

Développement, création et adaptation d'un allumage et d'une injection électronique sur un moteur Honda GX35

ing. G. VAN LAREBEKE
ECAM – Bruxelles

Cet article traite du développement, de la création et de l'adaptation d'un allumage et d'une injection électronique pour un moteur Honda GX35 servant à la propulsion d'un véhicule spécialement conçu afin de diminuer au maximum la consommation.

Mots-clefs : allumage, injection électronique, moteur

This article explains the development, the creation and the adaptation of an electronic ignition and injection system for a Honda GX35 engine used to power a small car. The aim of that car is to reduce as much as possible the consumption.

Keywords : electronic injection, ignition, engine

1. Introduction

Depuis l'invention du moteur à explosion, de nombreuses améliorations ont été apportées.

Ces dernières décennies l'avènement de l'électronique a permis d'optimiser la gestion des moteurs essentiellement au niveau de l'injection de carburant et de l'allumage.

L'ECAM participe depuis de nombreuses années au Shell Eco Marathon. Il s'agit d'une compétition internationale ayant pour but la conception de véhicules consommant le moins possible de carburant. Il y a plusieurs années, l'idée avait été lancée de concevoir une gestion électronique du moteur.

L'ECAM ayant terminé la fabrication d'un nouveau prototype cette année, l'idée de faire fonctionner ce nouveau-né avec un moteur commandé électroniquement s'est imposée d'elle-même.

Comme il est certain qu'une année ne suffira pas pour obtenir les meilleurs résultats, ce travail a été fait non seulement dans le but de concevoir une électronique de gestion, mais aussi afin de servir de référence pour les élèves voulant se pencher sur la gestion électronique d'un moteur dans le futur.

2. Objectifs

Initialement le véhicule était propulsé par un moteur thermique essence à carburateur et allumage fixe. Malheureusement un tel moteur ne permet pas d'avoir un bon rendement pour une large plage de vitesses de rotation. Il a donc été décidé de lui adapter une gestion électronique afin de pouvoir optimiser le rendement pour toute la plage de fonctionnement du moteur.

La gestion électronique du moteur se décompose en deux parties : gestion de l'allumage et gestion de l'injection de carburant.

2.1 Allumage

L'explosion dans un moteur à combustion n'est pas instantanée. En effet, entre le moment où l'étincelle se fait et le moment où tout le mélange a brûlé, il s'écoule un certain temps. Ce laps de temps dépend de nombreux paramètres, ce qui ne simplifie pas la gestion de l'allumage.

Les principaux paramètres sont : la richesse du mélange, l'indice d'octane de l'essence, la température du moteur, la pression dans la chambre de combustion et l'homogénéité du mélange.

Une fois ce temps connu, il faut calculer l'avance à l'allumage à donner afin que le mélange explose au bon moment et ainsi augmenter au maximum l'aire du diagramme pV et donc le rendement (en considérant tous les autres paramètres fixes).

On remarque qu'il est très difficile de trouver une formule permettant de connaître avec précision l'avance à l'allumage à donner au moteur en fonction des différents paramètres précités.

Pour simplifier cela, on emploie une cartographie. Il s'agit d'une table de données reprenant la valeur recherchée : à savoir l'avance à l'allumage ou la quantité d'essence à injecter en fonction d'un ou deux paramètres (la vitesse de rotation du moteur et/ou la charge).

2.2 Injection

Afin d'obtenir une bonne combustion, les proportions des constituants du mélange doivent être les plus proches possibles de la stoechiométrie. Il est courant d'employer un débit-mètre afin de connaître la quantité d'air aspirée par le cylindre. Dans ce cas, il est très facile de déterminer la quantité d'essence à injecter puisqu'elle est proportionnelle à la quantité d'air aspirée.

Il est aussi possible de travailler avec une cartographie comme c'est le cas pour l'allumage.

3. Modifications à apporter au moteur

Afin de pouvoir adapter une gestion électronique sur le moteur, diverses modifications ont dû être apportées.

3.1 Volant magnétique

L'allumage étant à l'origine un simple allumage par volant magnétique, ce dernier a été modifié afin de supprimer le magnétisme et une roue d'inertie comportant une fente a été usinée et adaptée sur ce qui restait du volant d'origine.

Un capteur optique servant à détecter le passage de cette fente permet d'avoir une impulsion par tour moteur.

3.2 Allumage

La bobine d'allumage a tout simplement été enlevée et une bobine de voiture connectée à l'électronique de gestion servira à allumer la bougie.

3.3 Capteurs

Capteur vilebrequin

Comme expliqué plus haut, le capteur vilebrequin est un capteur optique détectant le passage d'une fente. Il permet de détecter précisément un point sur les 360° d'un tour moteur.

Capteur arbre à came

Un moteur quatre temps faisant 2 tours de vilebrequin par cycle moteur, il a fallu trouver un moyen de pouvoir différencier le point mort haut de début d'admission du point mort haut de début de détente. Le seul endroit où cette information est disponible est sur l'arbre à came.

La poulie de came a été usinée afin d'y placer un petit aimant qui sera détecté.

Le bâti moteur a aussi été usiné afin de pouvoir recevoir le capteur à effet Hall.

Tubulure d'admission

Cet élément est la pièce maîtresse du système d'injection. En effet, elle permet d'adapter un injecteur sur un moteur initialement prévu pour fonctionner avec un carburateur.

