

INTRODUCTION - DEFINITION DU PROJET

Depuis 3 ou 4 ans ma V85 est équipée d'un moteur réalisé par UsiRacing.

Il s'agit d'une réplique du 810-30 ayant équipé mon auto à sa sortie d'usine, alors qu'à l'achat en 2001 il y avait eu un échange standard qui s'est avéré être un 810-05 (R12TS identifié par le marquage sur son AAC).

Je l'ai alimenté avec un carbu de R5Alpine (32DIR58).

Ceci m'a entraîné dans 2 domaines hors de mes compétences : les réglages de carburation et l'enregistrement de données.

Réglages du carburateur : il ne s'agit pas de régler simplement le régime et la richesse au ralenti mais d'ajuster les gicleurs d'air, d'essence et autres tubes d'émulsion pour obtenir le meilleur d'un moteur et d'un carbu qui n'ont pas été conçus l'un pour l'autre.

Sur le papier c'est *facile* : il "suffit" de connaître la richesse pour chaque palier de vitesse de rotation et d'ouverture de papillon et ensuite de corriger en fonction de ce qu'on a observé par rapport à un objectif.

J'ai donc monté une sonde lambda sur l'échappement.

Oui; mais bon....

... lire à la volée l'AFR, les tr/mn et estimer au doigt mouillé l'enfoncement de la pédale d'accélérateur tout en roulant sur route ouverte et en notant sur un carnet les chiffres lus, même à deux ce n'est pas possible.

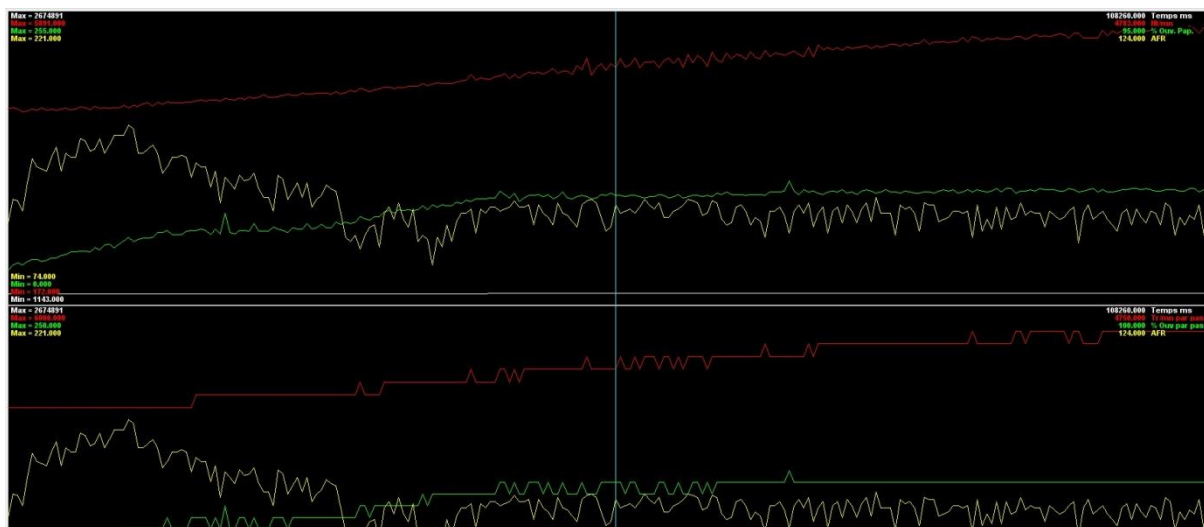
Donc il faut passer par un enregistreur de données qui peut s'acheter mais c'est tellement plus instructif d'essayer de le fabriquer... d'autant plus qu'il me fallait bien fabriquer quelque chose qui envoie un signal électrique proportionnel à l'enfoncement de l'accélérateur.

Là aussi c'est PhL qui m'a mis sur la voie grâce à ses enregistreurs basés sur des Arduino.

Il en a décrit un pour carbu (mais il n'enregistre pas l'ouverture du papillon des gaz) et un pour l'injection qui est beaucoup trop complet et complexe pour mon propre projet.

J'espérai pouvoir réunir les 2 pour en faire un qui réponde à mes besoins, c'est ce que mes petites mains ont réalisé et que je vais décrire dans les messages qui suivent cette longue introduction.

Pour les impatients voici directement le résultat :



Graphique séparé en 2 parties :

En haut, les valeurs brutes d'enregistrement : en rouge les tr/mn, en vert le % d'ouverture du papillon, en jaune (apparaît en blanc ici) l'AFR.

En bas les mêmes valeurs avec les mêmes couleurs, les régimes et % ouverture sont approximés par palier pour faciliter le dépouillement et l'inscription des AFR dans des cases.

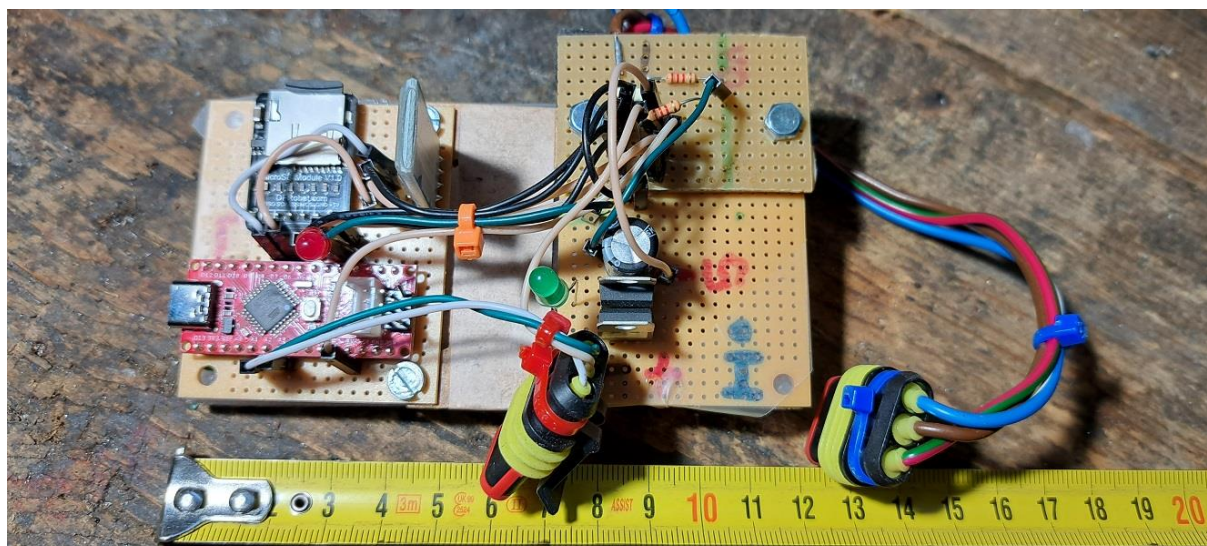
Dans la baie moteur ça ressemble à ça :



Le boîtier ouvert dans l'environnement :



Le montage sorti de l'auto :



MATERIEL

Le montage comporte un micro-processeur de la famille des Arduino, j'ai utilisé un « Seeduino Nano ».

