

## Conclusions

---

Les conclusions que es poden treure són vàries, però bàsicament se'n poden diferenciar dos grans blocs, les conclusions extretes de la part teòrica del treball i les pròpies de la realització de la part pràctica.

La incorporació de cada cop més sistemes electrònics no és una casualitat, ni de bon troç, és el producte de les pressions provocades per la legislació ambiental, per la competitivitat, i per les ganes de dominar un mercat tant important. En paral·lel els semiconductors han tingut una magnífica evolució, són barats i permeten obtenir millors rendiments, augmentar la seguretat, i en conjunt optimitzar el funcionament del vehicle. Com sempre quan es treuen els millors resultats és davant de pressions exteriors i competitivitat, aquesta pressió de la legislació ambiental no disminueix, al contrari, cada cop surten lleis més estrictes. La competència és dura, i els que volen sobreviure han d'evolucionar utilitzant les noves tecnologies, i tornem a estar parlant d'electrònica, a la qual se li pot preveure un futur esplendorós al igual que en altres sectors. L'electrònica permet obtenir el millor rendiment d'un sistema, permet que tot el conjunt del vehicle adquireixi un nivell d'integració molt elevat, on la interrelació és molt gran, tots els sistemes depenen de tots.

Amb l'augment de la integració de tots els sistemes tenim moltes millores, però també s'ha de procurar que el que poden ser punts en contra no ho siguin, d'aquests punts en contra cal destacar l'augment de complexitat del sistema. Aquest augment s'ha de produir d'una manera organitzada, i preveure qualsevol error o mal funcionament. És important que s'introdueixin funcions de supervisió que garanteixin el bon funcionament i la seguretat de les persones, però també han de garantir que quan el cotxe s'hagi de reparar ho pugui fer d'una manera ràpida i econòmica. Si ens fixem en les noves tendències i tecnologies, és bastant lògic pensar que els sistemes de diagnòstic tindran un paper molt important. Els diagnòstics encara han d'avançar molt per poder situar-se al nivell de nous sistemes com l'X-by-Wire. Recordem aquests sistemes i fem-nos una pregunta: "Si s'espatllen qui el arreglarà?", la resposta penso que és fàcil, sense el sistema de diagnòstic adequat potser ni els propis enginyers. Per tant el sistema de diagnòstic ha de ser una part del disseny a tenir molt en compte, s'han d'intentar coordinar tots els sistemes per obtenir en la seva globalitat els diagnòstics correctes.

Una coordinació a tant al nivell, amb la gran quantitat d'empreses implicades, possiblement només es pot aconseguir si hi ha un estandardització. Cal que estàndards que puguin sortir ho facin amb la suficient planificació i de mà de consorcis que integrin les empreses involucrades. Els estàndards poden ser molt beneficiosos, fins i tot imprescindibles, però si n'apareixen masses no sen treu el profit que n'hauríem de treure. Aquests fets, amb el nivells de tècnica i evolució actuals no es poden permetre.

També hi ha un altre aspecte a tenir en compte envers l'estandardització, no s'ha d'oblidar la recerca en altres tecnologies, que tot i que potser no es contemplin en l'estàndard poden ser millors. Seria un gran error dedicar-se a desenvolupar només l'estàndard i les tecnologies que l'integraran, amb el temps de desenvolupament d'aquest, que és de uns anys, ja poden haver aparegut tecnologies molt superiors. Des d'aquest punt de vista podríem concloure dient que un estàndard hauria de ser obert i tenir la suficient flexibilitat per poder absorbir els productes de noves investigacions.

Respecte la part pràctica, podria buscar motius pel fet que no s'hagi arribat a aconseguir un resultat que en un principi, personalment, hagués considerat l'ideal. Però la veritat, aquesta possibilitat ja es va assumir de bon començament perquè no era un treball planificat, aquest s'aniria definint segons les possibilitats tècniques que tinguéssim. Resulta evident que treballar amb un vehicle, que havia de ser la base de les proves, no és tant accessible com per exemple un ordinador, on s'hi poden realitzar moltes proves i modificacions amb tota seguretat.

El vehicle amb el qual hem realitzat diverses proves, és el del mateix tutor del projecte, que el va cedir molt amablement ja assumint algun risc. Al llarg de la realització del projecte ens varem donar compte que amb segons quin tipus de proves podíem comprometre la integritat del vehicle, motiu pel qual varem dedicar esforços a altres aspectes. La intenció va ser elaborar un treball teòric complet, tractant molts aspectes i molt diversos, sense oblidar la planificació de la part incompleta del treball. D'aquesta manera, si algun alumne disposa dels recursos suficients per continuar aquest treball podrà tenir tota la informació recollida, i aprofitar-la per començar des d'un nivell més alt i progressar més.

Al estudiar una indústria tant important com és la de l'automòbil, t'adones que hi ha uns punts que són claus per al desenvolupament tecnològic. Es necessària la competitivitat, el treball en equip a totes les escales, i per davant de tot l'organització i innovació. Sobretot és bo tenir-ho present en un país com el nostre que està notant els



efectes de la falta de innovació i visió de futur. Crec que és important sortir al mercat laboral amb unes idees clares i així intentar aplicar-les, al nivell que sigui, clar.

Ja per acabar, només voldria dir que malgrat que el treball no s'ajusti a la idealitat si que s'ajusta a les possibilitats que hem tingut. I de ben segur que els resultats que n'he pogut extreure em seran de gran ajuda, resultat que hem deixa totalment satisfet.

# Annex A

---

- I. Codi del programa d'escolta
- II. Resultats de l'escolta
- III. Codi del programa d'escolta millorat

En la creació d'ambdós programes d'escolta s'ha utilitzat el CVI/Labwindows, per a cadascun dels programes s'adjunta un fitxer .c i la seva corresponent llibreria .h. El primer programa com ja s'ha comentat en el treball no esta gens optimitzat, el principal objectiu era que realitzes la seva funció sense perdre temps en la seva programació, per aquest motiu l'estil no és gaire elegant.

