



GEII
Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Testeur de sonde lambda

RAPPORT DE PROJET TUTORE

Kohler André
Ruffio Anthony

Mr Lombard Christophe





GEII
Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Remerciements

Nous souhaitons remercier tous particulièrement Monsieur Lombard, notre tuteur tout au long de notre projet. Nous avons pu le solliciter, lui demander conseils, il a su nous aider grâce à ses connaissances.

Nous souhaitons également remercier Monsieur Gavignet qui a su nous apporter des conseils durant le déroulement de notre projet.

Et nous souhaitons aussi remercier nos camarades de classe à qui nous avons pu demander de l'aide et des conseils pour réaliser notre projet.



GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Sommaire

Table des matières

I) Présentation du projet :	4
1) Introduction	4
2) Mise en contexte	5
3) Déroulement du projet	6
4) Cahier des charges	7
5) Application système	8
II) Développement	9
1) Présentation du matériel	9
2) Logiciels utilisé	10
3) Méthodes de mesure	11
4) Communication I2C	12
5) Test communication écran	12
6) Communication CAN	13
7) Boîtier	13
III) Réorganisation du projet	14
1) Contexte	14
2) Présentation matérielle	14
3) Présentation logicielle	15
4) Stratégie	15
5) Communication	16
IV) Conclusion	16
1) Enseignement tirés	16
2) Résultats du projet	17
3) Perspectives futures	17



I) Présentation du projet :

1) Introduction

La sonde lambda, également appelée sonde à oxygène a pour rôle d'améliorer l'efficacité du moteur thermique d'un véhicule. Elle a été introduite par la norme Euro 1 en 1993 sur les véhicules essence.

Une sonde lambda est composée d'un corps en céramique pour résister aux hautes températures mais aussi suivant le type de sonde de zirconium ou de titane. La température nominale de fonctionnement est de 300°C à 600°C.

Sur les voitures les plus récentes il y a deux sondes lambda une avant et une après le catalyseur.



Les mesures de ces sondes servent à contrôler la quantité d'oxygène présente dans les gaz d'échappement ce qui permet de modifier l'injection afin de réduire les émissions polluantes du moteur.

On utilise la lettre grecque lambda (λ) qui donne un coefficient d'air lambda qui est égal à 1 lorsque le rapport air/essence optimal de 14,7 grammes d'air pour 1 gramme d'essence. La valeur du lambda doit être comprise entre 0,97 à 1,03 au-delà le taux de pollution est incorrect.

Pour que ces mesures de coefficient soient possibles il faut que le moteur soit à température. Si on mesure la tension aux bornes de la sonde on mesure une tension entre 0,1 et 0,9 V. Quand on se rapproche de 0,1 V le mélange est trop pauvre et vers 0,9 V le mélange est trop riche.



Les sondes lambda servent également à vérifier l'état du catalyseur.
Une sonde lambda a une durée de vie théorique d'environ 150.000km variable suivant les conditions d'utilisation (présence d'essence ou d'huile en excès lors d'une combustion).

Sonde lambda avec son connecteur



2) Mise en contexte

Notre projet a pour objectif de développer un boîtier autonome, dans le domaine automobile qui permet de mesurer la tension d'une sonde lambda par la méthode analogique ou via la prise OBD 2 du véhicule (interrogation du calculateur).

Ce boîtier doit également mesurer l'état du catalyseur, via la mesure de dépollution effectué entre les sondes lambda amont et aval.

Pour cela nous allons proposer une solution matérielle et logicielle qui sera destinée à diagnostiquer les deux sondes Lambda ainsi que le catalyseur.

Ce boîtier doit être autonome en énergie et doit être plug&play.

Le boîtier doit être facile d'utilisation et de compréhension.