Il est alimenté en +5V et une autre cosse est reliée à la masse.

Deux cosses sont reliées à un module Bluetooth. Cela permet de communiquer sans-fil avec un vieux smartphone (sans carte SIM). Le smartphone sert à commander de l'habitacle l'Arduino (démarrage/arrêt d'un enregistrement) et à afficher les mesures relevées dans le compartiment moteur.

Quatre autres cosses sont reliées à un module carte micro-SD. Sur cette carte on va enregistrer les valeurs mesurées à une cadence assez rapide (à chaque tour de vilebrequin).

Sur ma balade type d'environ 45 minutes il y a en moyenne un enregistrement toutes les 45 millisecondes soit près de 60.000 enregistrements. Heureusement chaque enregistrement n'occupe que peu de place sur la carte : les 45 minutes occupent seulement 1700ko d'une carte qui en contient 16Go !

Une cosse supplémentaire de l'Arduino allume (et éteint) une diode rouge signalant que l'enregistrement est en cours.

Enfin (et surtout) 3 cosses de l'Arduino reçoivent les mesures qui nous intéressent :

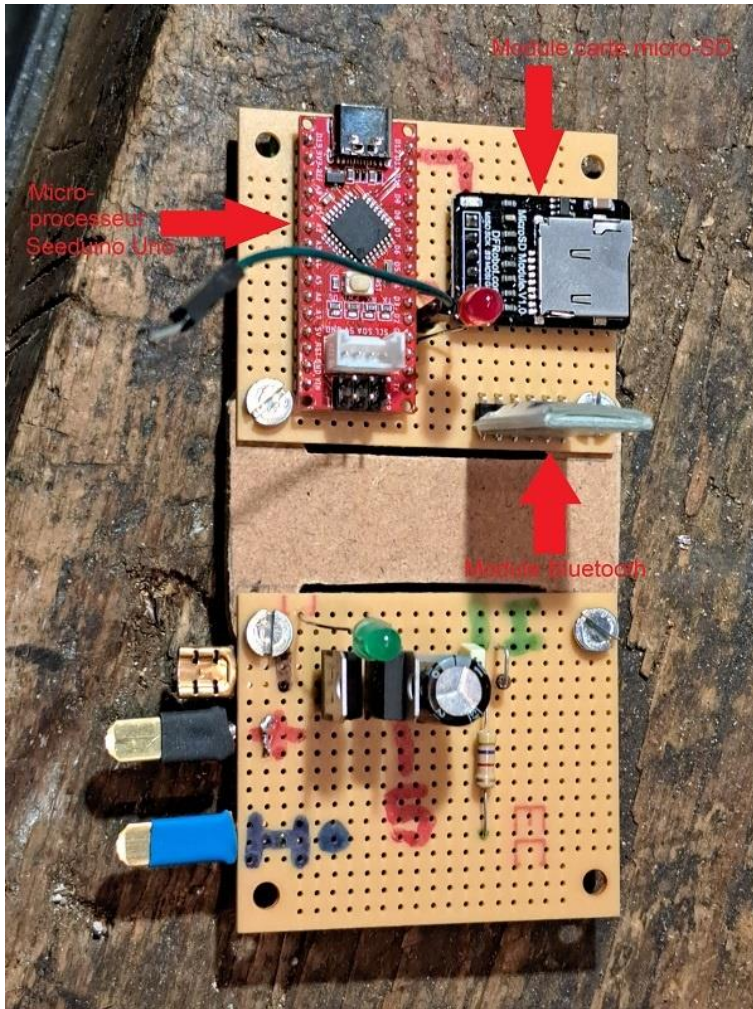
la première reçoit les impulsions de l'allumeur (borne RUP de la bobine) ce sont les mêmes impulsions qui alimentent le compte-tours au tableau de bord. L'Arduino déduit le régime en mesurant le temps (des microsecondes cette fois) écoulé entre 2 étincelles.

la seconde reçoit une tension 0-5V représentative de l'AFR, cette tension est directement fournie par le boîtier AFR (Innovate dans mon cas).

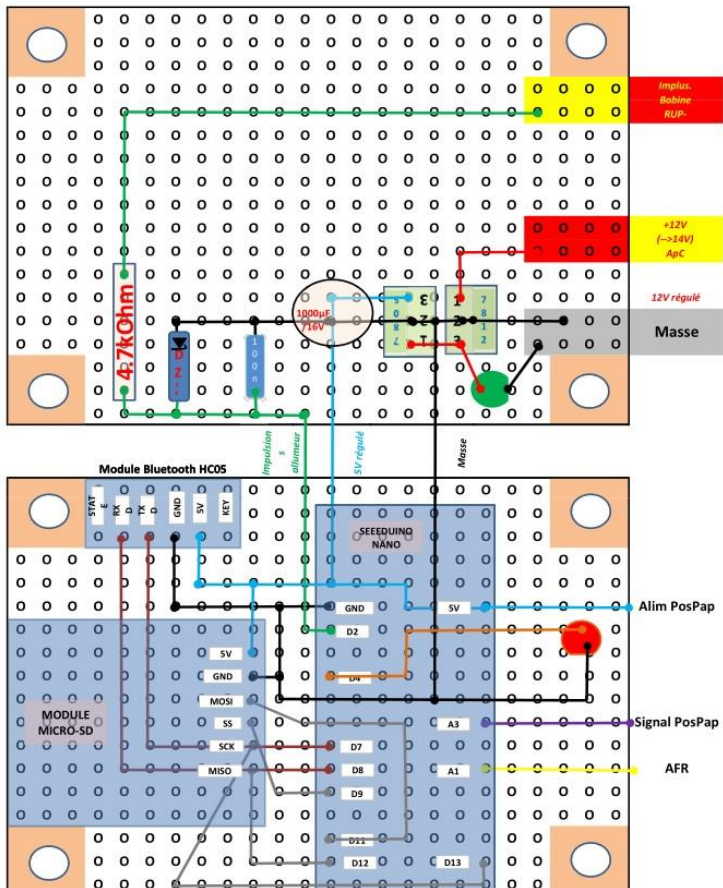
enfin la troisième reçoit une tension 0-5V représentative de la position du papillon des gaz. Je décrirai plus tard la façon dont j'ai créé cette grandeur.