Ce n'est cependant pas à proprement parler une pièce mécanique en mouvement comme le seraient un piston ou une bielle. Le porte injecteur sert simplement de pièce de jonction entre l'injecteur et le moteur ainsi que de support pour le filtre à air et il contient le papillon des gaz.

Pour la conception de cette pièce il a fallu faire des recherches sur le fonctionnement d'un système d'injection et les paramètres importants dont il faut tenir compte.

Le premier d'entre eux est le point d'injection afin d'avoir un mélange air-essence le plus homogène possible. La littérature technique nous informe que le meilleur endroit pour l'injection est sur la soupape d'admission (le

cône d'injection doit être juste sur la soupape ou en déborder légèrement). En effet, cette dernière est chaude et va donc favoriser l'évaporation des micro-gouttellettes d'essence. De plus, c'est à cet endroit que les turbulences sont les plus importantes, ce qui favorisera le mélange de l'air avec l'essence vaporisée afin qu'il soit le plus homogène possible.

Une fois les paramètres connus, il a fallu concevoir la pièce à proprement parler. C'est là que les plans cordialement envoyés par Mr. Maarten ont été d'une très grande aide.

Sur la plupart des moteurs, l'injecteur a sa place prévue dans la culasse. Malheureusement dans notre cas, il était impossible d'usiner cette dernière afin de placer le mieux possible l'injecteur, la disposition des soupapes et de l'arbre à came nous en empêchant.

Il a donc fallu concevoir une pièce permettant de placer l'injecteur le mieux possible compte tenu des paramètres précités.

Les plans reçus ont été imprimés en agrandi afin de pouvoir tracer l'angle optimal pour positionner l'injecteur et permettre de dessiner toute la pièce. Afin de limiter les pertes de charge, la tubulure d'admission de la culasse a été élargie, passant de 10 mm de diamètre à 12 mm. Les angles du porte-injecteur ont été choisis afin de limiter au maximum les pertes de charge tout en restant dans des dimensions acceptables.

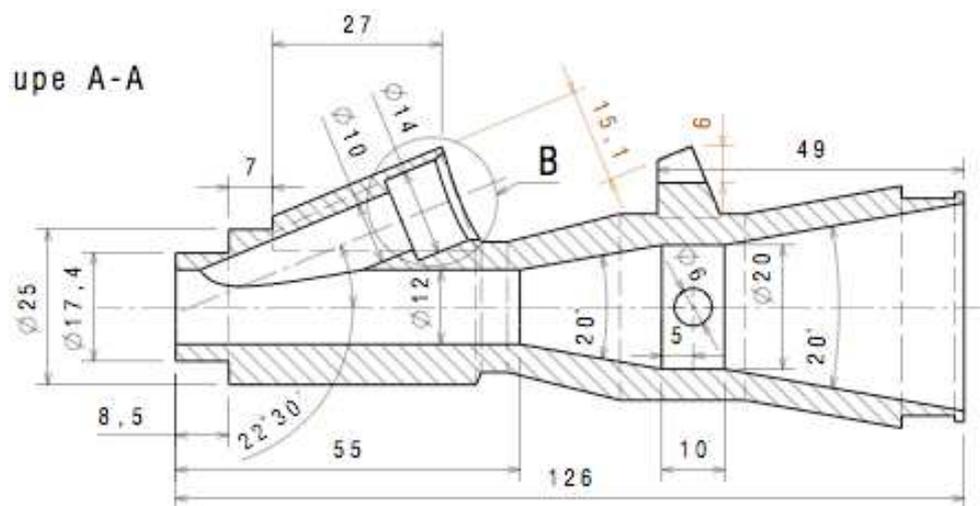


Figure 1 : Tubulure d'admission

Mise sous pression du réservoir

Pour faire fonctionner l'injecteur, il faut l'alimenter en essence sous pression et le règlement du Shell Eco Marathon nous autorise à embarquer deux litres d'air à cinq bars.

Cependant il y a des restrictions : la pression d'arrivée au niveau de l'injecteur ne peut pas varier et il doit être possible de vérifier que la pression dans la bouteille n'est pas supérieure à cinq bars.

Une bouteille de coca (réputée pour sa solidité) de deux litres a été employée comme réservoir d'air sous pression. Un premier "T" a été placé après la bouteille pour y adapter une valve de vélo afin de pouvoir mettre la bouteille sous pression. Ensuite, un premier manomètre destiné à vérifier que la pression n'excède pas cinq bars a été placé, suivi d'un détendeur réglable pour pouvoir adapter la pression de l'air envoyée au réservoir.

Du réservoir, un simple flexible est connecté à l'injecteur.

Il a fallu usiner certaines pièces nécessaires à l'interconnexion des différents composants tel que l'embout pour la bouteille de 2 litres, le bouchon du réservoir d'essence et la connexion à l'injecteur.