## I. Codi del programa d'escolta

### Fitxer .c

```

#include <formatio.h>
#include <ansi_c.h>
#include <utility.h>
#include <rs232.h>
#include <cvirte.h>
#include <userint.h>
#include "rs232obd.h"

static int panelHandle, configura;
int
    comport,
    baudrate,
    portindex,
    parity,
    databits,
    stopbits,
    inputq,
    outputq,
    nbytes,
    RS232Error,
    nCar,
    numero,
    i;

unsigned char *a,*buff,*cadenar,b;

int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((panelHandle = LoadPanel (0, "rs232obd.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (panelHandle);
    configura=LoadPanel(0,"rs232obd.uir",CONFIGURA);
    RunUserInterface ();
    DiscardPanel (panelHandle);
    DiscardPanel (configura);
    return 0;
}

int CVICALLBACK SCONF1 (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int Port;
    switch (event)
    {
        case EVENT_COMMIT:

            HidePanel(configura);
            DisplayPanel(panelHandle);

            GetCtrlVal(configura,CONFIGURA_COMPORT,&comport);
            GetCtrlVal(configura,CONFIGURA_BITSEGON,&baudrate);
            GetCtrlVal(configura,CONFIGURA_CUAENTRADA,&inputq);
            GetCtrlVal(configura,CONFIGURA_CUASORTIDA,&outputq);
            GetCtrlVal(configura,CONFIGURA_BITSDADES,&databits);
            GetCtrlVal(configura,CONFIGURA_BITSSTOP,&stopbits);

```



```

        GetCtrlVal(configura,CONFIGURA_PARITAT,&parity);
        RS232Error= OpenComConfig (1/*comport*/, "", baudrate, parity, databits,
        stopbits, inputq, outputq);
        RS232Error= OpenComConfig (2/*comport*/, "", baudrate, parity, databits,
        stopbits, inputq, outputq);

        //RE-CONFIGURACIÓ DEL COM (10,4Kbauds o 5bauds)

        Port=0x3f8;
        outp ( Port+3, inp( Port+3 ) | 0x80 );
        outpw( Port, numero );
        outp ( Port+3, inp( Port+3 ) & 0x7F );
        Port=0x2f8;
        outp ( Port+3, inp( Port+3 ) | 0x80 );
        outpw( Port, numero );
        outp ( Port+3, inp( Port+3 ) & 0x7F );

        break;
    }
    return 0;
}

int CVICALLBACK CONFIG2 (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            HidePanel(panelHandle);
            DisplayPanel(configura);
            numero=0x5A00;

            break;
    }
    return 0;
}

int CVICALLBACK CONFIG1 (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

            HidePanel(panelHandle);
            DisplayPanel(configura);
            numero=0x000B;

            break;
    }
    return 0;
}

int CVICALLBACK SORTIR (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
    }
    return 0;
}

```



```

int index = 0;
int CVICALLBACK REBRE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

            nCar=GetInQLen (1/*comport*/);
            cadenar=calloc(nCar,sizeof(char));
            buff=calloc(nCar*3+4,sizeof(char));
            nCar=ComRd (1/*comport*/, cadenar, nCar);
            buff[0] = '1';
            buff[1] = ':';
            buff[2] = '\0';
            for (i=0; i<nCar;i++){
                sprintf( &buff[i*3+3], "%02x ", cadenar[i] );
            }
            InsertTextBoxLine (panelHandle, PANEL_textbox, index++, buff);
            free(cadenar);
            free(buff);

            nCar=GetInQLen (2/*comport*/);
            cadenar=calloc(nCar,sizeof(char));
            buff=calloc(nCar*3+4,sizeof(char));
            nCar=ComRd (2/*comport*/, cadenar, nCar);
            buff[0] = '2';
            buff[1] = ':';
            buff[2] = '\0';
            for (i=0; i<nCar;i++){
                sprintf( &buff[i*3+3], "%02x ", cadenar[i] );
            }
            InsertTextBoxLine (panelHandle, PANEL_textbox, index++, buff);
            free(cadenar);
            free(buff);

            break;

        }
    }
    return 0;
}

```

## Fitxer .h

```

/*****
/* LabWindows/CVI User Interface Resource (UIR) Include File          */
/* Copyright (c) National Instruments 2004. All Rights Reserved.      */
/*                                                                    */
/* WARNING: Do not add to, delete from, or otherwise modify the contents */
/*   of this include file.                                           */
*****/

#include <userint.h>

#ifdef __cplusplus
    extern "C" {
#endif

    /* Panels and Controls: */

#define CONFIGURA          1

```



```

#define CONFIGURA_BITSSTOP          2
#define CONFIGURA_BITSDADES        3
#define CONFIGURA_PARITAT          4
#define CONFIGURA_SCONFI            5 /* callback function: SCONFI */
#define CONFIGURA_BITSSECON        6
#define CONFIGURA_COMPORT          7
#define CONFIGURA_CUAENTRADA        8
#define CONFIGURA_CUASORTIDA        9

#define PANEL                        2
#define PANEL_textbox                2
#define PANEL_CONFIG2                3 /* callback function: CONFIG2 */
#define PANEL_CONFIG1                4 /* callback function: CONFIG1 */
#define PANEL_SORTIR                  5 /* callback function: SORTIR */
#define PANEL_REBRE                    6 /* callback function: REBRE */

/* Menu Bars, Menus, and Menu Items: */

/* (no menu bars in the resource file) */

/* Callback Prototypes: */

int CVICALLBACK CONFIG1(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK CONFIG2(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK REBRE(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK SCONFI(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK SORTIR(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);

#ifdef __cplusplus
}
#endif

```



## II. Resultats de l'escolta

### Obd-2 function

1: fc 0d 0d 81 94 00 00 00 00

2: 00 ff 0d 0d c1 00 55 08 08

### Test

1: fc 0d 0d 81 b4 00

2: 00 ff 0d 0d c1 00

### Autoscan1

1: fc 0d 0d 81 b4 00 00 00 00 00 f5 c4 fe 09 8f 8d 87 86 8f 89 8f 8d 8e 7b 79 bf 8e b1 86 33 bf 6d 8b 9f 7a  
7b 7c 9f 03 02 09 03 f8 fc 09 bf bf 8f 8f 03 04 09 03 f4 fa 09 4c 78 bf bf 8f 8f 8e 03 06 09 03 f8 f8 09 ff  
fb ff b1 03 08 09 03 fc f6 f6 03 0a 09 03 fc f4 f6 03 0c 09 03 fc f2 f6 03 0e 09 03 fc f0 f6 03 30 07 03 f9 ce  
03 00 00 f7 03 12 09 03 fc cc f6 03 34 09 03 fc ea f6 03 36 06 03 fc 0d 0d 81 94 00 00 00 00 00 00 00