Une image du "truc" en cours d'élaboration : en effet j'ai effectué plusieurs adaptations au cours des 6 mois que m'a nécessité ce projet.

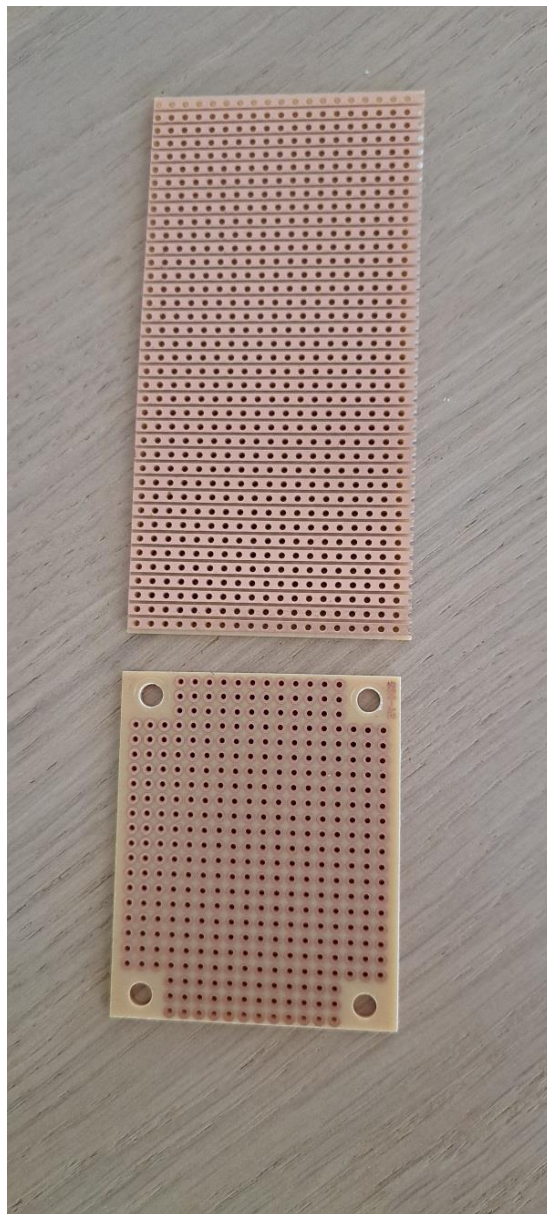
N'oubliez pas que je ne connaissais rien de rien sur le sujet et qu'il m'a fallu apprendre. Heureusement il y a eu des aides, principalement grâce à internet.



Et puis voici un schéma de mise en place et de câblage, nécessaire pour ne pas me perdre car j'ai utilisé des cartes à œillet puis les trouvant plus pratiques des cartes à bandes.



Je ne sais pas si ce sont les bons termes, voici une photo de carte à œillet (en bas) et de carte à bande (en haut).



Sur une première carte à œillet on trouve l'Arduino, le module Bluetooth, le module micro-carte SD et la diode rouge (enregistrement en cours).

Sur la seconde on trouve la partie alimentation comprenant un premier régulateur en 12V (pas forcément nécessaire) puis un second en 5V avec un gros condensateur (filtrage ?) et une diode verte signalant la mise sous tension.

A noter que j'ai utilisé des leds verte et rouge intégrant leur résistance de charge (quelques points de soudure en moins).

On y trouve aussi l'arrivée et le traitement des impulsions de l'allumeur (résistance et condensateur de découplage et diode Zener assurant que les impulsions ne dépassent pas 5V).

Une des évolutions a été provoquée par la présence de parasites sur l'enregistrement du régime moteur. Etant aussi une bille en électronique, c'est PhL qui m'a suggéré de remplacer la résistance et le condensateur uniques par 2 résistances moitiés et 2 condensateurs de même valeur.

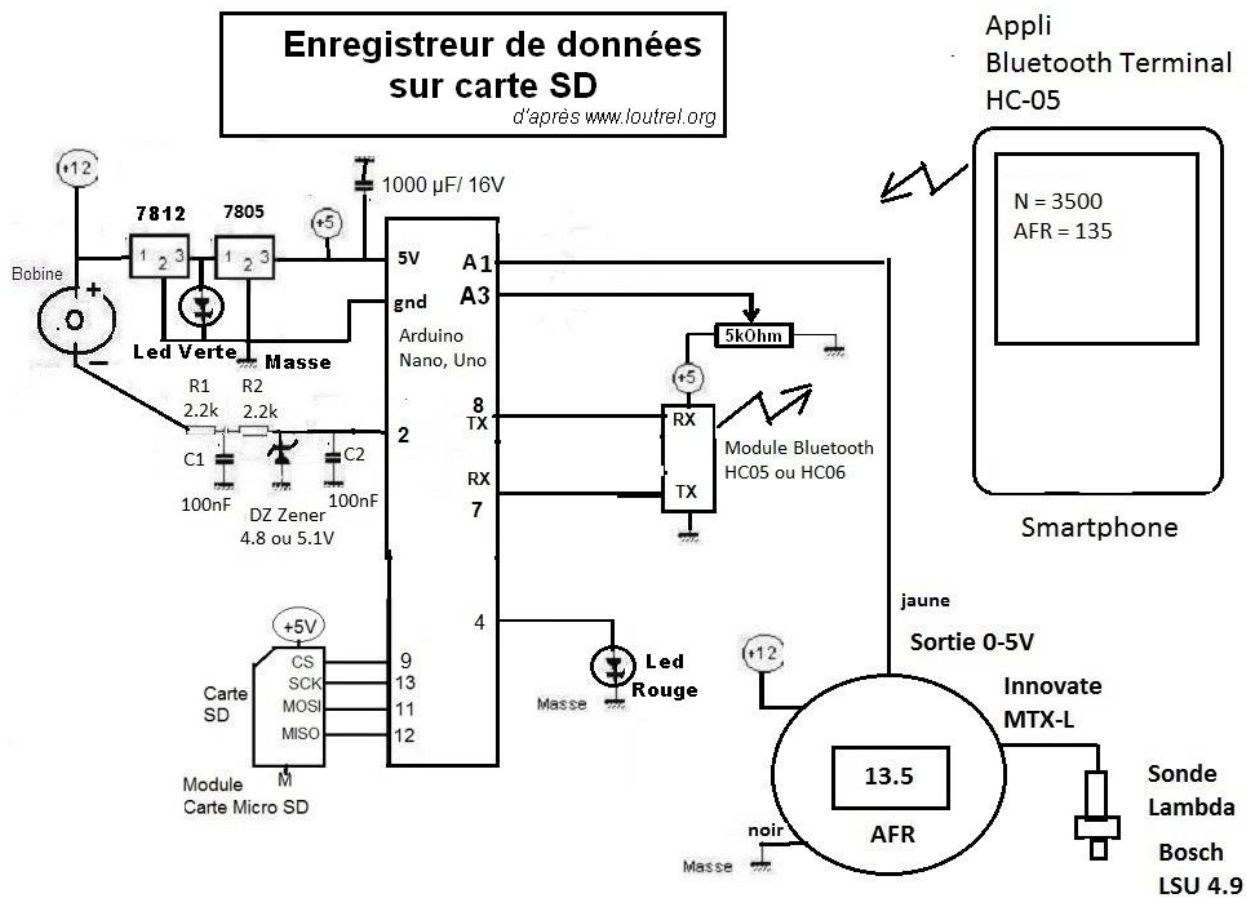
J'ai aussi rajouté une carte à bande pour faciliter les connexions vers l'extérieur. On la distingue sur la première photo du montage (celle avec le mètre-ruban). On y voit aussi les 2 résistances de 2,2k de l'antiparasitage des tr/mn mais les 2 condensateurs de 100nF sont masqués.