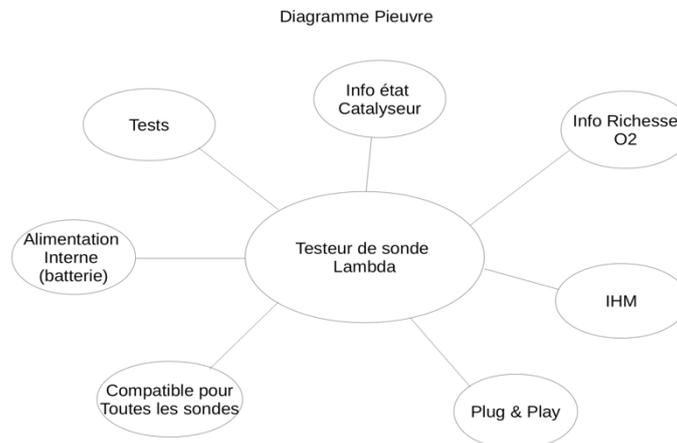
Testeur sonde lambda via prise OBD 2



Testeur sonde lambda aux bornes de la sonde

3) Déroulement du projet

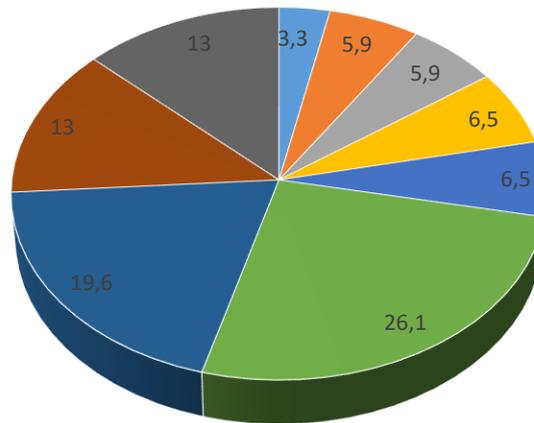
Diagramme de pieuvre qui présente les différentes fonctions du projet



Prévision du temps passé sur le projet



Temps passé en %



- Découverte et prise en main du sujet
- Analyse fonctionnelle
- Etudes et choix composants
- Conception du boîtier
- Câblage des différents composants
- Conception du programme
- Conception de l'IHM
- Simulation des sondes et du catalyseur
- Validation et tests

D'après le diagramme du temps prévisionnel ci-dessus nous avons étudié les différentes tâches à réaliser durant les phases du projet ainsi que le temps que cela pourrait prendre.

Il y a d'abord une phase de découverte, prise en main et planification du projet.

Ensuite une partie programmation et simulation des sondes Lambda

Une autre pour rendre le boîtier autonome et insérez tous les composants nécessaires à l'intérieur

Et pour finir une phase de finalisation, de validation et de test.

4) Cahier des charges

Pour notre projet, nous avons établis le cahier des charges suivant :

Le boîtier est un outil pour les techniciens automobiles, ainsi que les particuliers.

Le boîtier doit être compatible avec plusieurs technologies de connectique (Analogique, OBD2).



Le boîtier doit également être autonome en énergie.

Mais également de pouvoir tester plusieurs types de sondes lambda (Titane, Zirconium), également de vérifier l'état du catalyseur.

En ce qui concerne les coûts, nous n'avons de budget précis à respecter, néanmoins nous avons eu une contrainte de temps pour commander.

En effet le budget universitaire ferme la possibilité de passer commande de fin octobre et est à nouveau ouvert courant janvier 2018. Donc nous avons dû commander les composants avant fin octobre pour pouvoir réaliser des tâches de notre projet.

La principale contrainte du projet est que les tests sur une "vraie" sonde est impossible car pour que les sondes fonctionnent il faut qu'elles soient à leur température nominale de fonctionnement or dans une salle de TP reproduire cette situation est compliquée car la température doit être comprise entre 300 à 600°C et pour cela il faut démarrer un véhicule qui émet des gaz d'échappements.

C'est pour cela que nous allons simuler le fonctionnement des sondes via un script CAPL mais aussi avec un potentiomètre.

5) Application système

Pour proposer les deux méthodes de mesure il nous faut effectuer les mesures via analogique ou réseau CAN puis les transmettre via des programmes pour le traitement puis l'affichage sur un écran.