2: 00 ff 0d 0d c1 00 f5 01 8a d5 01 8a f5 01 8a d5 01 8a f5 3b e4 01 fe f6 09 70 cf 72 cd 78 c7 79 c6 70 cf  
76 c9 70 cf 72 cd 71 ce c4 fb c6 f9 60 df 71 ce 6e d1 79 c6 ec 93 60 df f2 ed 74 cb 60 df c5 fa c4 fb c3 fc  
60 ff 03 03 fc 02 fd 09 f6 03 07 f8 03 fc f6 09 60 df 60 df 70 cf 70 cf 03 03 fc 04 fb 09 f6 03 0b f4 05 fa f6  
09 f3 ec c7 f8 60 df 60 df 70 cf 78 c7 70 cf 71 ce 03 03 fc 06 f9 09 f6 03 07 f8 07 f8 f6 09 00 ff 04 fb 00 ff  
6e d1 03 03 fc 08 f7 09 f6 03 03 fc 09 f6 09 f6 03 03 fc 0a f5 09 f6 03 03 fc 0b f4 09 f6 03 03 fc 0c f3 09 f6  
03 03 fc 0d f2 09 f6 03 03 fc 0e f1 09 f6 03 03 fc 0f f0 09 f6 03 03 fc 10 ef 07 f8 03 06 f9 31 ee fc 03 ff 00 ff  
00 88 f7 03 03 fc 12 ed 09 f6 03 03 fc 33 ec 09 f6 03 03 fc 34 eb 09 f6 03 03 fc 35 ea 09 f6 03 03 fc 16 e9  
06 f9 03 00 ff 0d 0d c1 00

1: 00 00 fc 0d 0d 81 94 00

2: 00 ff 0d 0d c1 00

1: 00 00 00 00 00 00 00 00 fc 0d 0d 81 94 00 00 00

2: 00 ff 0d 0d c1 00

1: 00 00

2:

1: fc 0d 0d 81 b4 00 00 00 00 00 00 00

2: 00 ff 0d 0d c1 00

1: 00 00 00 00 00 00 fc 0d 0d 81 b4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 f5 f0 fe 09 8e 77 8f 86 8a 8c  
8d 8a 88 bf 7d bf 03 02 09 03 f0 fc 09 76 72 72 70 9f 69 68 65 8c 45 8f 49 03 04 09 03 f1 fa 09 8c 8b 86  
8c 8f 8b 88 bf 69 8f 8f 03 06 09 03 f7 f8 09 ff 74 ff ff b1 03 08 09 03 fc f6 f6 03 0a 09 03 fc f4 f6 03 0c 09  
03 fc f2 f6 03 0e 09 03 fc f0 f6 03 30 07 03 f9 ee 03 00 00 f7 03 12 09 03 fc ec f6 03 34 09 03 fc ea f6 03  
36 06 03 fc 0d 0d 81 b4

2: 00 ff 0d 0d c1 00 f5 01 8a d5 01 8a f5 01 8a f5 01 8a f5 0f f0 01 fe f6 09 71 ce c8 f7 70 cf 79 c6 75 ca 73  
cc 72 cd 75 ca 77 c8 60 df c2 fd 60 ff 03 03 fc 02 fd 09 f6 03 0f f0 03 fc f6 09 c9 f6 cd f2 cd f2 cf f0 60 ff d6  
e9 f7 e8 fa e5 73 cc da e5 70 cf f6 e9 03 03 fc 04 fb 09 f6 03 0e f1 05 fa f6 09 73 cc 74 cb 79 c6 73 cc 70  
cf 74 cb 77 c8 60 ff f6 e9 70 cf 70 cf 03 03 fc 06 f9 09 f6 03 08 f7 07 f8 f6 09 00 ff cb f4 00 ff 00 ff 6e f1 03  
03 fc 08 f7 09 f6 03 03 fc 09 f6 09 f6 03 03 fc 0a f5 09 f6 03 03 fc 0b f4 09 f6 03 03 fc 0c f3 09 f6 03 03 fc  
0d f2 09 f6 03 03 fc 0e f1 09 f6 03 03 fc 0f f0 09 f6 03 03 fc 30 ef 07 f8 03 06 f9 31 ee fc 03 ff 00 ff 00 88 f7



03 03 fc 32 ed 09 f6 03 03 fc 33 ec 09 f6 03 03 fc 14 eb 09 f6 03 03 fc 35 ea 09 f6 03 03 fc 36 e9 06 f9 03  
00 ff 0d 0d c1

1: 00 00 00 00 00 00 00 00 00 00

2: 00

#### Autoscan2

1: fc 0d 0d 81 b4 00 00 00 00 00 f5 c4 fe 09 8f 8d 87 86 8f 89 8f 8d 8e 7b 79 bf 8e 91 86 13 9f 6d 8b 9f 7a  
7b 7c bf 03 02 09 03 f8 fc 09 bf bf 8f 8f 03 04 09 03 f4 fa 09 6c 78 9f bf 8f 87 8f 8e 03 06 09 03 f8 f8 09 ff  
fb ff 91 03 08 09 03 fc f6 f6 03 0a 09 03 fc f4 f6 03 0c 09 03 fc f2 f6 03 0e 09 03 fc f0 f6 03 30 07 03 f9 ee  
03 00 00 f7 03 12 09 03 fc ec f6 03 34 09 03 fc ea f6 03 36 06 03 fc

2: 00 ff 0d 0d c1 00 d5 01 8a d5 01 8a f5 01 8a d5 01 8a f5 3b e4 01 fe f6 09 70 cf 72 cd 78 c7 79 c6 70 cf  
76 c9 70 cf 72 cd 71 ce c4 fb c6 f9 60 ff 71 ce 6e f1 79 c6 ec b3 60 ff f2 ed 74 cb 60 ff c5 fa c4 fb c3 fc 60  
ff 03 03 fc 02 fd 09 f6 03 07 f8 03 fc f6 09 60 ff 60 ff 70 cf 70 cf 03 03 fc 04 fb 09 f6 03 0b f4 05 fa f6 09 d3  
ec c7 f8 60 ff 60 df 70 cf 78 c7 70 cf 71 ce 03 03 fc 06 f9 09 f6 03 07 f8 07 f8 f6 09 00 ff 04 fb 00 ff 6e f1 03  
03 fc 08 f7 09 f6 03 03 fc 09 f6 09 f6 03 03 fc 0a f5 09 f6 03 03 fc 0b f4 09 f6 03 03 fc 0c f3 09 f6 03 03 fc  
0d f2 09 f6 03 03 fc 0e f1 09 f6 03 03 fc 0f f0 09 f6 03 03 fc 10 ef 07 f8 03 06 f9 31 ee fc 03 ff 00 ff 00 88 f7  
03 03 fc 32 ed 09 f6 03 03 fc 33 ec 09 f6 03 03 fc 14 eb 09 f6 03 03 fc 35 ea 09 f6 03 03 fc 16 e9 06 f9 03  
00 ff