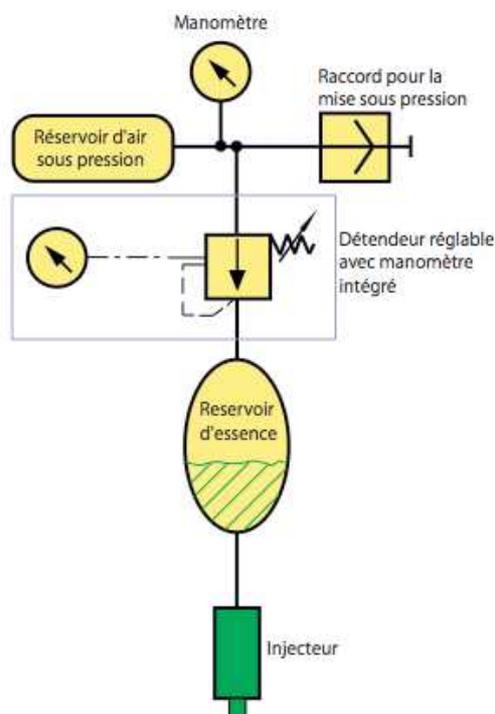


Figure 2 : Schéma pneumatique du circuit de mise sous pression du réservoir

4. Fonctionnement théorique initialement envisagé

4.1 Prélude

Afin de comprendre la suite de cet article, il est important de donner quelques informations basiques sur le fonctionnement d'un microcontrôleur ainsi que des explications plus détaillées sur certaines de ses caractéristiques.

Un microcontrôleur est une unité de traitement d'information de type microprocesseur à laquelle ont été ajoutés des périphériques internes tels que de la RAM, de la Flash, des convertisseurs analogiques digitaux, des timers et bien d'autres choses encore. Cela permet de réaliser facilement des montages sans nécessiter l'ajout de composants annexes.

Une chose très importante à savoir à propos du type de microcontrôleur employé pour ce projet est qu'il n'est pas multi-tâche. Il ne peut donc effectuer qu'une seule opération à la fois et cela conditionnera la stratégie de programmation.

Deux termes propres à la programmation de microcontrôleurs seront employés tout au long de cet article. Il s'agit d' *Interrupt* et *Timer*.

Un interrupt est un évènement externe ou interne au microcontrôleur qui arrête la tâche en cours d'exécution et oblige ce dernier à exécuter un code propre à cet évènement. Une fois le code exécuté, il reprend la tâche en cours d'exécution là où il l'avait arrêté, lorsque l'interrupt est survenu.

Les interrupts sont couramment employés pour que le microcontrôleur réagisse le plus rapidement possible à certains évènements.

Un timer, quant à lui, est un périphérique matériel intégré au microcontrôleur permettant de mesurer des durées. Ses caractéristiques sont la vitesse à laquelle il s'incrémente et sa capacité.

Elles seront choisies lors de la configuration du timer.

En ce qui concerne la vitesse d'incrémentation, plusieurs possibilités existent en fonction du type de timer utilisé. Seule la technique employée ici sera expliquée. Le timer est synchronisé avec l'horloge interne du microcontrôleur après avoir été divisée par un prescaler. Cela signifie que si le microcontrôleur fonctionne à 40MIPS (millions d'instructions par seconde) et que le prescaler du timer est de 1, ce dernier sera incrémenté tous les 1/40000000 secondes. Avec un "prescaler" de 8, il serait incrémenté tous les 1/5000000 secondes.

La capacité du timer quant à elle est, soit de 16 bits, soit de 32 en accouplant 2 timers 16 bits ensemble. Une particularité du timer est qu'il est capable de générer un interrupt lorsqu'il y a overflow.

4.2 Position des capteurs

Comme expliqué plus haut la position du capteur de phase ne peut être modifiée. Le point mort haut avant admission sera toujours dans la fenêtre de détection du capteur à effet Hall. Le seul capteur dont on peut régler le moment auquel il enverra un signal à la carte de gestion est le capteur vilebrequin. Ce dernier sera positionné de telle manière que l'impulsion se fasse 90° avant le point mort haut fin d'échappement, point qui sera considéré comme le 0° d'un cycle moteur.

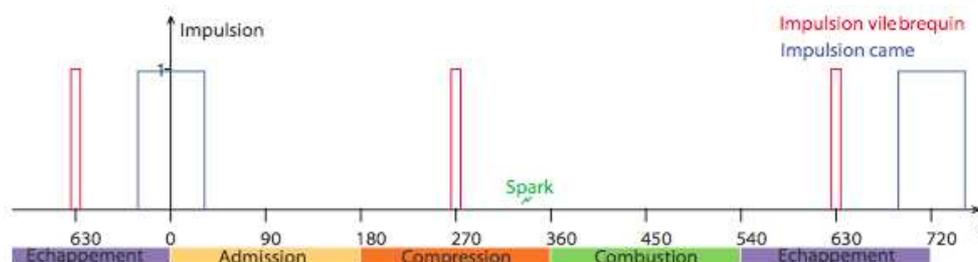


Figure 3 : Impulsions reçues par le microcontrôleur lors d'un cycle moteur

4.3 Hypothèses

Le style de conduite au Shell Eco Marathon est assez particulier. Afin de consommer le moins possible le moteur n'est employé que pour accélérer le véhicule jusqu'à atteindre environ 40 km/h.

Ensuite le moteur est coupé et le véhicule continue sur sa lancée en décélérant. Une fois la vitesse du véhicule passée en dessous d'approximativement 20 km/h, le moteur est redémarré et ainsi de suite.

Ce dernier n'est donc utilisé que pendant de courts moments d'accélération et le moment où le rendement est le meilleur est lorsque le papillon des gaz est grand ouvert car les pertes par aspiration sont les plus faibles. Le papillon n'aura donc que deux positions : fermé pour faire tourner le moteur au ralenti afin de le chauffer et grand ouvert lors des phases d'accélération.