Pour simuler la sonde de façon analogique nous utilisons un potentiomètre afin de faire des variations de tension.

Nous utilisons un script en CAPL pour la simulation des sondes sur le réseau CAN accessible via un câble avec connecteur DB9.

Le boîtier doit être autonome et portable il ne doit pas dépendre d'une prise secteur pour cela nous prévoyons d'implanter un système de batterie pour le rendre indépendant



II) Développement

1) Présentation du matériel

Pour la partie matérielle de notre projet nous avons besoin :

D'un PC équipé du logiciel Canalyzer pour la simulation de la sonde.

D'un potentiomètre pour la simulation analogique de la sonde.

D'un écran OLED pour l'affichage des informations.

D'une plaque Labdec pour le branchement de l'écran OLED.

D'une batterie 12v.

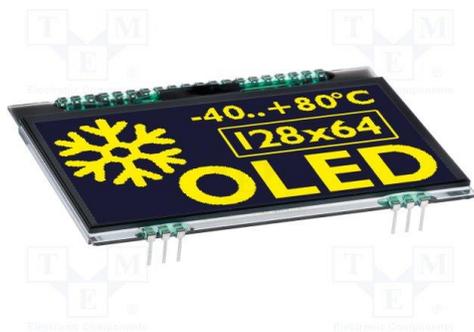
D'un boîtier.

D'un câble OBD2 vers DB9 pour assurer la liaison entre la prise OBD2 et la carte.

De connecteurs de sondes.

D'une carte Cypress Semiconductor

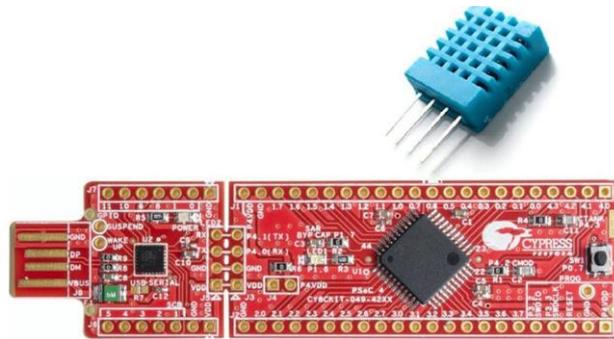
Pour l'alimentation de l'écran on a besoin de divers condensateurs, résistances et un régulateur de tension LM 317



Ecran OLED Jaune 128x64



Câble DB9 vers OBD II



Carte Cypress Perform

2) Logiciels utilisé

2.1) CANalyzer

C'est un logiciel d'analyse réseau très utilisé dans le monde automobile, il permet de communiquer entre un pc équipée d'une carte Vector et les différents réseaux d'un véhicule. Dans ce logiciel on peut également coder en CAPL (Communication Access Programming Language), c'est un langage de programmation adapté à l'automobile le CAPL est basé sur du langage en C avec l'ajout de fonctions spécifique à l'automobile comme l'envoi de messages CAN.

2.2) Pso creator

C'est un logiciel de développement conçu pour les cartes Cypress Pso, ce logiciel permet une programmation software mais également hardware de la carte. La particularité de ces cartes est que les circuits sont complètement configurables.

2.3) DesingSpark Mechanical

C'est un logiciel de modélisation 3D gratuit qui permet de modéliser diverses pièces mais également de créer des fichiers utilisables avec des imprimantes 3D.



