|  |  |
| --- | --- |
| Options | Description |
| GGA | Heure, la position et des données de type correctif. |
| GSA | le mode de fonctionnement du récepteur, les satellites actifs utilisés et la valeur de DOP. |
| GSV | Le nombre de satellites GPS en vue, Numéros d'identification satellites, l'altitude, l'azimut et valeur SNR. |
| RMC | Heure, date, position et vitesse.se mode est recommandé un minimum d’informations de navigation. |
| VTG | pour Direction (cap) et vitesse de déplacement (en noeuds et Km/h). |

* Exemple des trames les plus recommandées :

$GPGGA,064036.289,4836.5375,N,00740.9373,E,1,04,3.2,200.2,M,,,,,0000,\* 0E

|  |  |
| --- | --- |
| $GPGGA | Type de trame |
| 064036.289 | Trame envoyée à 06h40m36,289s (heure UTC) |
| 4836.5375 | Latitude 48,608958° Nord = 48°36'32.25" |
| N | Nord |
| 00740.9373 | Longitude 7,682288° Est = 7°40'56.238" |
| E | Est |
| 1 | Type de positionnement (le 1 est un positionnement GPS) |
| 04 | Nombre de satellites utilisés pour calculer les coordonnées |
| 3.2 | Précision horizontale ou HDOP (Horizontal dilution of precision) |
| 200.2 | Altitude 200,2 |
| M | mètres |
| ,,,,,0000 | D'autres informations peuvent être inscrites dans ces champs |
| \*0E | Somme de contrôle de parité, un simple XOR sur les caractères précédents |
| <CR><LF> | Fine trame |

* Une autre trame très courante pour les bateaux est la RMC, qui donne l'heure, la latitude, la longitude, la date, ainsi que la vitesse et la route sur le fond, mais pas l'altitude.

|  |  |
| --- | --- |
| $GPRMC | type de trame |
| 053740.000 | Heure UTC exprimée en hhmmss.sss : 5h37m40s |
| A | état A=données valides, V=données invalides |
| 2503.6319 | Latitude exprimée en ddmm.mmmm : 25°03.6319' = 25°03'37,914" |
| N | indicateur de latitude N=nord, S=sud |
| 12136.0099 | Longitude exprimée en dddmm.mmmm : 121°36.0099' = 121°36'00,594" |
| E | indicateur de longitude E=est, W=ouest |
| 2,69 | vitesse sur le fond en nœuds |
| 79,65 | route sur le fond en degrés |
| 100106 | Date exprimée en ddmmyy : 10 janvier 2006 |
| , | déviation magnétique en degrés (souvent, vide pour un GPS) |
| , | sens de la déviation E=est, W=ouest (souvent vide pour un GPS) |
| A | mode de positionnement A=autonome, D=DGPS, E=DR |
| \*53 | somme de contrôle de parité |
| <CR><LF> | Fine trame |

* + 1. L’envoi des SMS avec le module GSM :

La norme GSM autorise un débit maximal de 9,6 kbps, ce qui permet de transmettre la voix ainsi que des données numériques de faible volume, par exemple des messages textes (SMS, pour Short Message Service) ou des messages multimédias (MMS, pour Multimedia Message Service).

En Europe, le standard GSM utilise les bandes de fréquences 900 MHz et 1800 MHz.

L’utilisation du module GMS dans un Traceur GPS permet d’avoir la possibilité d’envoyer des messages d’alerte de position, envoyer des données avec le GPRS, passer des appels …

* + - 1. Les Commandes AT :

Les commandes AT sont définies dans la norme GSM 07.07 (pour les SMS cf. GSM 07.05). AT est l’abréviation de « Attention ». Ces 2 caractères sont toujours présents pour commencer une ligne de commande sous forme de texte (codes ASCII). Les commandes permettent la gestion complète du module GSM

* + - 1. Structure d’une commande AT :

AT+CMD= ?<CR> test la commande +CMD

La réponse sera : <CR><LF>OK<CR><LF>) AT+CMD<CR> applique la commande +CMD

* + - 1. Commandes de base :

ATI[<valeur>] : Affichage des paramètres du TA.

ATZ[<valeur >] : Chargement des paramètres par défaut mémorisés par l’utilisateur.