L'accélération des véhicules étant très faible, la vitesse de rotation entre deux cycles moteur sera considérée comme étant identique. La vitesse de rotation instantanée du moteur durant un cycle complet sera elle aussi considérée comme constante.

4.4 Fonctionnement

Afin de simplifier l'explication générale, on considérera que les *timer_allumage*, *timer_deb_inj* et *timer_injection* sont des timers qui décomptent. Ce n'est bien sûr pas le cas, mais cela permet de comprendre plus facilement. Par la suite le fonctionnement réel sera détaillé. Une variable booléenne *allu* sera employée. Si elle est à 1 cela signifie que lors du tour à venir il faudra allumer la bougie.

Considérons le moteur tournant à vitesse constante, au temps t_0 le moteur se trouve à 600° et en phase d'échappement. Au temps t_1 il est à 630° et le capteur vilebrequin envoie une impulsion au microcontrôleur, ce dernier démarre un timer que l'on appellera *timer_compte_tour*.

Il regarde ensuite si *allu* est à 1. Comme ce n'est pas le cas il ne fait rien d'autre.

Au temps t_2 le moteur est à 0° , le microcontrôleur reçoit une impulsion du capteur d'arbre à came. Suite à cette impulsion il met la variable *allu* à 1. Au temps t_3 le moteur est à 270° et le microcontrôleur reçoit une seconde impulsion du capteur vilebrequin, il lit et enregistre la valeur du *timer_compte_tour* avant de le remettre à zéro et de le redémarrer. Ensuite il regarde si la variable *allu* est à 1, cela étant vrai il calcule, à partir du nombre d'impulsions mis par le *timer_compte_tour* pour faire un tour, la vitesse de rotation du moteur et remet ensuite la variable *allu* à 0.

En fonction de cette vitesse, le microcontrôleur va chercher dans une table l'avance à l'allumage des 2 vitesses de rotation les plus proches de celle mesurée et fait une interpolation pour trouver la valeur exacte. La même technique est employée pour connaître le temps d'injection en fonction de la vitesse de rotation. La valeur du temps avant l'allumage de la bougie est mise dans le *timer_allumage* et une fois ce dernier arrivé à 0, une étincelle est produite.

Au temps t_4 le moteur est de nouveau à 630° , la seule différence est que la variable *allu* est maintenant à 0. Le microcontrôleur sait donc qu'il va devoir injecter au tour suivant. En fonction du moment d'injection voulu, le microcontrôleur calcule le temps à mettre dans le *timer_deb_inj*.

Lorsque ce timer arrive à 0, la valeur du temps d'injection est calculée en fonction de la vitesse de rotation du moteur et est mise dans le *timer_injection*. L'injecteur est ensuite allumé.

Une fois le *timer_injection* arrivé à 0, l'injecteur est coupé.

Et le cycle recommence.

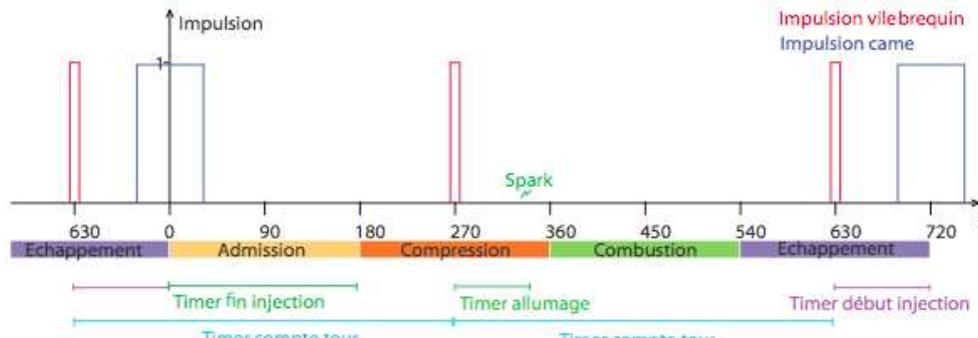


Figure 4 : Impulsions reçues par le microcontrôleur lors d'un cycle moteur ainsi que l'utilisation des différents timers

4.5 Choix du microcontrôleur

Le choix du microcontrôleur s'est basé sur plusieurs paramètres. Le premier étant la vitesse d'exécution, le deuxième le nombre de timers disponibles. Le choix s'est porté sur un dspic33fj256gp710 dont vous trouverez les caractéristiques ci-dessous.

Parameter Name	Value
Architecture	16-bit
CPU Speed (MIPS)	40
Memory Type	Flash
Program Memory (KB)	256
RAM Bytes	32,768
Temperature Range C	-40 to 85
Operating Voltage Range (V)	3 to 3.6
I/O Pins	85
Pin Count	100
System Management Features	PBOR
Internal Oscillator	7.37 MHz, 512 kHz
nanoWatt Features	Fast Wake/Fast Control
Digital Communication Peripherals	2-UART, 2-SPI, 2-I2C
Analog Peripherals	2-A/D 32x12-bit @ 500(kcps)
CAN (#, type)	2 ECAN
Capture/Compare/PWM Peripherals	8/8
16-bit PWM resolutions	16
Timers	9 x 16-bit 4 x 32-bit
Parallel Port	GPIO
Hardware RTCC	No
DMA	8

Figure 5 : Caractéristiques du microcontrôleur employé

5. Simulateur moteur

Dans le domaine de l'électronique et de la programmation il est rare de réussir à faire fonctionner un circuit et un programme du premier coup. Le code est souvent écrit par parties successives et testé au fur et à mesure pour vérifier qu'il répond correctement aux attentes.