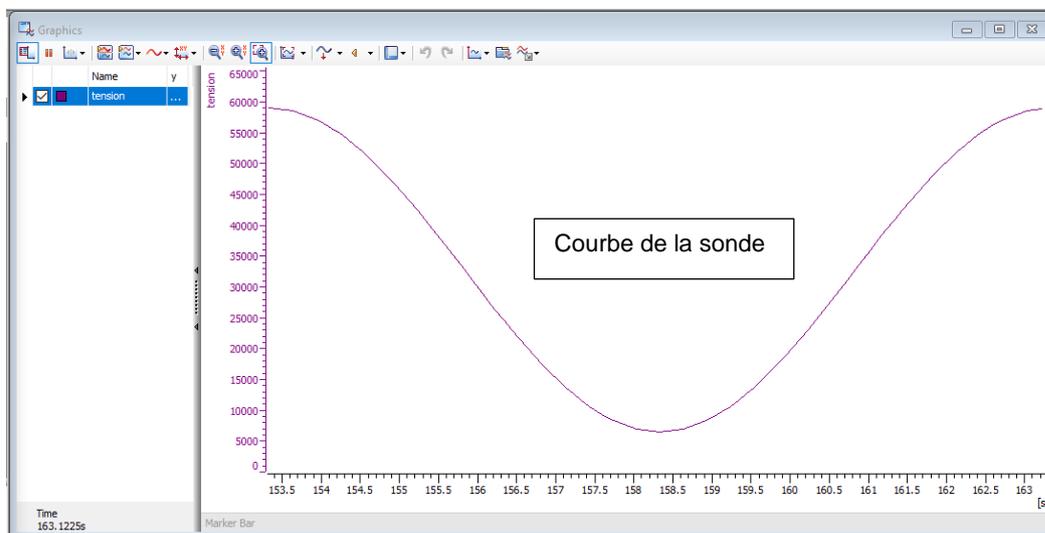
3) Méthodes de mesure

Pour tester le fonctionnement d'une sonde lambda il nous a fallu simuler son fonctionnement car nous ne pouvions pas réaliser de test pratique.

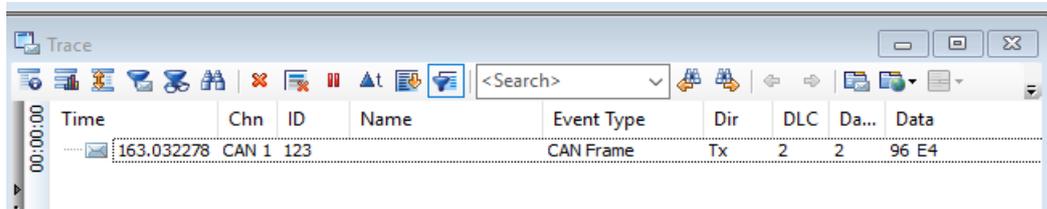
Le boîtier doit réaliser des tests en récupérant le signal en se connectant sur l'OBD2 du véhicule, pour simuler un véhicule nous avons utilisé le logiciel CANalyzer avec un programme en CAPL. Ce programme permet de simuler le signal périodique d'une sonde lambda en fonctionnement optimal.

Pour que l'on puisse récupérer ces données sur la trame adéquate afin de la transformer en tension avec notre carte Cypress PsoC et l'afficher sur un écran OLED.

Sur l'image ci-dessous on voit une capture d'écran du logiciel CANalyzer avec le programme CAPL qui permet de générer le signal que peut émettre de la sonde.



Sur la fenêtre ci-dessous on voit la trame qui sert à récupérer le signal périodique de la sonde, sur une variable en entier. La trame ayant pour adresse 0x123 dont les données se trouvent sur deux octets (soit 16 bits) sera convertie en variable réelle pour pouvoir visualiser la tension de la sonde et pour l'afficher sur l'écran à l'aide du logiciel PsoC et la carte Cypress.



En fonctionnement normal pour mesurer la tension d'une sonde lambda on doit mesurer directement sur les bornes de la sonde. Pour simuler cette sonde nous avons choisi de simuler celle-ci via un potentiomètre qui simule la tension du signal.

4) Communication I2C

Nous devons communiquer entre la carte Cypress et notre écran OLED pour cela nous utilisons la communication en I2C. Pour notre application l'utilisation de ce bus permet l'envoi de données vers l'écran et d'afficher des informations sur celui-ci. Nous utilisons 4 fils, un pour le signal de données (SDA), un signal d'horloge (SCL), une alimentation et une masse.

