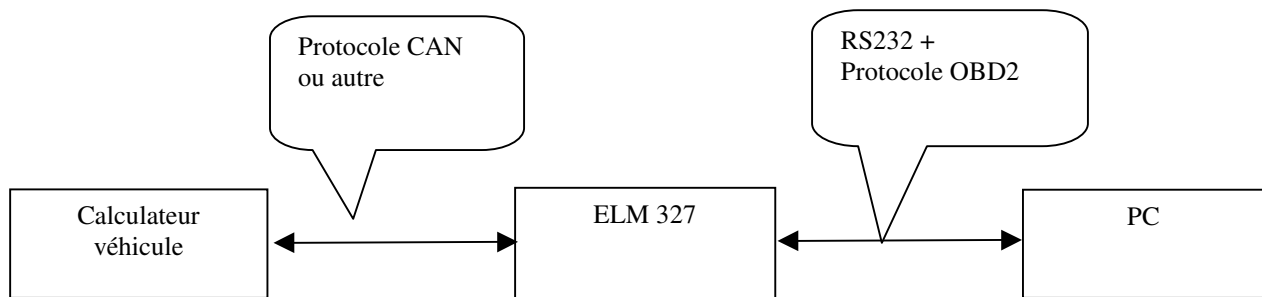


1. DESCRIPTION DU PROTOCOLE OBD2	1
1.1. 2 TYPES DE REQUÊTES OBD2 :.....	1
1.1.1. Requêtes de service et paramétrage de l'ELM 327.....	1
1.1.2. Requêtes d'interrogation du véhicule via des PID.....	2
1.2. DIALOGUE PC - ELM327	2
2. LA STRUCTURE DU LOGICIEL OBD2BV	2
2.1. TFORM1.....	2
2.2. TTHREADLISTEREQUETE	2
2.3. TREPONSE	3
2.4. MODE	3
2.5. PID	3
2.6. REG.....	3

1. Description du protocole OBD2



Le PC ne dialogue pas directement avec le calculateur du véhicule, mais transmet des requêtes à l'ELM327 qui se charge de traduire ces requêtes et les transmettre dans le protocole approprié au véhicule (SAE J1850, ISO 9141-2, CAN, etc.).

1.1.2 types de requêtes OBD2 :

1.1.1. Requêtes de service et paramétrage de l'ELM 327

Ces requêtes, de type « commandes AT » et ressemblant aux requêtes MODEM sont destinées à :

```

"ATZ\r",           // 0  Reset ALL
"ATRV\r",           // 1  Read the Voltage
"ATE0\r",           // 2  Echo OFF
"ATL0\r",           // 3  Line Feed OFF
"ATS0\r",           // 4  Space OFF
"ATSP0\r",          // 5  Set Protocol 0
"ATH1\r",           // 6  Set Headers ON
"ATDPN\r",          // 7  Describe the current protocol by nb
"0100\r",           // 8  Mode 1 PID 0 : Liste PID
"ATH0\r"            // 9  Set Headers OFF
  
```

1.1.2. Requêtes d'interrogation du véhicule via des PID

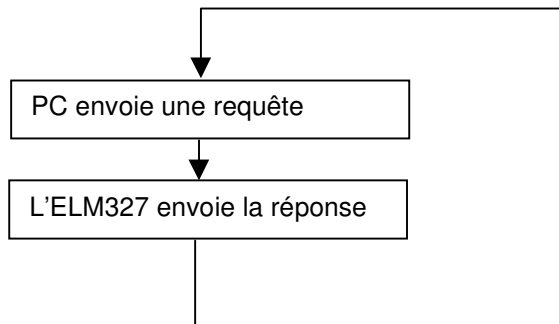
Celles ci sont destinées à être transmises au calculateur du véhicule pour effectuer le diagnostic et le reset des défauts :

Ces requêtes sont composées généralement d'un nombre 16 bits comprenant

- le mode (1 à 10) sur 1 octet
- le PID (1 à 255) sur 1 octet

La liste complète de ces requêtes se trouve dans le fichier **PidListCode.h**

1.2. Dialogue PC - ELM327



2. La structure du Logiciel OBD2bv

Répartie en Classes :

2.1. TForm1

C'est la classe du dialogue homme-machine

2.2. TthreadListeRequete

Cette classe gère les listes questions / réponses à poser à l'ELM.

Cette gestion est faite dans un thread pour ne pas bloquer l'application principale (Dialogue Homme Machine) si l'ELM tarde à répondre.

En fait, cette classe appelle des fonctions appartenant à Tform1 pour le traitement:

```

void __fastcall TthreadListeRequete::Execute()    // Corps du Thread
{
    switch (typeListe) {
        case 1: Form1->LanceListeReqInit(); break;
        case 2: Form1->LanceListeReqPid(); break;    // lister les PID valides
        case 3: Form1->LanceListeReqDiag1(); break;
        case 4: Form1->LanceListeReqPid2(); break;    // exécuter les requêtes PID
        case 9: Form1->LanceListeReqM9(); break;
    }
}
  
```



2.3. TREPONSE

classe qui analyse les réponses à une requête

2.4. MODE

Cette classe maintien à jour la table des modes -1:non testé 0:invalidé 1:Valide

2.5. PID

Cette classe maintien à jour la liste des PID valides sous forme d'un tableau.

2.6. REG

Classe destinée à lister les port COM du PC

Le plus gros du code se trouve en fait dans la classe Tform1