Une carte de développement a été achetée et un simulateur de moteur créé afin de pouvoir vérifier le bon fonctionnement du programme. La carte de développement est simplement une carte électronique avec un microcontrôleur installé dessus et déjà quelques composants et interfaces. De plus, il est possible de lui ajouter une carte fille sur laquelle toutes les connexions du micro se retrouvent et ainsi il est possible de lui brancher diverses entrées et sorties. Il va sans dire que le microcontrôleur installé sur cette carte est celui qui a été choisi plus haut.

Une fois le circuit et le programme fonctionnant correctement, une carte complète sera développée afin d'être embarquée sur le véhicule.

Le simulateur moteur, quant à lui, est simplement un montage qui permet de simuler les impulsions envoyées par le moteur au microcontrôleur et aussi de visualiser les moments de début d'injection, de fin d'injection et d'allumage.

Il est composé d'un disque gradué représentant les 720° d'un cycle 4 temps, d'un capteur optique pour l'impulsion vilebrequin, d'un capteur à effet Hall pour l'impulsion de came, de trois potentiomètres pour pouvoir choisir l'avance à l'allumage, le moment de début d'injection et le temps d'injection.

Un petit écran LCD permet d'afficher diverses valeurs telles que la vitesse de rotation, l'avance à l'allumage, etc...

Deux LEDs serviront, grâce à l'effet stroboscopique, à visualiser physiquement le moment de début d'injection, de fin d'injection et l'avance à l'allumage sur le disque. Ce dernier est fendu en deux extrémités pour simuler les impulsions de vilebrequin et un aimant est placé de manière à simuler l'impulsion arbre à came.

Un deuxième aimant a été placé en vis-à-vis pour une question de balourd mais le champ est positionné de telle manière qu'il n'est pas détecté par le capteur à effet Hall.

Les deux LEDs sont placées de part et d'autre du disque, l'une sert à afficher l'avance à l'allumage et l'autre peut, soit afficher le début d'injection, soit la fin d'injection. Le disque est entraîné par un petit moteur DC capable de tourner à 7000 tr/min ce qui correspond à une vitesse de rotation du moteur thermique de 14000 tr/min.

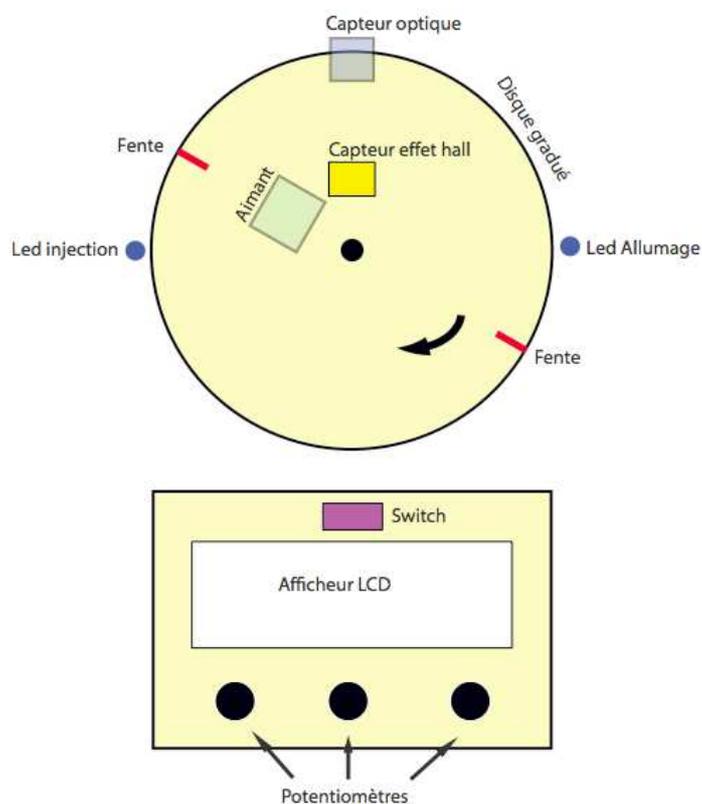


Figure 6 : Schéma du simulateur moteur

6. Programmation

6.1 Allumer les led au moment voulu en fonction des valeurs lues par les potentiomètres.

Déclaration et initialisation : Cette partie du code contient les déclarations des différentes variables qui serviront dans le reste du programme et des différentes bibliothèques, utiles afin de pouvoir appeler des fonctions prédéfinies.

1ère partie : Fonction lisant les potentiomètres. La fonction `get_pot` permet la lecture des potentiomètres dont les valeurs sont codées en 12 bits et elle transforme ces valeurs représentant l'angle d'avance à l'allumage, le temps d'injection et l'angle de début d'injection en leurs valeurs respectives numériques plus facilement utilisables.

2ème partie : Calcul de la vitesse de rotation du moteur.

Comme l'impulsion du vilebrequin est la seule impulsion donnant une information précise sur la position du moteur et que tous les temps nécessaires à calculer les positions d'avance à l'allumage et de début d'injection dépendent de lui, il faut que le microcontrôleur réagisse directement lorsqu'il reçoit une impulsion venant de sa part. Pour réaliser ceci, un interrupt_vilebrequin arrête les tâches en cours tel que l'affichage de valeurs sur l'écran LCD, et exécute le code lié à cette interruption.