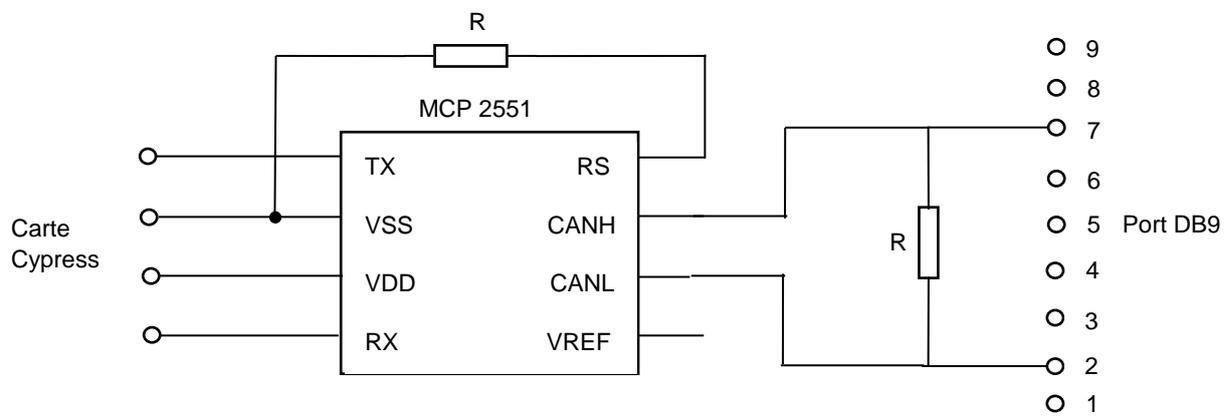
5) Test communication écran

Pour tester la communication I2C nous avons décidé de faire communiquer des données pour les afficher sur un écran LCD pour nous assurer que la communication I2C fonctionne entre la carte et l'écran car la partie programme est moins conséquente pour un écran LCD et plus complexe sur un écran OLED



6) Communication CAN

Pour la communication entre la carte Vector présente dans les ordinateurs et la carte Cypress ainsi que logiciels CANalyzer et Psoc creator. Il nous faut créer un moyen de communication compatible entre les deux, pour cela nous avons mis en place un MCP2551 et des résistances qui permet la transmission sur les réseaux CAN.



7) Boîtier

Un boîtier en plastique est nécessaire pour la finition du testeur. Celui-ci doit être conçu en CAO à l'aide de DesingSpark Mechanical pour être imprimé à l'aide d'une imprimante 3d et incorporer tous les éléments nécessaires à son bon fonctionnement (batterie, PCB, écran, etc...)



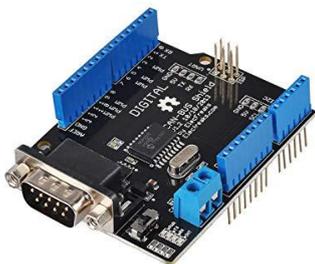
III) Réorganisation du projet

1) Contexte

Suite à de nombreux soucis liés à un manque de connaissance dans le domaine de la programmation, à la prise en main de PsoC creator et du manque d'informations concernant le logiciel sur internet, beaucoup de temps a été perdu pour l'avancement du projet. Suite à une entrevue avec notre tuteur de projet, un changement de stratégie a été mis en place dans le projet en remplaçant la carte Cypress par une carte Arduino Uno qui est plus simple à l'emploi, la programmation et plus d'informations disponible sur internet.

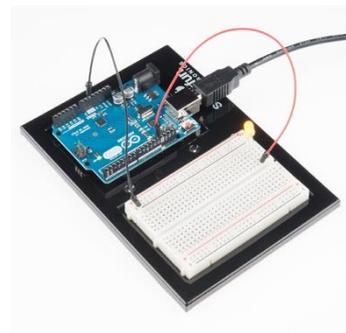
2) Présentation matérielle

Pour la suite de ce projet nous avons dû changer une partie du matériel utilisé initialement, en effet nous avons abandonné la carte Cypress Semiconductor au profit d'une carte Arduino Uno avec un Shield CAN.