Pour faciliter les nombreux montages/démontages nécessaires à la mise au point du système j'ai implanté 2 connecteurs 3 voies. Comme ils sont du même modèle je les ai différenciés par la couleur (rouge / bleu).

Sur le bleu les 3 voies sont : +12V, masse, RUP-

Sur le rouge on trouve : +5V, AFR, Position Papillon

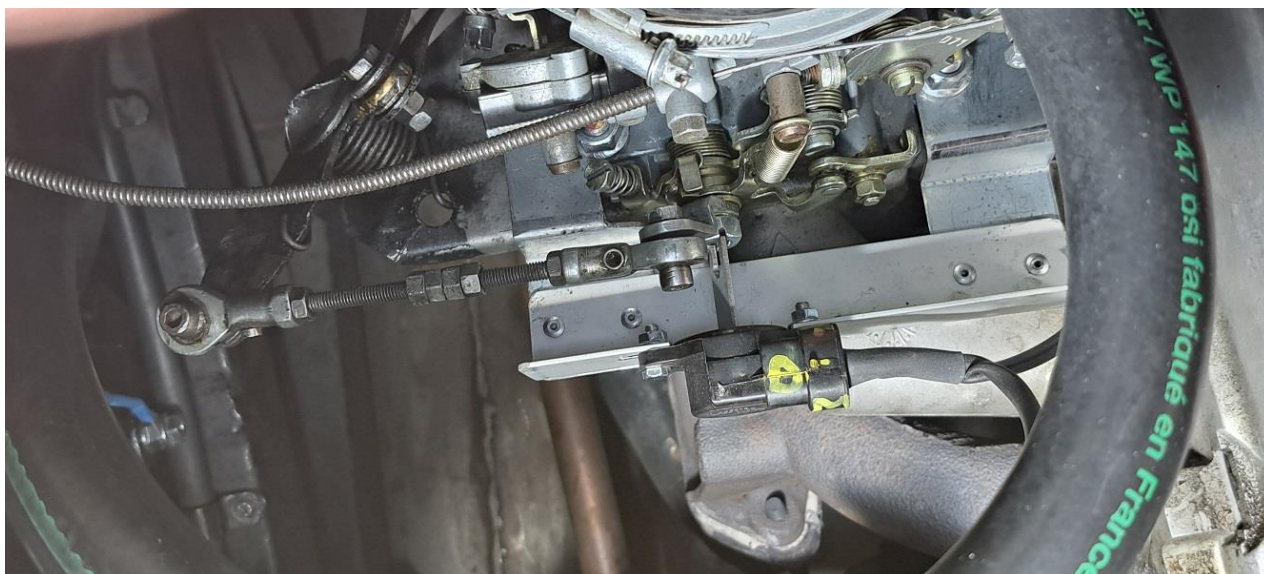
Voici le schéma du circuit, le dernier en date puisqu'il ya eu des évolutions.



La mesure de l'angle de papillon ; c'est pratiquement ma seule contribution à cet enregistreur : tout le reste je l'ai pompé ailleurs, modifié, adapté... donc j'en suis très fier !

Cela n'a pas été un long fleuve tranquille : j'ai commencé à vouloir utiliser un système existant puisque de nos jours l'angle papillon est une entrée de donnée nécessaire au fonctionnement des moteurs actuels. J'en trouve un du bon modèle (fonctionne en sens antihoraire), pas une ruine : 10€.

Je l'installe et au cours des essais je m'aperçois qu'il n'est pas linéaire : la tension monte proportionnellement à l'ouverture ; mais seulement jusqu'à un certain point, après ça redescend alors que ça devrait continuer à monter. J'imagine que j'ai dû le détériorer avec d'autres manip, j'en rachète un autre que je teste avant de le monter, neuf c'est pareil ; j'abandonne l'idée.



J'ai besoin d'un potentiomètre qui donne X kOhm à 0% d'ouverture et Y kOhm à 100% d'ouverture : c'est simple ! Je choisis une valeur de 5kOhm car sous 5V de tension d'alimentation cela ne fait qu'un courant négligeable traversant les fils et l'Arduino. Premier essai concluant, sauf qu'au retour du roulage d'essai le potentiomètre explose : ses 2 demi-boîtiers en plastique divorcent sous l'effet de la température dans le compartiment moteur.



C'était « pas assez cher, mon fils ! », j'achète un potentiomètre à carter métallique.



Le principe étant trouvé, le plus dur est d'accoupler l'axe de papillon du 1er corps à l'axe du potentiomètre, tout en maintenant le potentiomètre face au carburateur.

Je fabrique une console se fixant par-dessus la semelle du carbu et récupérant 3 de ses 4 fixations. Le potentiomètre est implanté sur cette console en regard de l'axe du papillon.



Pour l'accouplement je fais un trait de scie sur l'axe du potentiomètre (dont la longueur est ramenée à une valeur « kivabien »). Je fais un autre trait de scie sur l'écrou M8 (*) qui verrouille l'empilage de pièces sur l'axe de papillon et entre les 2 j'intercale un bout de tôle « d'à-peu-près » la largeur des traits de scie. Ça devrait marcher !