En premier lieu, il faut lire la valeur du timer_compte_tour, la stocker et réinitialiser ce dernier.

3ème partie : Usage de la valeur du timer_compte_tour pour allumer les LEDs au moment voulu.

La variable *allu* est ensuite employée pour savoir si l'on est en admission ou en détente. Si le signal est présent, un code sera exécuté ; sinon, un code différent sera exécuté.

Allu = 1 : dans ce cas, le moteur est 90° avant PMH explosion, par conséquent la bougie devra enflammer le mélange dans 90° diminué de l'angle d'avance à l'allumage. Comme la valeur du timer_compte_tour est connue, il suffit de la diviser par 360 et de la multiplier également par la différence entre 90 et la valeur d'angle d'avance à l'allumage lue sur le potentiomètre.

Ce résultat est la valeur qu'un timer fonctionnant à la même vitesse que timer_compte_tour, mettrait pour que le moteur atteigne l'angle d'avance à l'allumage désiré.

Cependant ceci est vrai uniquement si ce timer est démarré exactement au moment où le timer_compte_tour est lu. Or ceci n'est pas le cas puisque le microcontrôleur a dû faire des opérations entre-temps.

C'est pour cette raison que l'une des caractéristiques ayant mené au choix du microcontrôleur est la vitesse à laquelle il est capable d'effectuer des instructions.

Il faut maintenant trouver un moyen d'allumer la petite LED une fraction de seconde au bon moment afin de lire sur le disque en rotation si l'angle affiché est bien celui voulu. On ne peut pas demander au microcontrôleur de compter ce temps car tout le temps de calcul serait utilisé au décompte.

Un autre timer_allumage fonctionnant à la même vitesse que timer_compte_tour sera donc utilisé. Il faut cependant qu'il allume pendant une fraction de seconde la LED_allumage après avoir compté jusqu'à la valeur calculée ci-dessus.

Un deuxième `interrupt_allu` sera donc utilisé, cependant, comme cela a été expliqué dans le prélude, il n'est pas possible de choisir la valeur du timer à laquelle se fera cette interruption.

Le seul moment où un timer sait envoyer une interruption est lorsqu'il dépasse la valeur maximale qu'il est capable d'enregistrer. La valeur qui sera rentrée dans le `timer_allumage` sera donc la valeur maximale qu'il peut contenir diminuée de la valeur calculée plus haut.

Comme il a été décidé de pouvoir choisir la position de début d'injection, un troisième `timer_deb_inj` sera employé de la même façon que le `timer_allumage` ainsi qu'un `interrupt_deb_inj`.

La valeur avant le début d'injection est simplement la valeur du `timer_compte_tour` divisée par 360 et multipliée par la somme de la valeur de l'angle de début d'injection lue sur le potentiomètre et de 90° .

4ème partie : Traitement des interruptions.

Il faut maintenant traiter les interruptions créées par l'overflow de `timer_allumage` et `timer_deb_inj`.

L'`interrupt_allumage` est simplement chargé d'allumer la `LED_allumage` pendant une fraction de seconde.

En ce qui concerne l'`interrupt_deb_inj`, le travail à effectuer est un peu plus long. La première chose à faire est de démarrer un quatrième `timer_fin_inj` qui enverra un `interrupt_fin_inj` lorsqu'il y aura overflow. La valeur mise dans ce timer est la valeur maximale qu'il peut contenir diminué du temps d'injection calculé en fonction de la vitesse d'incrémentation du timer.

Il faut ensuite regarder dans quelle position est le switch permettant de choisir si l'on veut afficher l'angle de début d'injection ou pas. S'il est en position permettant d'afficher l'angle de début d'injection, il allume la `LED_inj` une fraction de seconde ; si non, il ne fait rien.

Maintenant une quatrième interruption doit être traitée. Cette opération n'est pas très compliquée car il suffit de lire la valeur du switch choisissant l'angle que l'on veut visualiser, et donc de choisir d'allumer la `LED_inj` une fraction de seconde ou pas.

5ème partie : Code principal (main)

Pour terminer, il faut écrire le code principal, celui qui sera exécuté au démarrage. Ce dernier permet de configurer les différents timers ainsi que le port analogique servant aux potentiomètres.

Il initialise aussi l'afficheur LCD, autorise les différentes interruptions et affiche sur l'écran LCD différentes valeurs.

Une fois le code écrit plusieurs petits soucis sont apparus, il s'agit principalement de petites choses sans gravité et dont la cause était souvent très simple à défaut d'être rapidement comprise.

6.2 Utilisation d'une cartographie afin de faire varier l'avance à l'allumage et le temps d'injection en fonction de la vitesse de rotation du moteur

Dans un premier temps il faut créer les deux vecteurs de la cartographie. Un pour l'avance à l'allumage et l'autre pour le temps d'injection. Comme il n'est pas possible de faire une cartographie pour toutes les valeurs de vitesse de rotation, un pas de 250 tr/min a été choisi car il permet d'illustrer l'utilisation de cartographies différentes tout en conservant des propriétés d'efficacité suffisantes dans le cadre de ce travail.

Une interpolation linéaire sera employée afin de trouver les valeurs intermédiaires. La valeur de l'indice 0 du vecteur sera celle pour une vitesse de rotation de 0 tr/min. Cela peut paraître bizarre puisque le moteur ne tournera jamais à cette vitesse. La raison de ce choix est très simple : grâce à cette mise en forme, la partie entière de la division de la vitesse de rotation par le pas donne l'indice de la valeur basse à aller chercher dans le vecteur. Il suffira ensuite de prendre en compte le reste de la division afin de faire une règle de trois.

