



Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FINAL DE CARRERA

DESENVOLUPAMENT D'UN ENTORN DE TEST AUTOMÀTIC PER A COMPONENTS DE VEHICLES HÍBRIDS

Development of an automatic test environment for
components of hybrids vehicles

Estudis: Enginyeria Electrònica i Màster en electrònica

Autor: Carlos Barceló Vernet

Director: Aleix Maixé i Sas

Professor ponent: Ferran Silva

Any: 2012

Índex general

Índex general.....	1
Col·laboracions	3
Agraïments.....	5
Resum del Projecte.....	6
Resumen del Proyecto	8
Abstract.....	10
1. INTRODUCCIÓ	12
1.1. CONTEXT DEL PROJECTE.....	12
1.1.1. VEHICLE ELÈCTRIC.....	12
1.1.2. VEHICLE HÍBRID	14
1.1.3. SBOX.....	19
1.2. OBJECTIUS.....	24
1.3. ESTRUCTURA DE LA MEMÒRIA.....	24
2. ESTRUCTURA DEL SISTEMA	25
2.1. TEST BENCH	25
2.1.1. DIAGRAMA DE BLOCS DEL TEST BENCH.....	25
2.1.2. ESQUEMA ELÈCTRIC DEL TEST BENCH.....	27
2.2. TARGETA D'ADQUISICIÓ	33
2.2.1. CONDICIONS DE CONNEXIÓ.....	35
2.3. SISTEMA DE CALIBRATGE.....	37
2.3.1. CALIBRATGE AMB TENSIÓ	38
2.3.2. CALIBRATGE AMB CORRENT.....	41
3. ESTRUCTURA SOFTWARE.....	43
3.1. INTERFÍCIE LABVIEW-CANOE	43
3.1.1. ESCRITPURA LABVIEW-CANOE.....	43
3.1.2. LECTURA LABVIEW-CANOE	47
3.2. PROGRAMA PRINCIPAL	49
3.2.1. SETUP GPIB	51

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

3.2.2.	COMPROVACIÓ DE PROGRAMES	53
3.2.3.	COMPROVACIÓ DELS RELÉS	53
3.2.4.	SELECCIÓ TIPUS DE SBOX	54
3.2.5.	CONSUM DE LA SBOX.....	56
3.2.6.	CALIBRACIÓ I MESURA DE TENSIONS.....	58
3.2.7.	CALIBRACIÓ I MESURA DE LA CORRENT	61
3.2.8.	REPORT	66
3.2.9.	ALTRES PROGRAMES.....	73
3.2.10.	PROJECTE LABVIEW	77
3.3.	ALTRES PROGRAMES.....	81
3.3.1.	ESCALA DE TENSIONS	81
3.3.2.	PROGRAMA ESCALA DE TENSIONS	84
3.3.3.	ISOLATION MONITORING	85
3.3.4.	CRASH	86
	Conclusions.....	89
4.	REFERÈNCIES	90
5.	APÈNDIX.....	92

Col·laboracions



Agraïments

No puc deixar aquests apartat en blanc, ja que, hi han hagut persones molt importants durant aquests anys d'estudi i de projecte que han fet possible que pugués arribar fins aquí.

M'agradaria agrair al meu director de projecte Aleix Maixé i Sas per guiar-me i ensenyar-me en la meva estància a l'empresa Lear Corporation. No em puc oblidar de l'equip de treball de l'empresa, ja que també han format part del meu aprenentatge durant els mesos de pràctiques.

M'agradaria agrair al meu professor ponent Ferran Silva per guiar-me i facilitar-me en tot moment els tràmits alhora de poder treballar i realitzar el projecte final de carrera a Lear Corporation.

No em puc deixar l'agraïment a totes les persones que he conegut durant aquest dos anys a la ETSETB perquè sense elles no hagués sigut el mateix.

Per últim, agrair a tota la meva família, als meus pares Guillermo Barceló Castellnou i Rosalia Vernet Ferré, als meus germans Rosa Maria, Anna Maria i Guillermo i a la meva àvia Rosa Ferre Vaqué que sempre m'han apoïat en tot i han sigut una guia i una referència a seguir per no perdrem en aquest llarg camí.

Resum del Projecte

L'objectiu principal del projecte és calibrar i verificar les tensions i les intensitats d'un component d'un vehicle híbrid de BMW anomenat SBOX que es fabrica a l'empresa Lear Corporation. Aquest procés de calibració i mesura es realitzarà mitjançant un desenvolupament d'un entorn de test automàtic basat amb el programa LabVIEW.