AT&F[<valeur >] : Chargement des paramètres par défaut mémorisés par le fabriquant.

ATE[<valeur >] : Echo des commandes AT (déf. 1).

ATQ[<valeur >] : Suppression du résultat (déf. 0).

* + - 1. Commandes General :

AT+CGMI : Demander la marque du fabricant. AT+CGMM : Demander le modèle.

AT+CGMR : Demande d'identification de révision.

AT+CGSN : Demander le numéro de série d'identification du produit.

* + - 1. Les commandes utilisées pour envoyé SMS :

AT+ CMGF=<valeur> : Message format.

valeur = 0 (mode PDU activer par défaut). valeur = 1 (mode texte).

AT+CMGS=<Numéro\_destination>,<type><CR><message><cntr+z>

<type> : Sélection du type des numéros de téléphone.

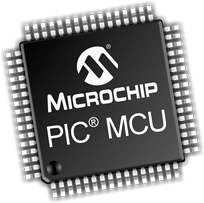
( 145 : Numérotation internationale (avec +). 129 : Les autres cas. )

<Numéro\_destination> : le Numéro de téléphone qui va recevoir le message SMS.

<message> : le message texte à envoyer.

<cntr+z> : fin du message et envoi du message à destination.

* + 1. Les microcontrôleurs :
       1. Le microcontrôleur vu global :

Les microprocesseurs possèdent un indéniable avantage sur la logique câblée, en effet pour modifier le fonctionnement d'une application il suffit de modifier le programme sans refaire de câblage.

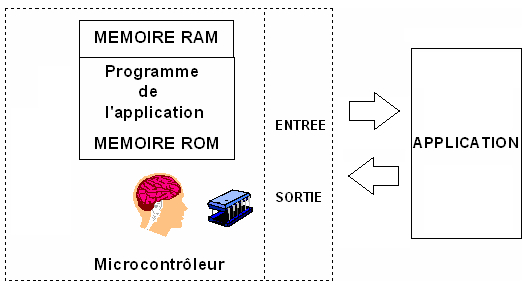
Les microcontrôleurs possèdent quant à eux la puissance d'un microprocesseur, mais ont un atout en plus, en effet ils possèdent dans le même boîtier, les périphériques intégrés.

Cela veut dire que le programme de l'application est en interne et non plus dans un circuit mémoire externe et que les périphériques d’entrées - sorties sont également intégrés, ce qui fait l’économie de nombreux circuits périphériques.

Cette caractéristique fait que les montages deviennent encore plus simples et la

programmation plus aisée (un système à base de microprocesseur, oblige le concepteur à réaliser un décodage d'adresse pour permettre au microprocesseur de ne dialoguer qu’avec un seul périphérique à la fois).

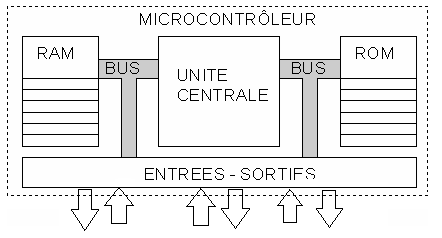
Un microcontrôleur seul peut donc gérer une application, sans faire appel forcément à d'autres circuits associés.



Sur la figure précédente le microcontrôleur possède en interne la mémoire programme contenant le programme de l'application ainsi que le port d’entrées - sorties qui vont permettre au microcontrôleur de s’interfacer avec l'application.

On le voit ici, par rapport au schéma à base de microprocesseur, on a encore franchi un degré d’intégration en rassemblant tous les circuits nécessaires au fonctionnement d'une application dans le même boîtier.

* + - 1. Structure interne d'un microcontrôleur :

Un microcontrôleur le plus simple qu'il soit possède au minimum les éléments suivants :

Une unité centrale qui est le cœur du système, également appelé CPU pour Central Processing Unit, dans cette unité centrale nous retrouverons plusieurs éléments telle que l’unité arithmétique et logique (UAL) que nous détaillerons dans un prochain numéro.

Une mémoire contenant le programme à exécuter par le microcontrôleur, généralement appelée mémoire morte ou ROM ( Read Only Mémory ) , mémoire à lecture seule. Cette mémoire a la particularité de sauvegarder en permanence les informations qu'elle contient, même en absence de tension ( ce qui est primordiale, sinon il faudrait reprogrammer le microcontrôleur à chaque remise sous tension ! ).