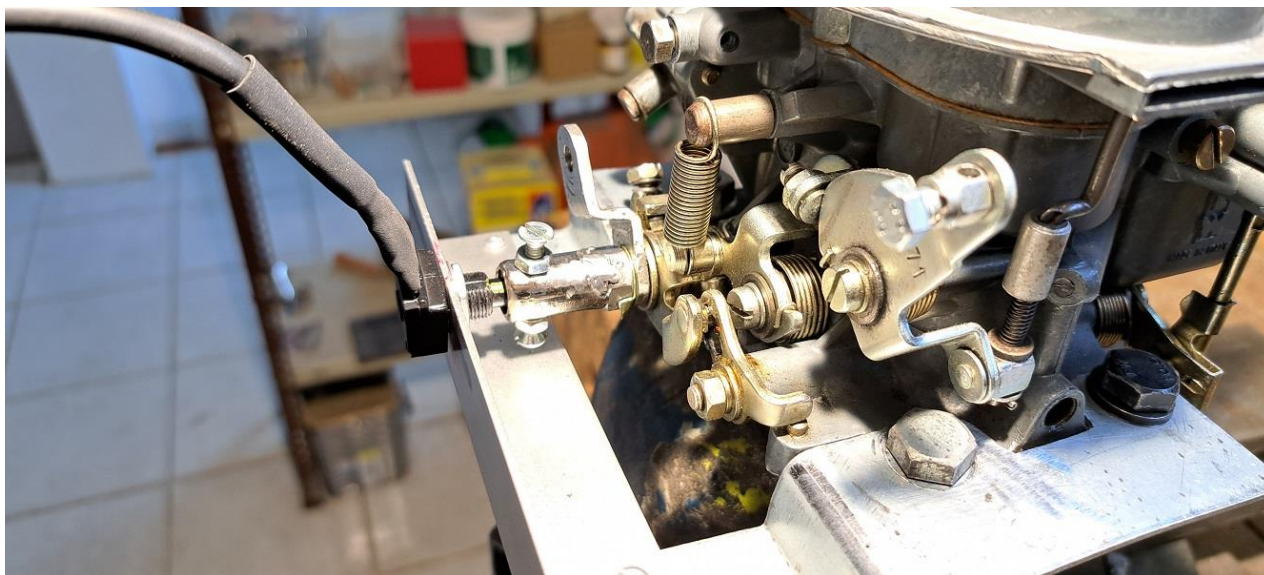


(*) l'écrou Weber d'origine est en M8 MAIS pas fin et en hauteur réduite, j'ai utilisé des écrous M8-pas fin hauteur normale qui donc dépassent de l'extrémité fileté de l'axe de papillon.

Ça devrait marcher ! Oui ; mais ça ne marche pas très bien : il y a trop de jeu et d'imprécision entre les traits de scie et la tôle, les mesures ne sont pas satisfaisantes !

J'abandonne cet accouplement et me lance dans un autre domaine inconnu pour moi : je soude un bout de tube sur un «écrou M8-pas fin hauteur normale ».

Le bout de tube a un diamètre intérieur juste supérieur au diamètre de l'axe du potentiomètre et une épaisseur suffisante pour pouvoir y implanter une (ou 2) vis de pression assurant l'entraînement.

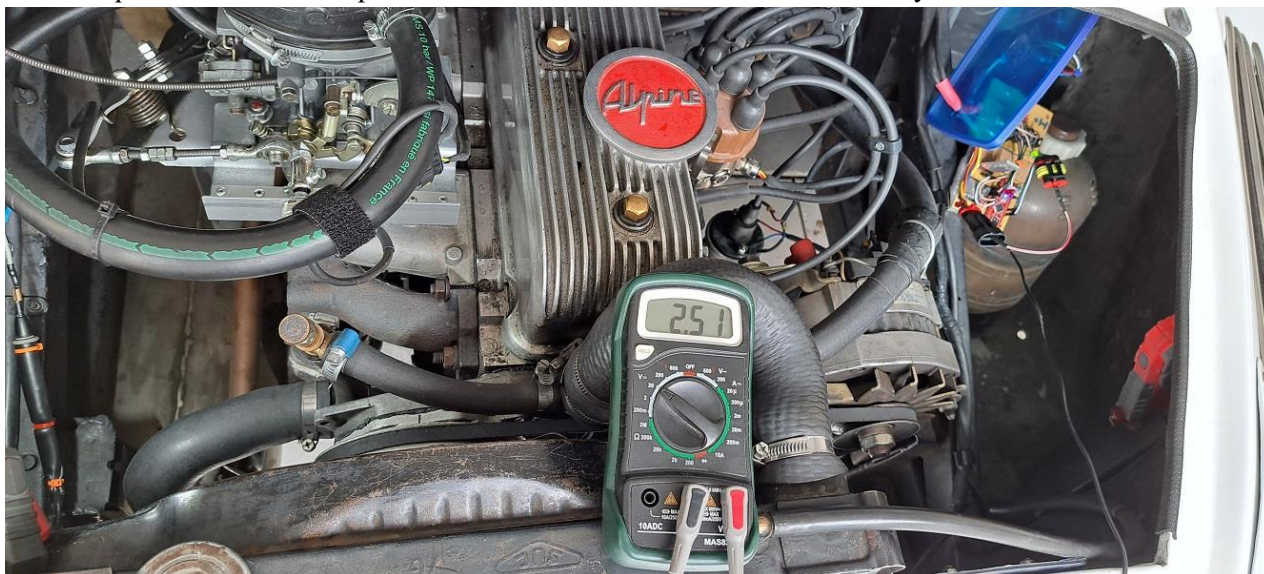


Phase d'étalonnage de la mesure de l'angle du papillon:

Un étalonnage est nécessaire afin de pouvoir informer l'Arduino que xxVolts correspondent à 0% d'ouverture et yyVolts correspondent à 100% d'ouverture.

En effet on ne peut pas avoir 0Volt = 0% et 5 Volts =100% car la course angulaire du potentiomètre (300°) et bien plus importante que celle du papillon des gaz (environ 90°).

J'ai fabriqué 2 fils volants me permettant d'intercaler mon voltmètre dans le système.



Je peux donc y lire xx & yy sans démarrer le moteur, juste en alimentant l'ensemble contact mis.

Ce sont ces étalonnages successifs qui m'ont permis d'éliminer mes premières réalisations peu précises ou peu reproductibles.