Cependant cette technique n'a pas fonctionné. En effet la cause de cette situation n'est pas déterminée avec certitude, mais elle était probablement due à l'utilisation d'une valeur décimale puisqu'elle a été résolue par l'utilisation de variables à virgule fixe. Cela consiste à réaliser tous les calculs en entier mais en prenant l'unité comme étant le dixième ou le 100ème.

De cette manière, on peut par simple division au final traiter des nombres décimaux.

6.3 Allumer la bougie et l'injecteur

Pour l'injection, il suffit lors de l'interrupt_deb_inj de mettre la sortie de commande injecteur à 1 et de la remettre à 0 lors de l'interrupt_fin_inj.

En ce qui concerne l'allumage, le travail est un peu différent. En effet, l'étincelle ne se produit pas lorsque la bobine est mise sous tension, mais bien lorsque l'alimentation est coupée. Cela est dû au fonctionnement de la bobine. Dans un premier temps, elle est mise sous tension et un fort courant

passé dans l'enroulement primaire. Lorsque le courant est interrompu, la variation du flux crée une force électromotrice servant à l'allumage de la bougie. Il faut donc mettre sous tension la bobine un certain temps avant sa coupure. Dans notre cas, cela sera fait au moment de l'interrupt_came.

Il suffit ensuite de couper l'alimentation de la bobine lors de l'interrupt_allumage afin d'obtenir une étincelle.

7. Développement de la carte microcontrôleur

La carte électronique et le programme fonctionnant correctement, il faut maintenant créer une carte complète dédiée à la gestion de l'injection et de l'allumage électronique du moteur.

Afin que cette carte soit évolutive il a été décidé de la rendre la plus polyvalente possible. En effet, il serait intéressant de pouvoir améliorer cette carte par l'ajout de capteurs, de périphériques, etc...

7.1 Alimentation de la carte

La seule source électrique sur le véhicule est une batterie de 12volts. Un bornier sera placé sur la carte afin d'y connecter la batterie. Le microcontrôleur fonctionnant en 3,3 V , un régulateur de tension sera employé. Il a comme but de réguler les 12 V en 3,3 V . L'écran LCD demandant du 5 V , un deuxième régulateur sera utilisé afin d'avoir du 5 V .

7.2 Le microcontrôleur et ses périphériques

Afin de fonctionner le microcontrôleur a besoin d'être alimenté en 3,3 V et d'un quartz pour avoir une fréquence sur laquelle se baser. Un port servant à la programmation du microcontrôleur sera aussi ajouté. Comme il existe plusieurs formats au microcontrôleur, deux des trois formats possibles seront placés sur la carte. Il s'agit du format PIM (Plug In Module) qui permet d'enlever et de remplacer facilement le microcontrôleur et du format TQFP (Thin Quad Flat Pack) qui est un standard. Dans le cas du format PIM, les capacités de filtrage sont intégrées. Cela n'étant pas le cas pour le format TQFP elles seront ajoutées à la carte.

7.3 Connectique injecteurs

Etant donné le travail accompli pour réaliser cette carte il serait intéressant de pouvoir la réemployer pour d'autres types de moteurs. Quatre connexions injecteurs seront donc installées.

Cela permettra de pouvoir employer cette carte sur des moteurs ayant jusqu'à 4 cylindres. Un injecteur standard fonctionne en 12 V et demande environ 1 A. Le microcontrôleur n'étant capable de fournir que 4 mA sous 3,3 V un transistor de puissance sera employé pour chaque injecteur.

Il s'agit d'un transistor de type Darlington capable de délivrer 10A et intégrant une diode de roue libre nécessaire dans le cas de charge inductive. Il sera connecté au microcontrôleur via une résistance limitant le courant.

7.4 Connectique bobine d'allumage

Bien que fonctionnant sans problème sur la carte de développement, la partie puissance servant à commander la bobine d'allumage n'a eu de cesse de tomber en panne. Le transistor de puissance rendait l'âme presque instantanément alors qu'il ne chauffait même pas. Afin de contourner le problème, un allumage à transistor a été acheté et connecté à la carte via une des sorties de puissance.

7.5 Connectiques capteurs vilebrequin et came

Les capteurs fonctionnant aussi en 5 V c'est cette tension qui sera utilisée pour leur alimentation. La sortie du capteur sera quant à elle connectée directement sur une entrée interruption du microcontrôleur.

7.6 Divers

Comme précisé plus haut le but est de créer une carte la plus évolutive possible. Divers autres entrées/sorties ont donc été ajoutées même si ces dernières ne seront pas utilisées tout de suite. Voici ces différentes entrées/sorties. Afficheur LCD, compteur de vitesse, horloge, port RS232, port CAN, 6 entrées analogiques, 6 entrées/sorties digitales, 6 sorties de 500mA, une sonde lambda, un port I2C et un port SPI.

8. Essai sur le banc et création de la cartographie

Le but de cette partie est de créer une cartographie du moteur permettant d'avoir le rendement maximum pour toute vitesse de rotation. Pour ce faire, un banc d'essais a été employé.