Une mémoire vive également appelée RAM ( Random Access Mémory ), cette mémoire permet de sauvegarder temporairement des informations. Il est à noter que le contenu d'une RAM n'est sauvegardé que pendant la phase d'alimentation du circuit. Le microcontrôleur pourra utiliser cette mémoire pour stocker des variables temporaires ou faire des calculs intermédiaires.

Un port d’entrées - sorties permettant au microcontrôleur de dialoguer avec l'extérieur, par exemple prendre l’état d'un capteur , d'un interrupteur ou bien pour allumer un LED ou piloter un relais ( via un transistor bien sûr ).

Des bus internes permettent la communication entre les différents éléments intégrés au microcontrôleur. Il existe trois sortes de bus que l'on détaillera dans un prochain cours.

* + - 1. Les microcontrôleurs PIC24F :

De nombreux fabricants se sont implantés sur le marché des microcontrôleurs.

La société Américaine Microchip technologie à mis au point le microcontrôleur PIC24.

Ce microcontrôleur très utilisé à l’heure actuelle dans le domaine embarqué est un compromis entre simplicité d'emploi et prix de revient.

Il existe de nombreuses versions de PIC possédant chacune des caractéristiques différentes, des tableaux comparatifs permettent de choisir le PIC le plus adéquat par rapport à l'application envisagée.

Dans notre cas, la carte est équipée d’un PIC14F 16Bit et voilà les caractéristiques les plus importantes :

* **CPU :**
* 16 \* 16bits registres de travail (Working Register Array)
* Multiplication 17bits \* 17bits en un cycle
* Division 32bits sur 16bits en un cycle
* Jeu d'instruction optimisé pour la compilation de programme en C :
* 76 instructions de base (déclinable en beaucoup plus)
* **Mode d'adressage flexible**
* Oscillateurs :
* Interne de 8Mhz (FRC) et de 31kHz (LPRC)
* PLL x4 pour sources internes ou externes
* Sélection de la source d'oscillation CPU par logiciel
* Postdiviseur d'oscillation CPU contrôlable par logiciel (réduction de la consommation)
* Surveillance de la source d'oscillation (Fail-Safe Clock Monitor : FSCM)
* Mémoire programme linéaire permettant d'accéder jusqu'à 12Mo
* Mémoire de données (RAM) permettant d'adresser jusqu'à 64Ko
* **Général :**
* Tension d'alimentation comprise entre 2.0V et 3.6V
* Interface JTAG permettant entre autre la programmation
* Périphériques embarqués :
* 2 modules 3-wire/4-wire SPI
* 2 modules I²C
* 2 modules UART :
* RS-232, RS-485, LIN 1.2
* Support IrDA externe ou interne
* Auto-Wake-Up àa la réception d'un bit de Start
* Auto-Baud detect
* Pile FIFO à 4 niveaux
* Port parallèle Maitre / Esclave (PMP/PSP)
* 8bits ou 16bits de données
* 16bits d'adresses
* Horloge temps réél (Hardware Real-Time Clocl/Calendar : RTCC)
* Heures, minutes, secondes, années, mois, jours avec alarme paramétrable
* Génération de CRC
  + - 1. GPIO General Purpose Input/Output :

Les ports GPIO (General Purpose Input/Output, c'est-à-dire entrée/sortie pour un usage général) sont des ports d'entrée/sortie très utilisés dans le monde des microcontrôleurs, en particulier dans le domaine de l'électronique embarquée. Selon la configuration, ces ports peuvent fonctionner aussi bien en entrée qu'en sortie.

Les périphériques GPIO comportent un ensemble de ports d'entrée/sortie qui peuvent être configurés pour jouer soit le rôle d'une entrée, soit le rôle d'une sortie. Lorsqu'un port GPIO est configuré en tant que sortie, on peut écrire dans un registre interne afin de modifier l'état d'une sortie. Lorsqu'il est configuré en tant qu'entrée, on peut détecter son état en lisant le contenu d'un registre interne. De plus, les périphériques GPIO peuvent produire des interruptions et des événements d'accès direct à la mémoire (EDMA).