1: 00 00 00 00 00 00 00 00 00 00 fc 0d 0d

2: 00 00 ff 0d 0d

1: 81 b4 00 00 00 00 00 00 00 00

2: c1 00 00

1: fc 0d 0d 81 b4 00 00 00 00 00 00 00 00 00 00 fc 0d 0d 81 b4 00 00 00 00 00 00 00 00 00 00 00 00 00 00

2: ff 0d 0d c1 00 00 ff 0d 0d c1 00 00

1: fc 0d 0d 81 b4 00 00 00 00 00 00 00 00 00 00 f5 f0 fe 09 8e 77 8f 86 8a 8c 8d 8a 88 bf 7d bf 03 02 09 03 f0  
fc 09 76 72 72 70 bf 49 68 65 8c 65 8f 49 03 04 09 03 f1 fa 09 8c 8b 86 8c 8f 8b 88 9f 69 8f 8f 03 06 09 03  
f7 f8 09 ff 74 ff ff b1 03 08 09 03 fc f6 f6 03 0a 09 03 fc f4 f6 03 0c 09 03 fc f2 f6 03 0e 09 03 fc f0 f6 03 10  
07 03 f9 ce 03 00 00 f7 03 32 09 03 fc ec f6 03 34 09 03 fc ca f6 03 16 06 03 fc 0d 0d 81

2: ff 0d 0d c1 00 d5 01 8a d5 01 8a d5 01 8a f5 0f f0 01 fe f6 09 71 ce c8 f7 70 cf 79 c6 75 ca 73 cc 72 cd  
75 ca 77 c8 60 ff c2 fd 60 ff 03 03 fc 02 fd 09 f6 03 0f f0 03 fc f6 09 c9 f6 cd f2 cd f2 cf f0 60 ff f6 e9 f7 e8  
fa e5 73 cc fa e5 70 cf f6 e9 03 03 fc 04 fb 09 f6 03 0e f1 05 fa f6 09 73 cc 74 cb 79 c6 73 cc 70 cf 74 cb  
77 c8 60 ff f6 e9 70 cf 70 cf 03 03 fc 06 f9 09 f6 03 08 f7 07 f8 f6 09 00 ff cb f4 00 ff 00 ff 6e f1 03 03 fc 08  
f7 09 f6 03 03 fc 09 f6 09 f6 03 03 fc 0a f5 09 f6 03 03 fc 0b f4 09 f6 03 03 fc 0c f3 09 f6 03 03 fc 0d f2 09  
f6 03 03 fc 0e f1 09 f6 03 03 fc 0f f0 09 f6 03 03 fc 30 ef 07 f8 03 06 f9 31 ee fc 03 ff 00 ff 00 88 f7 03 03 fc  
32 ed 09 f6 03 03 fc 33 ec 09 f6 03 03 fc 34 eb 09 f6 03 03 fc 35 ea 09 f6 03 03 fc 36 e9 06 f9 03 00 ff 0d  
0d c1

1: b4 00 00 00 00 00 00 00 00 00 00 00 00 00

2: 00

#### Obd-2 function

1: fc 0d 0d 81 b4 00 00 00 00

2: 00 ff 0d 0d c1 00 55 08 08



## Autoscan

1: fc 0d 0d 81 b4 00 00 00 00 00 f5 e4 fe 09 8f 8d 87 86 8f 89 8f 8d 8e 7b 79 bf 8e b1 86 33 bf 6d 8b bf 7a 7b 7c bf 03 02 09 03 f8 fc 09 9f 9f 8f 8f 03 04 09 03 f4 fa 09 6c 78 9f 9f 8f 87 8f 8e 03 06 09 03 f8 f8 09 ff fb ff b1 03 08 09 03 fc f6 f6 03 0a 09 03 fc f4 f6 03 0c 09 03 fc f2 f6 03 0e 07 03 f9 f0 03 00 00 f7 03 30 09 03 fc ce f6 03 32 09 03 fc cc f6 03 14 06 03 fc 0d 0d 81 b4

2: 00 ff 0d 0d c1 00 f5 01 8a f5 01 8a f5 01 8a d5 01 8a f5 3b e4 01 fe f6 09 70 cf 72 cd 78 c7 79 c6 70 cf 76 c9 70 cf 72 cd 71 ce c4 fb c6 f9 60 df 71 ce 6e d1 79 c6 ec 93 60 df f2 ed 74 cb 60 ff c5 fa c4 fb c3 fc 60 ff 03 03 fc 02 fd 09 f6 03 07 f8 03 fc f6 09 60 ff 60 ff 70 cf 70 cf 03 03 fc 04 fb 09 f6 03 0b f4 05 fa f6 09 f3 ec c7 f8 60 ff 60 ff 70 cf 78 c7 70 cf 71 ce 03 03 fc 06 f9 09 f6 03 07 f8 07 f8 f6 09 00 ff 04 fb 00 ff 6e d1 03 03 fc 08 f7 09 f6 03 03 fc 09 f6 09 f6 03 03 fc 0a f5 09 f6 03 03 fc 0b f4 09 f6 03 03 fc 0c f3 09 f6 03 03 fc 0d f2 09 f6 03 03 fc 0e f1 07 f8 03 06 f9 0f f0 fc 03 ff 00 ff 00 88 f7 03 03 fc 30 ef 09 f6 03 03 fc 31 ee 09 f6 03 03 fc 32 ed 09 f6 03 03 fc 33 ec 09 f6 03 03 fc 34 eb 06 f9 03 00 ff 0d 0d c1