En aquests moments, a l'empresa Lear Corporation, la verificació de la SBOX és lenta perquè es tenen de moure molts cables per realitzar una mesura i difícil, ja que, es necessita personal qualificat doncs un possible error en el procés podria danyar el dispositiu. El objectiu per tant, és realitzar una verificació de la SBOX amb l'objectiu que sigui: el més automàtica possible, és a dir, que es tingui el operari hagi de fer el mínim de manipulacions possibles, el més ràpid possible, és a dir, que es pugui realitzar en minuts i que el procés sigui fàcil perquè no faci falta cap persona qualificada per manipular el test.

El test automàtic es dividirà en 2 parts. La estructura dels sistema o també anomenada estructura hardware on es realitzarà el disseny d'un banc de proves i on hi haurà diferents dispositius que ajudaran a calibrar i validar el dispositiu, i la estructura software on dissenyarem diferents programes LabVIEW per controlar aquests dispositius i la SBOX.

Un cop finalitzat el test es crearà un report en forma d'arxiu Excel on es podrà observar si les tensions i corrents que s'han calibrat s'han validat correctament, és a dir, compleixen amb els marges d'error que ens ha especificat BMW per al seu correcte funcionament. D'aquesta manera, si compleix amb totes les especificacions voldrà dir que el dispositiu està llest per instal·lar-lo al cotxe i sinó es sabrà en quins valors de tensió o de corrent falla per poder corregir l'error més fàcilment.

Per realitzar tot aquest banc de proves s'ha realitzat un estudi teòric dels dispositius i dels programes utilitzats durant el transcurs del projecte. Aquest estudi es pot observar en el apartat de l'annex.

Per últim, cal dir, que en el transcurs dels mesos de pràctiques a Lear Corporation també s'ha realitzat diferents treballs que estan a l'annex, com per exemple, un document de formació per al personal de Lear per poder manipular el banc de proves i un document de pràctiques amb el projecte VERDE que ha servit per aprendre a manipular el programa CANoe (programa que es farà servir en el transcurs del projecte).

Resumen del Proyecto

El objetivo principal del proyecto es calibrar y verificar las tensiones y las intensidades de un componente de un vehículo híbrido de BMW llamado SBOX que se fabrica en la empresa Lear Corporation. Este proceso de calibración y medida se realizará mediante un desarrollo de un entorno de test automático basado en el programa LabVIEW.

En estos momentos, a la empresa Lear Corporation, la verificación de la SBOX es lenta porque se tienen que mover muchos cables para realizar una medida y difícil, ya que, se necesita personal calificado, pues un posible error en el proceso podría dañar el dispositivo. El objetivo por tanto, es realizar una verificación de la SBOX con el objetivo de que sea: lo más automática posible, es decir, que se tenga el operario tenga que hacer el mínimo de manipulaciones posibles, lo más rápido posible, es decir, que se pueda realizar en minutos y que el proceso sea fácil para que no haga falta ninguna persona calificada para manipular el test.

El test automático se dividirá en 2 partes. La estructura del sistema o también llamada estructura hardware donde se realizará el diseño de un banco de pruebas y donde habrá diferentes dispositivos que ayudarán a calibrar y validar el dispositivo, y la estructura software donde diseñaremos diferentes programas LabVIEW para controlar estos dispositivos y la SBOX.

Una vez finalizado el test se creará un report en forma de archivo Excel donde se podrá observar si las tensiones y corrientes que se han calibrado se han validado correctamente, es decir, cumplen con los márgenes de error que nos ha especificado BMW para su correcto funcionamiento. De esta manera, si cumple con todas las especificaciones querrá decir que el dispositivo está listo para instalarse en el coche y sinó se sabrá en qué valores de tensión o de corriente falla para poder corregir el error más fácilmente.

Para realizar todo este banco de pruebas se ha realizado un estudio teórico de los dispositivos y los programas utilizados durante el transcurso del proyecto. Este estudio se puede observar en el apartado del anexo.

Por último, cabe decir, que en el transcurso de los meses de prácticas en Lear Corporation también se ha realizado diferentes trabajos que están en el anexo, como por ejemplo, un documento de formación para el personal de Lear para poder manipular el banco de pruebas y un documento de prácticas con el proyecto VERDE que ha servido para aprender a manipular el programa CANoe (programa que se utilizará en el transcurso del proyecto).

Abstract

The main objective of this project is to calibrate and verify the voltage and the current intensities of one hybrid BMW vehicle's component called SBOX manufactured by Lear Corporation. This measurement and calibration process is made through the development of one automate testing environment based on the LabVIEW program.

At Lear Corporation to verify and gauge this component it is now a very slow and difficult process due to the many wires they have to move and because it demands highly skilled staff to avoid errors that could seriously damage the SBOX during the process.

At present, the main goal is to achieve the most automatic measurement and control system of the SBOX as to reduce as much as possible any manipulation by the staff and to ensure the easiest and quickest process, with no need of qualified personnel to manipulate the test.