* + - 1. Les interruptions :

Une interruption est une action, déclenchée par un événement précis, qui va arrêter le programme principal, s'exécuter, puis reprendre le programme principal à l'endroit où il s'était arrêté. Les interruptions peuvent être déclenchées par:

• Les timers

• Les pins interruptibles

• Les interfaces de communication

• Le convertisseur analogique-numérique

Les interruptions sont fondamentales dans le fonctionnement du PIC car elles permettent d'effectuer plusieurs choses à la fois, et de définir des priorités dans les différentes tâches à accomplir.

Pour utiliser les interruptions, il faut tout d'abord les activer. Il faut pour cela fixer les valeurs des registres suivants :

La routine d'interruption est structurée de la façon suivante:

• Sauvegarder les registres

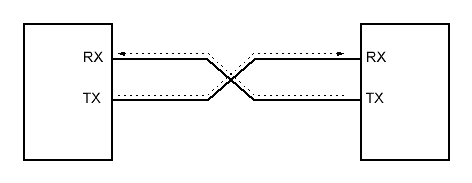
• Identifier la source de l'interruption

• Effectuer l'action associée à cette interruption

• Restaurer les registres

• Revenir au programme principal

* + 1. Protocole de communication des liaisons série :
       1. Protocole de communication  série UART/USART:

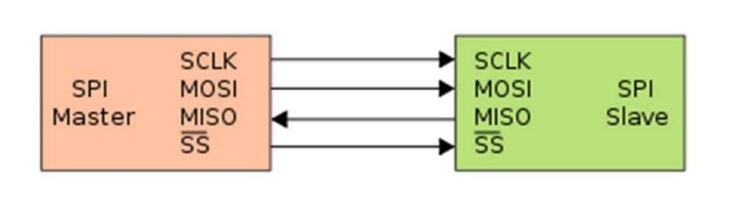
La liaison série est, en télécommunications et en informatique, l'action d'envoyer des données bit par bit sur un canal de communication ou un bus informatique. L'octet à transmettre est envoyé bit par bit par l'émetteur sur la ligne Tx, vers le récepteur sur la ligne Rx qui le reconstitue. La vitesse de transmission de l'émetteur doit être identique à la vitesse d'acquisition du récepteur.

Il existe deux principales formes de transmission série : synchrones et asynchrones. Selon les modes prises en charge par le matériel, le nom du protocole de communication comprend habituellement un A si elle prend en charge les communications asynchrones, et un S si elle prend en charge les communications synchrones.

Les deux formes sont :

* UART Universal Asynchronous Receiver / Transmitter.
* USART Universal Synchrone-Asynchrone Receiver / Transmitter.
  + - 1. Protocole de communication  série SPI :

Une liaison SPI (Pour Serial Peripheral Interface) est un bus de données série synchrone qui opère en Full-duplex. Les circuits communiquent selon un schéma maître-esclaves, où le maître s'occupe totalement de la communication. Plusieurs esclaves peuvent coexister sur un bus, la sélection du destinataire se fait par une ligne dédiée entre le maître et l'esclave appelée chip select.

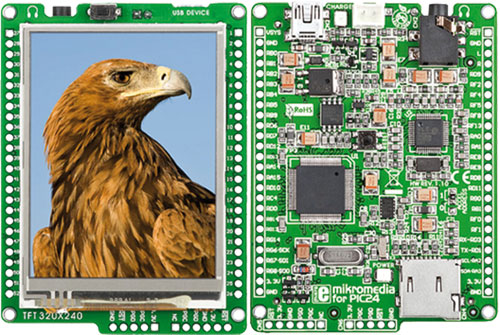


Le bus SPI contient 4 signaux logiques SCLK — Horloge (généré par le maître)

MOSI — Master Output, Slave Input (généré par le maître)

MISO — Master Input, slave Output (généré par l'esclave) SS — Slave select, actif à l'état bas (généré par le maître)

* 1. Recherche technique :
     1. Environnements de développement :
        1. La carte Mikromedia PIC 24 :

Mikromedia PIC24 est un système de développement compact avec beaucoup de périphériques embarqués qui permet le développement de dispositifs avec des contenus multimédias. La partie centrale du système est un microcontrôleur PIC24FJ256GB110 16 bits. Mikromedia PIC24 propose plusieurs fonctionnalités, modules intégrés tels que stéréo codec MP3, TFT 320x240 écran tactile, accéléromètre, un connecteur USB, connecteur audio, lecteur de carte MMC / SD, 8 Mbit mémoire flash, des plots de connexion 2x26 et autres. Il est livré préprogrammé avec bootloader USB, mais peut également être programmé avec des programmeurs externes,comme Mblape Real Ice, mikroProg ™ ou ICD2 / 3. Mikromedia est une plate-forme pratique pour le développement des appareils mobiles.