1: 00 00 00 00 00 00 00 00 00 fc

2: 00 00 ff

1: 0d 0d 81 94 00 00 00 00 00 00 00 00 00 00 fc 0d 0d 81

2: 0d 0d c1 00 00 ff 0d 0d c1

1: b4 00 00 00 00 00 00 00 00 00 00 fc 0d 0d 81 b4 00 00 00 00 00 00 00 00 00 f5 f0 fe 09 8e 77 8f 86 8a 8c 8d 8a 88 bf 7d bf 03 02 09 03 f0 fc 09 76 72 72 70 bf 69 68 45 8c 65 8f 49 03 04 09 03 f1 fa 09 8c 8b 86 8c 8f 8b 88 bf 49 8f 8f 03 06 09 03 f7 f8 09 ff 74 ff ff 91 03 08 09 03 fc f6 f6 03 0a 09 03 fc f4 f6 03 0c 09 03 fc f2 f6 03 0e 07 03 f9 f0 03 00 00 f7 03 30 09 03 fc ee f6 03 32 09 03 fc cc f6 03 34 06 03 fc 0d 0d 81 94 00 00 00 00 00 00 00

2: 00 00 ff 0d 0d c1 00 f5 01 8a f5 01 8a f5 01 8a f5 0f f0 01 fe f6 09 71 ce c8 f7 70 cf 79 c6 75 ca 73 cc 72 cd 75 ca 77 c8 60 df c2 fd 60 df 03 03 fc 02 fd 09 f6 03 0f f0 03 fc f6 09 c9 f6 cd f2 cd f2 cf f0 60 df f6 e9 d7 e8 da e5 73 cc fa e5 70 cf d6 e9 03 03 fc 04 fb 09 f6 03 0e f1 05 fa f6 09 73 cc 74 cb 79 c6 73 cc 70 cf 74 cb 77 c8 60 df f6 e9 70 cf 70 cf 03 03 fc 06 f9 09 f6 03 08 f7 07 f8 f6 09 00 ff cb f4 00 ff 00 ff 6e f1 03 03 fc 08 f7 09 f6 03 03 fc 09 f6 09 f6 03 03 fc 0a f5 09 f6 03 03 fc 0b f4 09 f6 03 03 fc 0c f3 09 f6 03 03 fc 0d f2 09 f6 03 03 fc 0e f1 07 f8 03 06 f9 0f f0 fc 03 ff 00 ff 00 88 f7 03 03 fc 30 ef 09 f6 03 03 fc 31 ee 09 f6 03 03 fc 32 ed 09 f6 03 03 fc 33 ec 09 f6 03 03 fc 34 eb 06 f9 03 00 ff 0d 0d c1 00

1: 00 00 00 00 00 00

### III. Codi del programa d'escolta millorat

#### Fitxer .c

```

/*-----*/
/*
/* PROGRAMA: Programa d'escolta per a ISO-9142
/*
/* PROPÒSIT: La intenció d'aquest programa és captar la comunicació, entre
/*           un segon PC i el vehicle, no recomanable per a velocitats
/*           superiors a 10,4Kbauds
/*-----*/

```

```

/*-----*/
/* Llibreries necessàries
/*-----*/

```

```

#include <formatio.h>
#include <ansi_c.h>
#include <utility.h>
#include <rs232.h>
#include <cvirte.h>
#include <userint.h>
#include "rs232obd12.h"

```

```

/*-----*/
/*     Variables globals
/*-----*/

```

```

static char proj_dir[MAX_PATHNAME_LEN];
static char file_name[MAX_PATHNAME_LEN];

```

```

static int          panelHandle,
                  configura;

```

```

int                baudrate,
                  parity,
                  databits,
                  stopbits,
                  inputq,
                  outputq,
                  nbytes,
                  RS232Error,
                  Tadqui;

```

```

unsigned char      buff[3000];

```

```

/*-----*/
/*     Prototips de les funcions
/*-----*/

```

```

void rebre_fins_temps(unsigned int Tadqui);
void rebre_fins_tecla(void);
void taula_a_textbox(unsigned char *buff);

```

```

/*-----*/
/*      Punt d'entrada de l'aplicació      */
/*-----*/

int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1;          /* out of memory */
    if ((panelHandle = LoadPanel (0, "rs232obd12.uir", PANEL)) < 0)
        return -1;
    if((configura = LoadPanel(0,"rs232obd12.uir",CONFIGURA)) < 0)
        return -1;
    GetProjectDir (proj_dir);
    DisplayPanel (panelHandle);
    RunUserInterface ();
    CloseCom(1);
    CloseCom(2);
    DiscardPanel (panelHandle);
    DiscardPanel (configura);
    return 0;
}

/*-----*/
/*      Botó rebre: Aquest botó esta inhabilitat fins que no s'han configurat      */
/*      el COM1 i el COM2, per realitzar la configuració s'ha                      */
/*      d'entrar al menu en qüestió .Al prémer el botó rebre es quan              */
/*      comencem a guardar les dades que al aturar-se, ja sigui per              */
/*      la fi de temps o per prémer una tecla, es mostraran per                  */
/*      pantalla                                                                    */
/*-----*/

int CVICALLBACK REBRE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    unsigned int Tadqui;
    unsigned int Tipusaturada;

    switch (event)
    {
        case EVENT_COMMIT:

            //netegem el textbox i inicialitzem
            ResetTextBox (panelHandle, PANEL_TEXTBOX, "");

            //Apaguem el hardware handshaking, sino no funciona en loopback
            SetCTSMode(1, LWRS_HWHANDSHAKE_OFF);
            SetCTSMode(2, LWRS_HWHANDSHAKE_OFF);

            //buidem els buffers del COM1 i COM2
            FlushInQ(1);
            FlushOutQ(1);
            FlushInQ(2);
            FlushOutQ(2);

            //Time-out per la comanda de llegir el contingut del port sèrie
            SetComTime (1, 0.001);
            SetComTime (2, 0.001);

            ResetTextBox (panelHandle, PANEL_TEXTBOX_2, "");
            InsertTextBoxLine (panelHandle, PANEL_TEXTBOX_2,1, "Rebent...");
            InsertTextBoxLine (panelHandle, PANEL_TEXTBOX_2,4, "COM1 R:");
            InsertTextBoxLine (panelHandle, PANEL_TEXTBOX_2,5, "COM2 T:");
    }
}