The automatic test will be divided into two parts. The systems structure, also known as hardware structure, where a testing-bench design will be performed and where there will be different devices that will help to calibrate and validate the component and the software structure where different LabVIEW programs will be designed to control these devices and the SBOX. Once the test has finished a report in the form of an Excel file will be created.

This report will allow observing if the validation of the voltages and currents has been done correctly and if they accomplish with the error margins specified by BMW for a correct performance. If all the specifications are being accomplished it would mean that the device is ready to be installed into the car. If not, it would help knowing which tensions and currents fail and the errors would be easily corrected.

To carry on with this test-bench a theoretical study was made of all the devices and programs used during the development of the project. This study is annexed for information.

Only to mention that, during the practical training period at Lear Corporation, some other different field-works were carried on, which are compiled in the attachment, as for instance, one educational document for the Lear's staff to be able to manipulate the testing-bench and another training document with the GREEN project which has served for the staff to learn how to work with the CANoe program (a program to be used during the project).

1. INTRODUCCIÓ

1.1. CONTEXT DEL PROJECTE

En aquest apartat s'introduirà en el projecte explicant de manera breu el funcionament del tipus de vehicles elèctrics i especialment els tipus de vehicles elèctrics on s'instal·la la SBOX. Per altra banda, també s'explicarà les funcions i característiques principals d'aquest dispositiu.

1.1.1. VEHICLE ELÈCTRIC

El vehicle elèctric convencional s'impulsa mitjançant un o varis motors elèctrics. L'energia amb que s'impulsa el motor es guarda a la bateria d'alta tensió i es carrega a la xarxa elèctrica mitjançant un carregador AC/DC que converteix la tensió en continua. El inversor s'encarrega de convertir la tensió en alterna per al correcte funcionament del motor. Per altra banda, també hi ha un convertidor DC/DC que s'encarrega de reduir la tensió a 12 V per la bateria i aquesta subministra la tensió en les parts i circuits electrònics del cotxe que s'hagi d'alimentar a 12V.

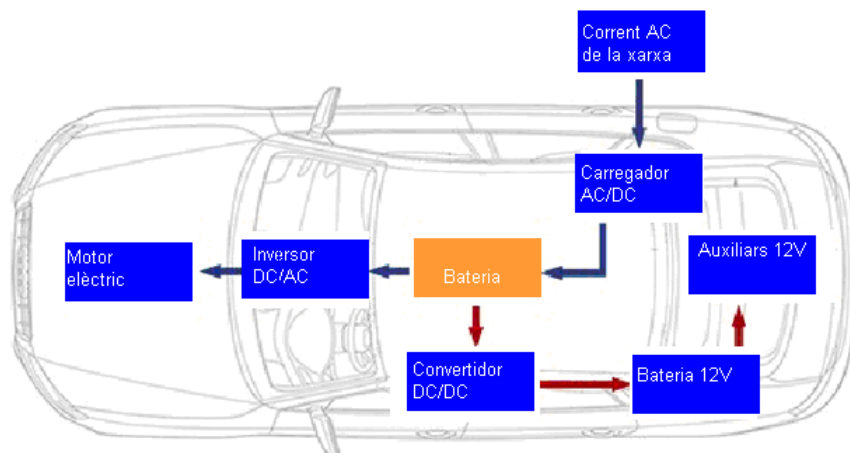


Figura 1.1 Diagrama de blocs d'un vehicle elèctric [1]

1.1.1.1. Característiques dels vehicles elèctrics

A part del vehicle elèctric convencional que s'ha vist en l'apartat anterior, en la següent taula es pot observar les diferents característiques per a altres tipus de vehicles elèctrics.

TIPUS DE EVs	Bateria EVs	Híbrid EVs	Fuel Cell EVs

Propulsió	- Motor elèctric	- Motor elèctric - Motor de combustió interna	- Motor elèctric
Sistema d'energia	- Bateria - Ultracapacitador	- Bateria - Ultracapacitador - ICE generating unit	-Piles de combustible - Necessita bateria o ultracapacitador per augmentar la densitat de potència per a l'arrencada
Font d'energia i infraestructura	-Instal·lacions elèctriques per la xarxa de càrrega	-Estacions de gasolina -Instal·lacions elèctriques per la xarxa de càrrega (per endolls de Híbrid)	- Hidrogen -Producció d'hidrogen i infraestructura de transport
Característiques	-Zero emissió -Alta eficiència energètica -Independència dels crús -Abast relativament curt - Alt cost inicial -Comercialment disponible	- Baixa emissió -Economia de combustible superior als vehicles ICE -Temps de conducció llarg - Dependència del crús (per endolls de Híbrids) - Augment en el consum de	-Costos de combustible cel·lular, cicle de vida i fiabilitat -Infraestructura d'hidrogen

		combustible i reducció en les emissions en funció del nivell de potència del motor i la bateria, així com el cicle de conducció -Comercialment disponible	
--	--	--	--

Taula 1.1 Característiques del vehicle híbrid [2]

A continuació, s'explicarà de forma més detallada el tipus de vehicle elèctric on s'instal·la el dispositiu SBOX que es tractarà en el projecte.