* + - 1. Composons du Mikromedia PIC24 :

**PIC24FJ256GB110 microcontrôleur**

100-pin 16 bits PIC24 microcontrôleur avec 256 Ko de mémoire programme, 16384 octets de mémoire de données, 16 MIPS et fonctionnement à pleine vitesse USB 2.0 OTG (On-The-Go) Interface conforme.

**Écran couleur TFT**

320x240 pixel écran couleur TFT avec rétroéclairage LED basé sur le contrôleur HX8347D et capable d'afficher 262K couleurs RVB.

**Écran tactile**

Écran tactile de type résistif adapté pour une utilisation avec stylet en plastique ou du bout des doigts pour l'entrée d'utilisateur facile.

**CODEC audio stéréo**

VS1011E décodeur audio avec une interface SPI pour la reproduction rapide des fichiers audio MP3 et WAV.

**Mémoire flash supplémentaire**

M25P80 mémoire flash série avec interface bus SPI 25MHz peut stocker jusqu'à 8Mbits de données et soutient plus de 100.000 effacements / programme de cycles par secteur.

**Interface pour carte microSD**

Utilisez des cartes microSD en option permet de grandes quantités de stockage de données supplémentaires.

**Interface USB**

Connecteur miniUSBsupporte l'interface USB de pour alimenter ou permettre la connexion à des PC et autres périphériques USB.

**ICD2 / 3 programmeurs connecteur**

Une nap de programmation à 6 broches permet de connecter facilement d'autres marques de programmeurs de PIC et les débogueurs en circuit si nécessaire. (Utilisez Conseil de connecteur ICD2 optionnel pour raccorder facilement l'ICD2 de Microchip / 3).

* + - 1. Préparation de la carte Mikromedia PIC24:

Pour avoir la possibilité d’utilisé cette carte, il a fallu faire quelle que branchement et soudure pour préparer la carte pour la partie programmation.

Le kit se compose de 3 parties. La première partie c’est la petite carte mère qui contient tous les composons électronique pour le fonctionnement de la carte. La 2ème partie c’est la partie qui permet de relier d’autre périphérique ou module complémentaire dans note cas les modules GPS et GSM. La 3ème partie ce sont les modules GPS et GMS.

* + - 1. Choix d’outil de développement :

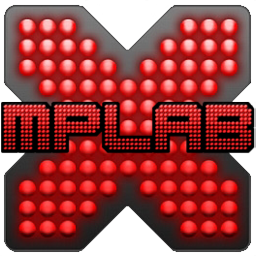
Pour programmer la carte faux avoir un IDE de programmation, un compilateur de code, et enfin une sonde qui permette de passer le programme compilé au microcontrôleur ainsi que le débogage. Pour cela en a le choix entre deux environnements.

* + - * 1. MikroC Pro :

MikroC PRO pour les PIC propre à Mikromedia, est un compilateur ANSI C complet pour les appareils mikromedia qui contiennes des microcontrôleurs PIC de Microchip. Une solution pour développer sur des appareils Mikromedia. Il dispose d’un IDE, compilateur puissant avec des optimisations avancées, beaucoup de bibliothèques de matériel et de logiciels, et des outils supplémentaires qui vous aideront dans votre travail. Compilateur est livré avec dossier et beaucoup d'exemples prêt-à-utiliser, conçu pour vous aider à démarrer en un rien de temps aide complet. Licence de compilateur comprend mises à jour gratuites et un support technique de vie du produit, de sorte que vous pouvez compter sur notre aide, tout en développant.

L’inconvénient de cet outil c’est son prix.

Il faut aussi avoir une sonde pour passer le programme compilé sur la carte.