8.1 Explication du fonctionnement du banc et mode opératoire.

Le banc d'essais est constitué d'un moteur asynchrone piloté par un automate relié via une courroie au moteur thermique. Grâce à l'électronique de contrôle il est possible d'imposer une certaine vitesse de rotation au moteur électrique et de mesurer le couple nécessaire au maintien de cette vitesse de rotation. Couple qui peut être positif dans le cas où le moteur électrique entraîne le moteur thermique et négatif dans le cas contraire.

Du côté électrique : nous avons une puissance calculée par le produit de la vitesse de rotation et du couple.

Du côté thermique : une puissance sous la forme d'un débit d'essence dont le PCI (Pouvoir Calorifique Inférieur) est connu.

Il est donc tout à fait possible de connaître le rendement du moteur thermique et d'essayer d'améliorer ce dernier en modifiant la richesse du mélange ainsi que l'avance à l'allumage. Cependant, l'accouplement entre les deux moteurs étant réalisé à l'aide de deux poulies de tailles très différentes et d'une courroie trapézoïdale, une partie de la puissance mécanique est dissipée sous forme de frottements et ne sera donc pas mesurée par le banc d'essais. Il faudrait afin d'avoir une mesure correcte connaître le rendement de la transmission.

Cela n'est cependant pas nécessaire dans notre cas, puisque ce qui nous intéresse est de maximiser le rendement pas de le connaître. Or une image du rendement est simplement la puissance mesurée par le banc divisée par la quantité d'essence injectée. Et cette dernière est proportionnelle au temps d'injection moins le temps de latence de l'injecteur (à savoir 1 ms dans le cas de l'injecteur utilisé). Il serait donc facile de demander à l'automate qui gère le moteur électrique d'envoyer la puissance mesurée au microcontrôleur via une connexion série. Ce dernier diviserait cette puissance par le temps d'injection moins 1 ms et l'afficherait sur l'écran LCD. Il suffirait ensuite de faire varier les trois paramètres (à savoir le temps d'injection, le moment d'injection et l'avance à l'allumage) de manière à maximiser le rapport puissance mesurée sur temps d'injection. Le

problème est que la liaison série de l'automate est compliquée et le temps a manqué pour la mettre au point.

Une autre stratégie a donc été utilisée. Dans un premier temps on recherchera les valeurs de temps d'injection et d'avance à l'allumage maximisant la puissance. Pour ce faire, le moteur thermique est entraîné à la vitesse de 1000 tr/min et via les potentiomètres de réglage, la puissance maximum est trouvée par essais successifs. On commencera par régler le moment d'injection pour que celui-ci ait lieu durant la phase d'admission. L'avance à l'allumage sera quant à lui fixé à 0° et le premier paramètre que l'on fera varier est le temps d'injection. En augmentant progressivement ce temps, on remarque que la puissance mesurée passe par un maximum. Une fois ce maximum trouvé, on fait varier l'avance à l'allumage en tentant d'améliorer encore la puissance. Ensuite, le temps d'injection est de nouveau modifié afin d'optimiser la quantité d'essence pour l'avance à l'allumage trouvé. Il est donc possible par itérations successives de trouver l'optimum de puissance pour un régime donné. La variation de position d'injection n'a montré aucune modification mesurable de la puissance développée. Il va sans dire que cette procédure est lente.

Les valeurs d'avance à l'allumage et de temps d'injections pour ce régime sont enregistrées dans un tableur. La même procédure est ensuite employée par pas de 250 tr/min.

9. Essais sur circuit

Depuis la création de cette électronique le véhicule a participé à 3 compétitions. La première fit surtout remarquer le manque de fiabilité du système. Plusieurs améliorations aussi bien au niveau électronique qu'au niveau de la programmation ont été effectuées et lors de la deuxième compétition le véhicule n'a souffert d'aucune panne et la distance parcourue à l'aide de 12 ml d'essence a été de 3300 mètres. Pour la 3^{ème} compétition une nouvelle cartographie a été faite à l'aide d'une sonde lambda. Cela a permis d'augmenter de 1000 mètres la distance parcourue avec les 12 ml d'essence.

10. Perspectives

Maintenant que l'électronique ne pose plus de soucis et que tous les petits problèmes ont été résolus, une des améliorations majeures à apporter serait de faire une rétroaction entre la sonde lambda et le temps d'injection. Cela permettrait de tenir compte des conditions climatiques ainsi que de la température du moteur. Il serait aussi intéressant de programmer le port CAN afin de pouvoir envoyer diverses données à un data-logueur. Ces données pourront ensuite être analysées afin d'optimiser la consommation. Les possibilités d'améliorations sont nombreuses, et la plupart sont souvent réalisables au niveau de la programmation.

11. Références bibliographiques

MYKE PREDKO. *Programming and customizing the PIC microcontroller*. 3rd ed. McGraw Hill

HÜTTE. *Manuel de l'Ingénieur*. Paris et Liège. Librairie Polytechnique. Ch. Béranger, Editeur.

Techniques de l'ingénieur. <http://www.techniques-ingenieur.fr>

LUCIO DI JASIO. *Programming 16-bit Microcontrollers in C. Learning to Fly the PIC 24*. Newnes.

NIGEL GARDNER. *PIC C. An introduction to programming the Microchip PIC in C*. Bluebird Electronics.

A. GRAHAM BELL. *Four-Stroke Performance Tuning*. Haynes Publishing.

DAN O'SULLIVAN AND TOM IGOE. *Physical Computing : Sensing and Controlling the Physical World with Computers*. Thomson Course Technology