Sur la photo on lit que 2,51V correspondent à 0% d'ouverture du papillon (ralenti).

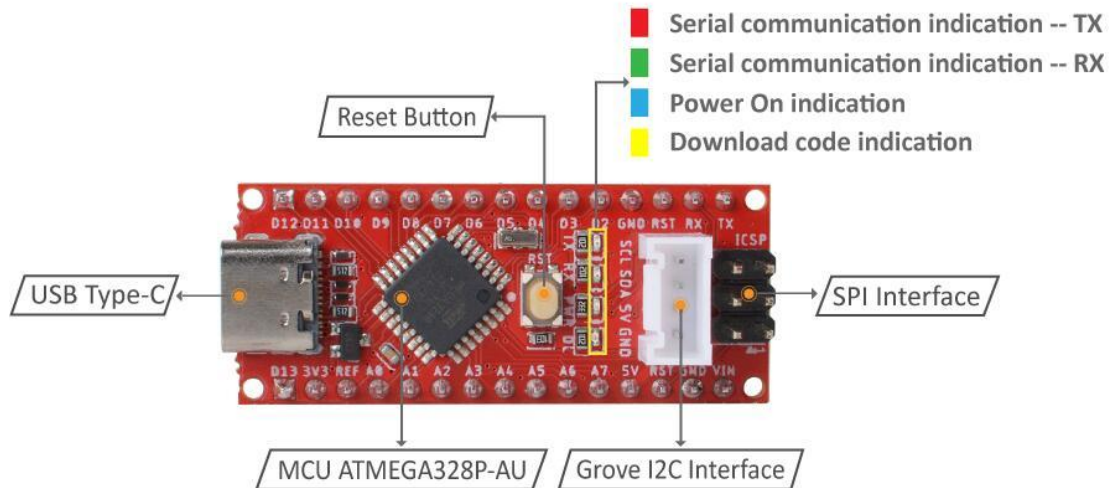
Le même montage m'a aussi permis de mesurer que l'ouverture du second corps commence à environ 66%/70% d'ouverture du 1er (imprécision due aux jeux dans les palonniers de commande).

Après avoir installé l'ensemble dans l'auto avec des câblages provisoires mais bien protégés car le système va quand même y rester certainement plusieurs mois, la suite c'est la description de la partie logicielle.

Partie logicielle

Les Arduino comprennent un jeu d'instruction qui vont permettre de commander le microprocesseur, lui dire quoi faire dans tel ou tel cas.

La liste de ces instructions est préparée sur son PC et est ensuite transférée (ils disent « téléversée ») dans l'Arduino par liaison USB ; une prise USB-C est intégrée à l'Arduino, elle apparaît sur cette photo. C'est elle qui assure aussi l'alimentation de l'Arduino lorsqu'il est connecté au PC.



Les instructions sont préparées sur le PC après avoir installé le logiciel de communication opérateur-PC en libre accès qui se nomme « Arduino IDE ».

Il y a une communauté des utilisateurs d'Arduino très riche – même si souvent en anglais – qui m'a beaucoup aidé. On trouve ainsi la définition et la description complète de chaque instruction (en français) ce qui permet de s'y reporter lorsqu'on a un doute sur la syntaxe ou sur ce que commande exactement une instruction.

J'ai utilisé cette richesse tout d'abord pour tester la communication entre l'Arduino et le module Bluetooth d'une part et le module carte micro-SD d'autre part. Par exemple voici la liste d'instructions (ils disent « le croquis ») qui m'a permis de tester la liaison Bluetooth :

```
#include "SoftwareSerial.h"
SoftwareSerial mySerial(7,8);
int val;
void setup() {
  mySerial.begin(9600);
  pinMode(4, OUTPUT);
}
void loop() {
  if (mySerial.available()) {
    val = mySerial.read();
    if (val == 'L') {digitalWrite(4, HIGH);}
    if (val == 'A') {digitalWrite(4, LOW);}
  }
}
```

Lorsque je tape « L » au clavier du Smartphone, la led rouge (qui servira à visualiser qu'un enregistrement est en cours) s'allume ; lorsque je tape « A » elle s'éteint.

Il faut ouvrir une parenthèse concernant la communication avec le Smartphone.

Sur celui-ci il faut télécharger une application de communication entre son clavier & son écran d'une part

et sa propre carte BlueTooth d'autre part. Il en existe plusieurs, j'ai téléchargé « Bluetooth Terminal HC-05 ». Une fois qu'on a appairé le Smartphone et l'Arduino le fonctionnement de cette couche de communication homme/module BlueTooth est transparent.

Ensuite

Maintenant qu'on sait communiquer entre l'Arduino et le PC, le module BlueTooth ou le module carte micro SD, on peut passer à la mise au point du croquis/programme principal permettant de transformer cet ensemble en un enregistreur de données.

Ben là j'ai fait comme le Bernard-l'hermite (celui qui s'installe dans la coquille d'un autre).

J'ai chargé dans IDE les 2 croquis mis à disposition par PhL : celui pour un moteur à carbu et celui pour un moteur injection. Et j'ai tâché de comprendre les similitudes et les différences entre les 2 afin d'en composer un répondant à mes contraintes.

Je reconnais que je n'ai compris que les grandes lignes, il y a certaines subtilités qui me sont passées au-dessus de la tête. Cependant un croquis pour Arduino a toujours la même structure, il est composé de 2 parties principales et d'un nombre variable d'annexes.

La première partie principale est exécutée une fois au lancement du programme, on y trouve des choses comme les définitions des variables avec leur type, utilisant plus ou moins de mémoire. On y déclare à quoi sont affectées les différentes bornes de l'Arduino, par exemple : la borne A1 est une borne qui reçoit un signal analogique 0-5V provenant de l'Innovate et donc représentatif de l'AFR, alors que la borne 4 est une sortie qui commande la led rouge.

La seconde partie principale est une boucle, elle exécute un certain nombre d'instructions et quand elle a fini sa liste elle recommence au début jusqu'à ce qu'une condition se réalise et l'arrête.

Les annexes sont en fait des routines qui exécutent chacune un certain nombre d'instructions. Chaque fois qu'on a besoin d'exécuter ces instructions on appelle ce sous-programme plutôt que de copier à chaque fois la liste des instructions concernées.

Un sous-programme peut par exemple avoir comme tâche de lire la tension sur la borne A1 et inscrire la valeur correspondante de l'AFR dans une variable qu'on aura justement nommée AFR.

Mais ce peut-être : « Afficher sur le Smartphone les valeurs des 3 grandeurs mesurées ainsi que le temps écoulé depuis le début de l'enregistrement ».