* + - * 1. Mplab IDE X:

MPLAB X IDE est un logiciel qui s'exécute sur un PC (Windows, Mac OS, Linux) pour développer Des applications pour microcontrôleurs Microchip et contrôleurs de signaux numériques. C'est ce qu'on appelle un environnement de développement intégré (IDE), car il fournit un seul «environnement» intégré pour développer du code pour microcontrôleurs embarqués.

Avantage IDE et compilateur gratuit.

* + - * 1. Sonde de programmation à utiliser :

Dans les deux IDE sois MikroC Pro ou Mplab IDE X faux avoir une programmeuse pour pouvoir programmer la mikromedia pic 24.

* + - **Sous MikroC :**

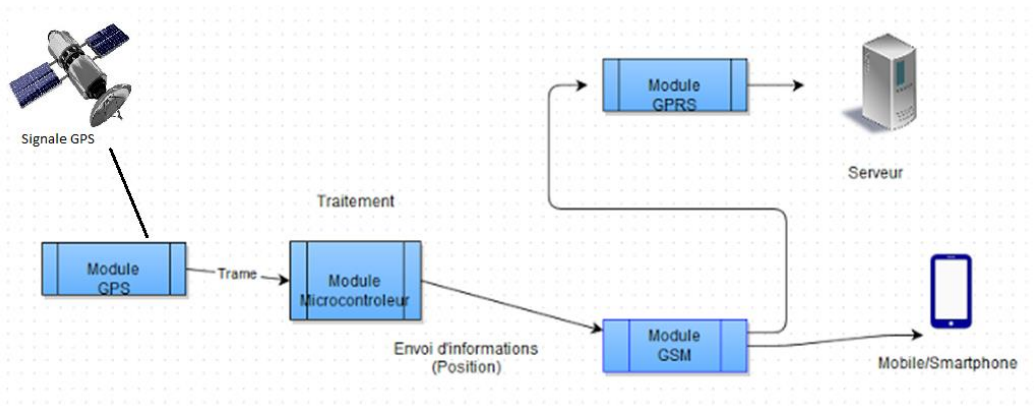
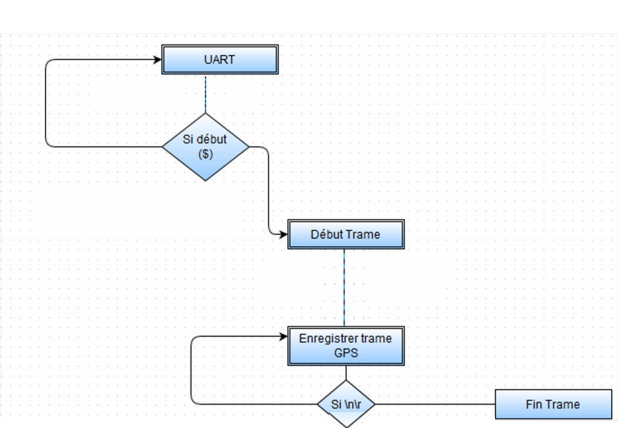
mikroProg ™ est un moyen rapide USB 2.0 programmeur avec le matériel mikroICD ™ In-Circuit Debugger. Intelligent ingénierie permet mikroProg pour soutenir PIC10®, PIC12®, PIC16®, PIC18®, dsPIC30 / 33®, PIC24® et dispositifs PIC32® en un seul programmeur! Il prend en charge plus de 897 microcontrôleurs de Microchip. Des performances exceptionnelles, l'opération facile et un design élégant sont il est top fonctionnalités.

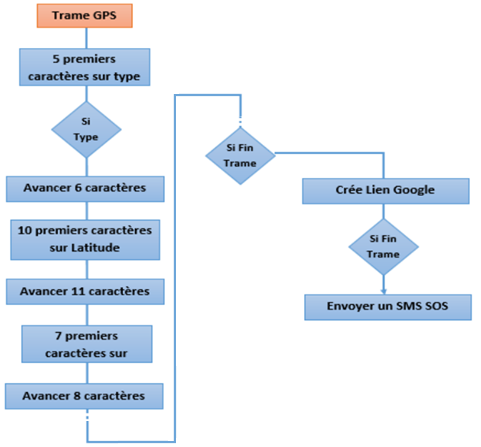
* + - **Sous Mplab IDE X :**

MPLAB® ICD 3 In-Circuit Debugger système est plus rentable de matériel à grande vitesse débogueur / programmeur de Microchip Microchip Flash Digital Signal Controller (DSC) et le microcontrôleur (MCU) appareils. Il permet de déboguer et de programmes de microcontrôleurs PIC® Flash et dsPIC® dscs avec l'interface utilisateur graphique puissant, pourtant facile à utiliser de MPLAB Integrated Development Environment (IDE).