```

```

//en funció del tipus de aturada seleccionada la funcio rebre es diferent
GetCtrlVal(panelHandle,PANEL_TADQUISICIO,&Tadqui);
GetCtrlVal(panelHandle,PANEL_TIPUSATURADA,&Tipusaturada);

if (Tipusaturada==1)
{
    rebre_fins_temps(Tadqui);
    taula_a_textbox (buff);
}
else if(Tipusaturada==0)
{
    rebre_fins_tecla();
    taula_a_textbox(buff);
}

ResetTextBox (panelHandle, PANEL_TEXTBOX_2, "");
InsertTextBoxLine (panelHandle, PANEL_TEXTBOX_2,4, "COM1 R:");
InsertTextBoxLine (panelHandle, PANEL_TEXTBOX_2,5, "COM2 T:");

break;

return 0;
}

/*-----*/
/* Botó configurar: Al premer configura l'únic que fem es amagar el */
/* panell normal i mostrar el de configuració */
/*-----*/

int CVICALLBACK CONFIG (int panel, int control, int event,
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

            HidePanel(panelHandle);
            DisplayPanel(configura);

            break;

        return 0;
    }
}

/*-----*/
/* Botó netejar: Al premer neteja borrem el contingut de text que hi */
/* ha en el panell on es presenten les adquisicions */
/* panell normal i mostrar el de configuració */
/*-----*/

int CVICALLBACK CLEAN (int panel, int control, int event,
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

            ResetTextBox (panelHandle, PANEL_TEXTBOX, "");

            break;

        return 0;
    }
}

```

```

/*-----*/
/*      Botó guardar:          Guardem el contingut de la pantalla en un fitxer      */
/*                               i netejem la pantalla.                               */
/*-----*/

int CVICALLBACK GUARDAR (int panel, int control, int event,
                        void *callbackData, int eventData1, int eventData2)
{
    unsigned char    bufftexttemp[150]="",
                    *buffdefinitiu,
                    *bufftext;

    int              ind=0,
                    index=0,
                    longitud=0,
                    i;

    switch (event)
    {
        case EVENT_COMMIT:

            GetNumTextBoxLines (panelHandle, PANEL_TEXTBOX, &index);
            if(index==0)
            {
                MessagePopup ("Atenció!", "La pantalla esta buida...");
                break;
            }
            GetTextBoxLineLength (panelHandle, PANEL_TEXTBOX, (index-1), &i);

            bufftext=calloc(84*(index+1),sizeof(char));
            for(ind=0;ind<index;ind++)
            {
                GetTextBoxLine (panelHandle, PANEL_TEXTBOX, ind, bufftexttemp);
                strcat (bufftext, bufftexttemp);
            }

            longitud=strlen (bufftext);
            buffdefinitiu=calloc(longitud+1,sizeof(char));
            strcpy (buffdefinitiu,bufftext);

            if (FileSelectPopup (proj_dir, "*.txt", "*.txt", "Nom de l'arxiu a guardar",
            VAL_OK_BUTTON, 0, 1, 1,0, file_name) > 0)
            {
                ArrayToFile (file_name, buffdefinitiu,VAL_UNSIGNED_CHAR,longitud, 1,
                VAL_GROUPS_TOGETHER, VAL_GROUPS_AS_ROWS,VAL_CONST_WIDTH, 10,
                VAL_ASCII, VAL_TRUNCATE);
            }

            free(bufftext);
            free(buffdefinitiu);
            break;
    }
    return 0;
}

```

```

/*----- */
/*      Botó carregar:      Carreguem un fitxer que hem guardat anteriorment      */
/*      mitjançant la funció guardar d'aquest programa.      */
/*----- */

```

```

int CVICALLBACK CARREGAR (int panel, int control, int event,
                          void *callbackData, int eventData1, int eventData2)
{
    long          tamany;
    unsigned char bufftemp[97]="",
                  *bufftext;
    int           in=0,
                  a=0,
                  index=0,
                  longitud=0;

    switch (event)
    {
        case EVENT_COMMIT:
            if (FileSelectPopup (proj_dir, "*.txt", "*.txt", "Nom de l'arxiu a llegir",VAL_OK_BUTTON, 0, 1, 1, 0, file_name) > 0)
            {
                GetFileInfo (file_name,&tamany);;
                bufftext=calloc((int)tamany,sizeof(char));
                FileToArray(file_name,bufftext,
                VAL_UNSIGNED_CHAR,(int)tamany,1,VAL_GROUPS_TOGETHER, VAL_GROUPS_AS_ROWS,
                VAL_ASCII);

                ResetTextBox (panelHandle, PANEL_TEXTBOX, "");
                longitud=strlen (bufftext);
                while(a<longitud){
                    for(in=0;in<=(23*4+3);in++)
                    {
                        bufftemp[in]=bufftext[a];
                        a++;
                    }
                    InsertTextBoxLine(panelHandle,
                PANEL_TEXTBOX,index,bufftemp);
                    index++;
                }
                free (bufftext);
            }

        break;
    }
    return 0;
}

```

```

/*----- */
/*      Botó Sortir(configuració): Al sortir de la configuració es quan      */
/*      pròpiament realitzem la configuració      */
/*      segons el valors elegits per l'usuari.      */
/*----- */

```

```

int CVICALLBACK SCONF (int panel, int control, int event,
                       void *callbackData, int eventData1, int eventData2)
{
    int N;
    float baudrateReal;

    char buf[30];

```





```

switch (event)
{
    case EVENT_COMMIT:

        HidePanel(configura);
        DisplayPanel(panelHandle);
        SetCtrlAttribute (panelHandle, PANEL_TEST, ATTR_DIMMED, 0);
        GetCtrlVal(configura,CONFIGURA_BITSSEGON,&baudrate);
        GetCtrlVal(configura,CONFIGURA_CUAENTRADA,&inputq);
        GetCtrlVal(configura,CONFIGURA_CUASORTIDA,&outputq);
        GetCtrlVal(configura,CONFIGURA_BITSDADES,&databits);
        GetCtrlVal(configura,CONFIGURA_BITSSSTOP,&stopbits);
        GetCtrlVal(configura,CONFIGURA_PARITAT,&parity);

        //Aproximem el valor obtingut a baudrate al més pròxim que podem
        obtenir

        N=RoundRealToNearestInteger(((1843200)/(16*baudrate)));
        baudrateReal=(1843200)/(float)(N*16);
        Fmt (buf, "%s<%f[p2]", baudrateReal);
        strcat (buf, " bauds");
        MessagePopup ("Atenció! BaudRate Real",buf);

        // Programem el COM1 i el COM2
        RS232Error= OpenComConfig (1/*comport*/, "", baudrate, parity,
databits,stopbits, inputq, outputq);
        RS232Error= OpenComConfig (2/*comport*/, "", baudrate, parity,
databits,stopbits, inputq, outputq);

        //SetCtrlAttribute (panelHandle, PANEL_ATURAR, ATTR_DIMMED,
0);

        SetCtrlAttribute (panelHandle, PANEL_REBRE, ATTR_DIMMED, 0);
        ResetTextBox (panelHandle, PANEL_TEXTBOX_2, "");
        InsertTextBoxLine (panelHandle, PANEL_TEXTBOX_2,0,
"Configurat!");
        break;
    }
    return 0;
}

/*-----*/
/*      Botó Sortir:      Sortim del programa      */
/*-----*/

int CVICALLBACK SORTIR (int panel, int control, int event,
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
    }
    return 0;
}