NOTE: CAN BUS shield comes unassembled, soldering is required!!

Shield CAN



Carte Arduino Uno (Sparkfun)



3) Présentation logicielle

3.1) Arduino

Arduino est un logiciel de programmation Java pour les cartes Arduino. Il permet de programmer divers programmes en langage C++ pour différent domaine grâce à des entrées et des sorties.

3.2) DesignSpark PCB

DesignSpark PCB est un logiciel de CAO électronique pour la conception des circuits imprimés. Ce logiciel a la particularité d'être gratuit, il est développé par RS Components. Comme tous les logiciels de CAO électronique on fait son schéma en premier puis on crée les liaisons et les soudures pour l'impression du PCB

4) Stratégie

Suite au changement de carte pour la programmation du testeur de sonde lambda, un changement de stratégie a eu lieu. L'acquisition des données analogique du potentiomètre se font sur la carte Arduino, la réception de la trame CAN et ses données se fait grâce au Shield CAN que l'on branche directement sur la carte Arduino Uno. Le Shield permet la réception de données du réseau CAN, il comprend un MCP 2551 comme ce que nous avons prévu avec la carte Cypress et un MCP 2561, le shield comprend un port DB9 qui permet de communiquer avec les 2 fils CAN H et CAN L. Les données réceptionnées sont sur 2 octets qui doivent être assembler et converti en variable réelle pour obtenir la tension souhaitée.



5) Communication

Nous nous sommes rendu compte que pour envoyer des données sur l'écran OLED il n'y avait pas de librairie I2C compatible avec celui-ci car il a un contrôleur de données SSD 1309 et les librairie I2C sont seulement compatible avec le SSD 1306, à cause de cela nous avons changé le moyen de communication, nous somme passer de l'I2C au SPI car il n'y avait que des librairies SPI compatibles avec le contrôleur SSD 1309 et donc le transfert de données entre la carte Arduino et l'écran OLED possible.

6) SPI

Nous devons communiquer entre la carte Arduino et notre écran OLED pour cela nous utilisons la communication en SPI car en I2C cela est impossible. Le SPI permet l'envoi de données vers notre écran et y afficher des informations. Il est un bus qui opère en full-duplex, il se compose d'un fil SCLK qui est l'horloge, MOSI est l'entrée esclave, MISO la sortie esclave et SS qui est actif à l'état bas.

IV) Conclusion

1) Enseignement tirés

Le projet nous à permis de découvrir les cartes Cypress Perform et tous son environnement tels que Psoc Creator.

De développer nos connaissances dans CANalyzer dans le domaine de la communication ainsi que le CAPL. Découvrir les moyens de communication tels que l'I2C et le SPI pour communiquer entre une carte et un écran et y envoyer des données et afficher des informations relatives à notre projet.

Le projet nous a permis de travailler en autonomie apprendre à gérer notre temps la façon de travailler en groupe et chacun de son côté et échanger les informations nécessaires pour continuer à avancer.



2) Résultats du projet

Nous avons perdu beaucoup de temps sur le début du projet dû au manque d'information pour la programmation sur la carte Cypress, également un manque de connaissance au niveau de la programmation du langage C du fait de sa découverte cette année. A cause de cela le projet n'a pas pu aboutir à sa finalité.

Nous avons réussi à communiquer en I2C avec l'écran LCD et simuler le signal d'une sonde lambda de façon analogique avec le potentiomètre. Et la communication CAN pour afficher diverse information sur l'écran LCD.

Il nous reste encore à envoyer les informations de la sonde sur l'écran OLED en SPI, rendre le boîtier autonome avec une batterie, réaliser un PCB pour remplacer la carte arduino et son shield, et imprimer un boîtier à l'aide d'une imprimante 3D pour y incorporer le tout.

3) Perspectives futures

Au vu de la non finalité du projet réaliser les travaux restant pour aboutir à sa fin.

Rendre le boîtier compatible avec les 2 types de sonde lambda car actuellement le projet a été conçu pour les sondes au zirconium et pas les sonde au titane.