Le MPLAB ICD 3 In-Circuit sonde débogueur est connecté à l'ordinateur de l'ingénieur de conception en utilisant une interface haut débit USB 2.0 et est relié à la cible avec un connecteur compatible avec le MPLAB ICD 2 ou systèmes REAL ICE MPLAB (RJ-11). MPLAB ICD 3 prend en charge toutes les têtes d'émulation.

MPLAB ICD 3 a amélioré la vitesse par rapport à MPLAB ICD 2

* + 1. Algorithmes :
       1. Diagramme général traceur GPS/GSM
       2. Réception du tram GPS et le découpage :
       3. Traitement de la position et l’envoi par SMS :



1. Perspective :

Ce qui est prévu maintenant est bien évidemment le développement principalement comme convenu dans notre calendrier prévisionnel, mais aussi de s'occuper de la finalisation.

Tout d'abord nous allons faire quelque test, resussir a afficher et controler l'afficheur tactileafin de comprendre la programmation sur le microcontrôleur et en parallèle préparer comme il faut notre environnement de développement pour ne pas avoir de soucis par la suite.

Ensuite nous allons développer tout en étant accompagné par notre tuteur qui nous recadrera s’il faut pour qu'on puisse gérer ce projet avec le peu de temps qu'il nous reste pour pouvoir aboutir a un traceur GPS/GSM qui fonctionnera.

Il faut notamment programmer les protocoles, mais aussi programmer pour les fonctionnalités GSM (ce que nous allons attaquer en premier ), et ensuite programmer les fonctionnalités GPS car nous aurons besoin d'une sonde de programmation et débugage qu'on n'a pas actuellement.

Pour finir, il faudra s'occuper de finaliser ce projet avec un rapport et une présentation et un code propre d'ici 2 mois qui fonctionnent, mais aussi faire le bilan pour nous même de tout ce qu'on apprit, car le but est de gagner de l'expérience sur plusieurs domaines lors de ce projet.

1. Conclusion :

Cette première partit du projet du cours «projet tutoré» nous a permis d'en apprendre énormément sur ce qu'est un traceur GPS/GSM, mais aussi sur la carte Mikromédia. Avant même d'avoir commencé le projet en lui même,la partie théorique avec notre tuteur Monsieur TOUATI fut très enrichissant notamment avec les conseils précieux qu'il nous a donnés pour la réalisation du projet traceur GPS/GSM.

Évoluer au sein d'un projet n'est pas qu'une histoire de travail personnel à rendre au professeur, mais il convient de comprendre quel est le but de ce projet et de travailler avec son binôme de la meilleure des manières, chacun en apportant ses compétences que ce soit dans le social ou le technique.

En effet nous avons pu voir que lorsqu’un projet est assez conséquent il ne suffit pas de coder pour le réaliser, mais toute un plan de travail doit être planifier tout en faisant attention constamment au délai, de plus nous devions tout d'abord nous renseigné énormément sur le sujet avant de pouvoir commencer quoi que se soit,ainsi des recherches ont était fais de manière générale, mais aussi dans un second temps des recherches plus orienté technique.

Nous avons rencontré quelque difficulté sur la recherche technique, car la programmation sur un microcontrôleur est une découverte pour nous.La plupart des taches ont était réaliser à temps vu qu'on s'est efforcé a respecté le calendrier prévisionnel et la plus grande des difficultés a était de gérer la période d'examen qui nous a stoppés sur nos démarches concernant ce projet.

Ainsi,nous avons appris énormément en cette première partit du projet et acquis une expérience supplémentaire qui nous a permis de mieux comprendre à quoi sert de planifier, de s'organiser et de respecter des délais, car nous devons maintenant attaquer le développement sur microcontrôleur qui sera un autre challenge très intéressant pour nous.