Voici maintenant un descriptif synthétique du croquis/programme.

La base de tout est le comptage par l'Arduino des impulsions d'allumage. Dans notre cas d'un 4 cylindres, l'Arduino réagit toutes les 4 étincelles (2 tours de vilebrequin).

Tant que le Smartphone n'a pas envoyé un caractère (n'importe lequel, majuscule, minuscule, chiffre) par Bluetooth, l'Arduino est en attente d'enregistrement.

Il va relever les 3 valeurs – tr/mn, AFR, position du papillon - et se contenter de les afficher sur l'écran du Smartphone. Cet affichage se fait toutes les 2 secondes et non tous les 2 tours de vilo, sinon cet affichage serait illisible.

Dès qu'un caractère quelconque est envoyé par le Smartphone (commande de l'utilisateur), l'Arduino passe en mode enregistrement : il allume la led rouge et chaque fois qu'il lit les 3 valeurs il les enregistre sur la carte SD, tout en continuant leur affichage toutes les 2 secondes.

L'enregistrement sur la carte se fait à un format compatible avec Excel , ce qui permettra le dépouillement, j'y reviendrai.


Bien sur il y a tout un tas d'autres annexes que je ne décris pas, comme par exemple avant d'enregistrer, s'assurer qu'il y a une carte dans le lecteur, qu'elle est correctement formatée qu'on peut y écrire sans effacer ce qui y est déjà, etc.....

Lorsqu'à nouveau un caractère est entré à l'aide du clavier du Smartphone l'enregistrement est alors arrêté. Le clavier fonctionne donc comme un bouton poussoir qui démarre puis arrête l'enregistrement : premier appui sur le bouton (=premier caractère, n'importe lequel, entré au clavier) démarre l'enregistrement ; puis second appui sur le bouton (=nouveau caractère, toujours n'importe lequel, entré) arrête l'enregistrement.

Et enfin ?

On est prêt pour rouler et enregistrer.

Et on obtient un fichier au format tableur Excel dont voici le début :



	A	B	C	D	E	F
1	Enregistrement tr/mn, AFR & angle papillon (d'après www.loutrel.org)					
2						
3	Temps ms	Nt/mn	AFR	% Ouv. Pap.	Tr/mn par pas	% Ouv par pas
4	1143	2470	118	18	2500	20
5	1191	2455	116	15	2500	20
6	1241	2458	112	14	2500	10
7	1290	2502	105	15	2500	20
8	1338	2435	109	11	2500	10
9	1387	2448	105	16	2500	20
10	1436	2435	101	14	2500	10
11	1485	2421	100	14	2500	10
12	1535	2439	106	11	2500	10
13	1584	2427	96	14	2500	10
14	1633	2448	103	13	2500	10
15	1682	2450	103	13	2500	10
16	1731	2429	104	9	2500	10
17	1781	2445	101	10	2500	10
18	1830	2406	117	14	2500	10
19	1881	2393	113	10	2500	10
20	1931	2420	108	12	2500	10
21	1981	2405	112	13	2500	10
22	2031	2388	106	13	2500	10
23	2081	2399	118	10	2500	10
24	2132	2396	113	11	2500	10
25	2182	2406	113	14	2500	10
26	2232	2374	111	14	2250	10
27	2282	2372	121	12	2250	10
28	2333	2365	110	11	2250	10
29	2384	2354	116	14	2250	10
30	2436	2352	121	14	2250	10
31	2487	2370	121	11	2250	10
32	2537	2387	120	13	2500	10
33	2587	2421	115	13	2500	10

On voit bien les colonnes correspondant à ce qu'on recherche :

- * Colonne A : le temps
- * Colonne B : le régime moteur en tr/mn
- * Colonne C : l'AFR (x10)
- * Colonne D : l'angle d'ouverture du papillon en %.

Il y a 2 autres colonnes dont je n'ai pas parlé.

L'objectif est de comparer l'AFR à une valeur cible, mais à l'intérieur de cases d'une certaine longueur (tr/mn) et d'une certaine hauteur (% ouverture papillon).

On n'a donc pas besoin de la valeur précise au tr/mn près du régime moteur, ni de la valeur précise au % près de l'ouverture papillon.

J'ai créé 2 variables supplémentaires (une pour tr/mn et l'autre pour %) calculées par l'Arduino qui donne la valeur arrondie par pas des tr/mn & %.

On lit alors :

- * Colonne E : le régime moteur par pas de 250tr/mn (uniquement des valeurs multiples de 250)
- * Colonne F : l'ouverture du papillon par pas de 10% (uniquement 0%, 10%, 20%, 30%....., 100%)

Pour l'instant je n'ai pas roulé en recherchant à reproduire telle ou telle configuration particulière de façon stable pendant plusieurs secondes.

Mais j'ai fait des enregistrements de roulage aussi long que 45 minutes de balade.

Le nombre de lignes du fichier Excel qui en résulte est faramineux : 45 minutes = 59100 lignes dans le

fichier !

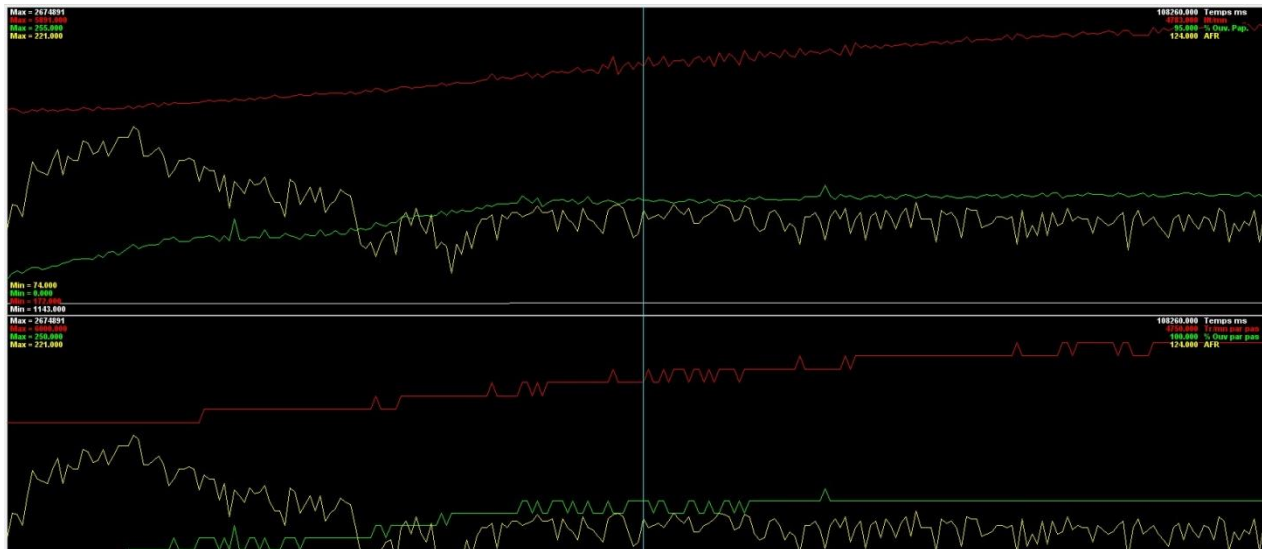
Première satisfaction le fichier est tout petit (1674ko alors que la moindre carte micro-SD en accepte plusieurs Go) donc pas de problème de stockage ni de vitesse d'acquisition.