```

```

/*-----*/
/*      Funció rebre_fins_temps: Adquirim els caràcters que entren per el COM1      */
/*                                  i COM2, utilitzant un bucle per tal de poder      */
/*                                  establir un ordre d'arribada del caràcters      */
/*                                  els arribats pel COM1 es mostren amb una R      */
/*                                  davant i els arribats per el COM2 amb una T.      */
/*                                  L'adquisició s'acaba quan ha transcorregut el      */
/*                                  temps escollit                                  */
/*-----*/

```

```
void rebre_fins_temps(unsigned int Tadqui)
```

```

{
    int a,
        i=0,
        nCar1,
        nCar2;

    a=Timer();

    while(Timer()<=(a+Tadqui))
    {
        nCar1=ComRd(1,&buff[i+1],1);
        if(nCar1==1)
        {
            buff[i]='R';
            i=i+2;
        }
        nCar2=ComRd(2,&buff[i+1],1);
        if(nCar2==1)
        {
            buff[i]='T';
            i=i+2;
        }
    }
}

```

```

/*-----*/
/*      Funció rebre_fins_temps: Adquirim els caràcters que entren per el COM1      */
/*                                  i COM2, utilitzant un bucle per tal de poder      */
/*                                  establir un ordre d'arribada del caràcters      */
/*                                  els arribats pel COM1 es mostren amb una R      */
/*                                  davant i els arribats per el COM2 amb una T.      */
/*                                  L'adquisició s'atura quan es prem alguna        */
/*                                  tecla.                                          */
/*-----*/

```

```
void rebre_fins_tecla(){
```

```

    int i=0,
        a=0,
        nCar1,
        nCar2;
    char ch=0;
    SetStdioWindowVisibility (1);
    while(ch!=1){

        nCar1=ComRd(1,&buff[i+1],1);

        if(nCar1==1)
        {
            buff[i]='R';
            i=i+2;
            a++;
        }
    }
}

```



```

        }
        nCar2=ComRd(2,&buff[i+1],1);
        if(nCar2==1)
        {
                buff[i]='T';
                i=i+2;
                a++;
        }
        if(KeyHit ())
        {
                ch=1;
                while(KeyHit())
                GetKey();
        }
    }
    SetStdioWindowVisibility (0);
}

/*----- */
/*      Funció taula_a_textbox:  Presenta per pantalla la informació que      */
/*      s'ha obtingut en lguna de les funcions rebre.                        */
/*----- */
/*----- */
/*----- */

void taula_a_textbox(unsigned char *buff)
{
        unsigned char  bufftemp[200];
        int            in=0,
                    a=0,
                    index=0,
                    longitud=0;

        ResetTextBox (panelHandle, PANEL_TEXTBOX, "");
        longitud=strlen (buff);

        while(a<longitud)
        {
                for(in=0;in<=23;in++)
                {
                        bufftemp[in*4]=buff[a];
                        sprintf( &bufftemp[in*4+1], "%02x ", buff[a+1] );
                        a=a+2;
                }
                InsertTextBoxLine(panelHandle, PANEL_TEXTBOX,index,bufftemp);
                index++;
        }
}

```

## Fitxer .h

```

/*****
/* LabWindows/CVI User Interface Resource (UIR) Include File      */
/* Copyright (c) National Instruments 2004. All Rights Reserved.  */
/*----- */
/* WARNING: Do not add to, delete from, or otherwise modify the contents */
/* of this include file.                                             */
/*****

#include <userint.h>

```



```

#ifdef __cplusplus
extern "C" {
#endif

    /* Panels and Controls: */

#define CONFIGURA 1
#define CONFIGURA_BITSSTOP 2
#define CONFIGURA_BITSDADES 3
#define CONFIGURA_PARITAT 4
#define CONFIGURA_SCONF 5 /* callback function: SCONF */
#define CONFIGURA_CUAENTRADA 6
#define CONFIGURA_CUASORTIDA 7
#define CONFIGURA_BITSSECON 8

#define PANEL 2
#define PANEL_TEXTBOX 2
#define PANEL_SORTIR 3 /* callback function: SORTIR */
#define PANEL_REBRE 4 /* callback function: REBRE */
#define PANEL_TEXTBOX_2 5
#define PANEL_CONFIG 6 /* callback function: CONFIG */
#define PANEL_CLEAN 7 /* callback function: CLEAN */
#define PANEL_CARREGAR 8 /* callback function: CARREGAR */
#define PANEL_GUARDAR 9 /* callback function: GUARDAR */
#define PANEL_TIPUSATURADA 11
#define PANEL_TADQUISICIO 12

    /* Menu Bars, Menus, and Menu Items: */

    /* (no menu bars in the resource file) */

    /* Callback Prototypes: */

int CVICALLBACK CARREGAR(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK CLEAN(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK CONFIG(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK GUARDAR(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK REBRE(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK SCONF(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK SORTIR(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);

#ifdef __cplusplus
}
#endif

```

## Bibliografia

---

Jurgen, Ronald K.; (1999). Automotive electronics handbook. 2nd ed. New York : McGraw-Hill, cop.

Ribbens, William B.:(1998). Understanding automotive electronics. 5th ed. Boston: : Newnes, cop.

Bauer, Horst.; (2000). Automotive handbook. 5th revised and extended ed. Stuttgart : Robert Bosch, cop.

Directiva 98/69/CE- "... relativa a las medidas que deben adoptarse contra la contaminación atmosférica causada por las emisiones de los vehículos de motor y por la que se modifica la Directiva 70/220/CEE del Consejo."

Davies, N.K.H.; Diagnostics at the roadside. Vehicle Diagnostics in Europe, IEE Colloquium on, 23 Feb 1994. Pages: 6/1 - 6/7.

Godavarty, S.; Broyles, S.; Parten, M.; Interfacing to the on-board diagnostic system. Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd , Volume: 4 , 24-28 Sept. 2000. Pages: 2000 - 2004 vol.4.

Greening, P.; On-board diagnostics for control of vehicle emissions. Vehicle Diagnostics in Europe, IEE Colloquium on , 23 Feb 1994. Pages: 5/1 - 5/6.

Harnett. P.; Miller. P.; O'Hara. M.; and Automotive Motorola Industrial Group.; 42V Powernet enabling technologies.

Holland, R.; The introduction of networking into the automotive industry. New Product Introduction in Electronics (Ref. No. 2000/047), IEE Seminar , 12 April 2000. Pages:3/1 – 3/6.

ISO structure and related activities [automobile electronics diagnostics] Communication Standards for European On-Board-Diagnostics Seminar (Ref. No. 1998/294), IEE , 16 Feb. 1998. Pages: 3/1 – 3/3.



Khan, I.A.; Automotive electrical systems: architecture and components. Digital Avionics Systems Conference, 1999. Proceedings. 18th , Volume: 2 , 24-29 Oct. 1999. Pages: 8.C.5-1 - 8.C.5-10 vol.2.

Kopetz,H.; Research Report 10/2001. A Comparison of TTP/C and FlexRay. Institut für Technische InformatikTechnische Universität Wien, Austria.

Kopetz,H.; Automotive electronics. Real-Time Systems, 1999. Proceedings of the 11th Euromicro Conference on , 9-11 June 1999.

Krug, M.; Schedl, A.V.; New demands for invehicle networks. EUROMICRO 97. 'New Frontiers of Information Technology'. Proceedings of the 23rd EUROMICRO Conference , 1-4 Sept. 1997. Pages: 601 – 605.

Leen, G.; Heffernan, D.; Expanding automotive electronic systems. Computer , Volume: 35, Issue: 1, Jan. 2002. Pages: 88 – 93.

Nicastri, P.R.; Huang, H.; Jump starting 42V PowerNet vehicles. Aerospace and Electronic Systems Magazine, IEEE , Volume: 15 , Issue: 8 , Aug. 2000. Pages:25 – 31.

Samuel, J.; Diagnostic Trouble Code Definitions. Communication Standards for European On-Board-Diagnostics Seminar (Ref. No. 1998/294), IEE , 16 Feb. 1998. Pages: 11/1 - 11/3.

Samuel, J.; Emission Related Diagnostic Services. Communication Standards for European On-Board-Diagnostics Seminar (Ref. No. 1998/294), IEE , 16 Feb. 1998. Pages: 10/1 - 10/6.

Scott, K.; The Communication Protocols: ISO 9141/KWP2000. Communication Standards for European On-Board-Diagnostics Seminar (Ref. No. 1998/294), IEE , 16 Feb. 1998. Pages: 4/1 - 4/4.

Scott, K.; The importance of automotive diagnostics in the search of excellence. AUTOTESTCON '92. IEEE Systems Readiness Technology Conference, Conference Record, 21-24 Sept. 1992. Pages: 27 – 30.

Roberts, C.I.; Peters, M.; Standardisation of serial diagnostics. Vehicle Diagnostics in Europe, IEE Colloquium on , 23 Feb 1994. Pages:3/1 – 3/4.

Richard Stobart, University of Sussex.; Automotive Electronics – Bringing a New Dimension to Driving.



Wang Xing; Huiyan Chen; Huarong Ding; The application of controller area network on vehicle. Vehicle Electronics Conference, 1999. (IVEC '99) Proceedings of the IEEE International, 6-9Sept.1999. Pages:455 - 458 vol.1.

### **Recursos Web**

Dearborn Group Technology

<http://www.dgtech.com>

OBDII Automotive Diagnostics Homepage

<http://obddiagnostics.com/index.htm>

OBD-2 Vehicle Explorer Scan Tool Browser

<http://www.obd-2.com/>

OBD-2 Homepage

<http://www.obd-2.de>

OBD-II - On-Board Diagnostic System

<http://www.obdii.com/>

Howstuffworks "Auto Channel"

<http://auto.howstuffworks.com/>

OpenDiag

<http://www.opendiaq.org/>

Jeff Noxon's Homepage. Opendiaq Schematics & PCB Layout

<http://www.planetfall.com/~jeff/obdii/>

Andy Whittaker's Homepage

<http://www.andywhittaker.com>

VAG-COM - Programa de Diagnòstico de Vehículos Volkswagen, Audi, Seat y Skoda

<http://www.vag-com-espanol.com>

Autoxuga movil - franquicia taller, calculo motor coches

<http://personal.telefonica.terra.es/web/autoxugamovil/SCANNER/SCANNER.htm>



Deciphering OBD-II Codes

<http://www.flatratetech.com/pub34.htm>

Basics of engine management

<http://hometown.aol.com/dvandrews/ems.htm#basics>

Motor – Actualidad. La contaminación no va a más

<http://motor.terra.es/motor/articulo/html/mot1594.htm>

Chevron Home Page. Fuel Technical Bulletins

<http://www.chevron.com/prodserv/fuels/bulletin/>

Desarrollo de Contenidos - Motor vs Contaminación

<http://www.oni.escuelas.edu.ar/2001/santa-fe-sur/motor/marcos.htm>

Automotive Training and Resource Site

<http://www.autoshop101.com/>

Ecutek

<http://www.ecutek.com/>

Motorola Automotive

<http://e-www.motorola.com/webapp/sps/site/homepage.jsp?nodeId=02Wcbf>

Literature on Controller Area Network

<http://www.can.bosch.com/content/Literature.html>

CAN BUS

<http://www.canbus.galeon.com/electronica/canbus.htm>

Interfacing The Serial / RS-232 Port

<http://www.beyondlogic.org/serial/serial1.htm#33>

FlexRay - The communication system for advanced automotive control applications

<http://www.flexray-group.com/index.php>

TTTech - Time-Triggered Technology

<http://www.tttech.com/technology/ttagroup.htm>





X-by-Wire, New Technologies for 42V Bus

[http://www.me.sc.edu/Research/AARG/honors\\_thesis.pdf](http://www.me.sc.edu/Research/AARG/honors_thesis.pdf)

Why Drive-by-Wire?

<http://www.edmunds.com/news/innovations/articles/43033/article.html>

Delphi technical papers

<http://www.delphi.com/pdf/techpapers/>

Electromechanical Braking (Brake-by-Wire)

<http://e-www.motorola.com/webapp/sps/site/application.jsp?nodeId=04J7IV07jSLR2Q>

km77. Articulos de tècnica

<http://www.km77.com/tecnica/motor/portada.asp>