La difficulté cependant est l'interprétation/dépouillement : Excel ne permet pas de manipuler confortablement autant de lignes.

Sur les recommandations de PhL j'ai téléchargé un logiciel qui transforme un tel fichier en courbes.

Il s'agit de MegaLogViewer, la version de base gratuite suffit largement à mon utilisation.

C'est ce logiciel qui m'a permis de poster l'image qui figure tout en tête de ce sujet.



Nota pour comprendre qu'on n'a pas besoin des valeurs précises des tr/mn et % d'ouverture mais plutôt des valeurs arrondies par palier :

Si on regarde le graphique qui indique les valeurs cibles on a seulement 3 zones :

Le tableau des valeurs d'AFR cibles (objectif).

Ouverture Papillon (charge)	100 %	AFR compris entre 12.5 et 13.5 (priorité puissance)							
	90 %	AFR compris entre 12.5 et 13.5 (priorité puissance)							
	80 %	AFR compris entre 12.5 et 13.5 (priorité puissance)							
	70 %	AFR entre 13 et 14 (stabilité)	AFR entre 13.5 et 14.5 (économie)				AFR compris entre 12.5 et 13.5 (priorité puissance)		
	60 %		AFR entre 13.5 et 14.5 (économie)						
	50 %		AFR entre 13.5 et 14.5 (économie)						
	40 %		AFR entre 13.5 et 14.5 (économie)						
	30 %	AFR compris entre 12.5 et 13.5 (priorité puissance)							
	20 %	AFR compris entre 12.5 et 13.5 (priorité puissance)							
	10 %	AFR compris entre 12.5 et 13.5 (priorité puissance)							
		1500	2000	3000	4000	5000	6000	7000
Régime Moteur									

1- celle dite de priorité puissance (AFR entre 12.5 & 13.5)

2- celle dite de stabilité (AFR entre 13 & 14)

3- celle d'économie (AFR entre 13.5 & 14.5)

Carte des résultats

Pour compléter le dépouillement je me suis créé un fichier Excel avec 3 onglets

* le premier onglet comporte des cases représentant la cible, c'est une copie sur tableur des valeurs cibles montrées au-dessus. J'ai seulement divisé par 4 le pas du régime moteur (pas de 1000tr/mn → 250tr/mn).

* le second onglet possède les mêmes cases vierges et j'y inscris les AFR lus par MegaLogViewer à partir du fichier enregistré sur la micro-SD.

* le dernier calcule la différence des 2 et me montre si je suis trop riche (en rouge) ou trop pauvre (en bleu) et de combien.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC		
1	Valeurs Cible AFR (objectifs) fonction du régime moteur & de l'ouverture des gaz (charge)																														
2	% ouverture papillon	100%	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	
3		90%	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
4		80%	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
5		70%	13,5	13,5	13,5	13,5	13,5	14	14	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13
6		60%	13,5	13,5	13,5	13,5	13,5	14	14	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13
7		50%	13,5	13,5	13,5	13,5	13,5	14	14	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13
8		40%	13,5	13,5	13,5	13,5	13,5	14	14	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13
9		30%	13,5	13,5	13,5	13,5	13,5	14	14	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13
10		20%	13,5	13,5	13,5	13,5	13,5	14	14	14	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13
11		10%	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5
12		0%	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5	13,5
13			500	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000	6500	7000	Régime moteur (tr/mn)														
14	Valeurs Cible AFR (objectif)																														
15	13	AFR de... à...	12,5	13,5																											
16	13,5	AFR de... à...	13	14																											
17	14	AFR de... à...	13,5	14,5																											

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD					
1	Interprétation des enregistrements (Fichiers 36.csv & 38.csv = environ 40 minutes de roulage)																																		
2	% ouverture papillon	100%																														2ème corps ouvert			
3		90%																																	
4		80%																																	
5		70%																																	
6		60%																																	
7		50%																																	
8		40%																																	1er corps seul
9		30%																																	
10		20%																																	
11		10%																																	
12		0%																																	
13			500	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000	6500	7000	Régime moteur (tr/mn)																		
14	Légende des résultats																																		
15	Trop riche = nombre négatif																																		
16	Trop pauvre = nombre positif																																		

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC						
1	Données relevées sur fichiers d'enregistrement (Fichiers 36.csv & 38.csv)																																		
2	% ouverture papillon	100%																																	
3		90%																																	
4		80%																																	
5		70%																																	
6		60%																																	
7		50%																																	
8		40%																																	
9		30%																																	
10		20%																																	
11		10%																																	
12		0%																																	
13			500	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000	6500	7000	Régime moteur (tr/mn)																		
14	Valeurs Cible AFR (objectif)																																		
15	AFR de... à...	12,5	13,5																																
16	AFR de... à...	13	14																																
17	AFR de... à...	13,5	14,5																																

CONCLUSION

J'ai mis au point un outil qui va me permettre de m'appuyer sur des valeurs pour tâcher d'ajuster la carburation de ce moteur.

J'ai tenu à expliquer de façon exhaustive toute ma démarche pour que non seulement cela serve à ceux qui voudraient se lancer dans une entreprise similaire mais aussi pour montrer que même ne connaissant pas un domaine, on arrive à y faire des choses.

On trouve énormément d'info sur internet et je dois aussi remercier PhL qui m'a vraiment « mis le pied à l'étrier » : s'il n'avait pas défriché de façon approfondie le terrain, je n'aurais jamais imaginé me lancer.

Il me reste maintenant un autre défi : je ne connais rien aux carburateurs et il va bien falloir que je trouve ce que je dois changer si c'est trop riche ici et trop pauvre là.