1.1.2. VEHICLE HÍBRID

Un vehicle híbrid pot funcionar amb més d'una font d'energia. Normalment s'utilitza aquest terme per a designar els automòbils que poden funcionar amb motors elèctrics i de combustió. En el que l'energia elèctrica que l'impulsa prové de bateries i, alternativament, d'un motor de combustió interna que mou un generador. Normalment, el motor també pot impulsar les rodes de forma directa.

En el disseny d'un automòbil híbrid, el motor tèrmic és la font d'energia que s'utilitza com última opció, i es disposa d'un sistema electrònic per determinar quin motor utilitzar, i quan fer-ho.

En el cas d'híbrids gasolina-elèctrics, quan el motor de combustió interna funciona, ho fa amb la seva màxima eficiència. Si es genera més energia de la necessària, el motor elèctric s'utilitza com a generador i carrega la bateria del sistema. En altres situacions, funciona només el motor elèctric, alimentant-se de l'energia guardada a les bateries.

N'hi ha alguns en què es possible recuperar l'energia cinètica en la frenada i la converteix en energia elèctrica.

La combinació d'un motor de combustió operant sempre a la seva màxima eficiència, i la recuperació d'energia en la frenada (útil especialment a la ciutat), fa que aquests vehicles tinguin millors rendiments que els vehicles convencionals.

Tots els vehicles elèctrics utilitzen bateries carregades per una font externa, fet que ocasiona problemes d'autonomia de funcionament. Aquesta queixa habitual, s'evita amb els cotxes híbrids, ja que és el motor tèrmic (normalment de gasolina), el que, mitjançant un generador, carrega les bateries.

1.1.2.1. Avantatges i inconvenients

Els avantatges i inconvenients per la elecció d'un vehicle híbrid respecte a un vehicle elèctric convencional i un vehicle de combustible són:

- Avantatges:
 - Menys soroll que un motor tèrmic.
 - Resposta més ràpida.
 - Recuperació d'energia en desacceleracions.
 - Més autonomia que un elèctric simple.
 - Més suavitat i facilitat d'ús.
 - Recarrega més ràpid que un elèctric.
 - Millor funcionament en recorreguts curts.
 - Consum molt inferior a un vehicle de combustible.
- Inconvenients:
 - Més pes que un vehicle tradicional (vehicle de combustible).
 - Més possibilitat d'espantllar-se.
 - De moment, el preu. És més car que un vehicle de combustible.

1.1.2.2. Classificació dels vehicles híbrids (HEV)

Es pot classificar els vehicles híbrids en tres tipus paral·lel, sèrie i sèrie/paral·lel.

1.1.2.2.1. Paral·lel

És un sistema on el motor elèctric i el motor tèrmic poden fer girar les rodes.

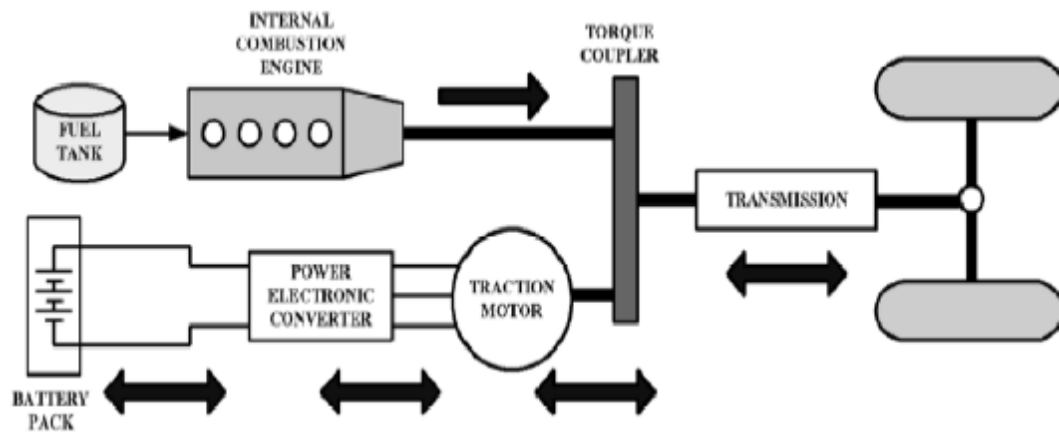


Figura 1.2 Diagrama d'un híbrid paral·lel [2]

1.1.2.2.2. Sèrie

Sistema on el motor tèrmic genera electricitat i la tracció només la proporciona el motor elèctric.

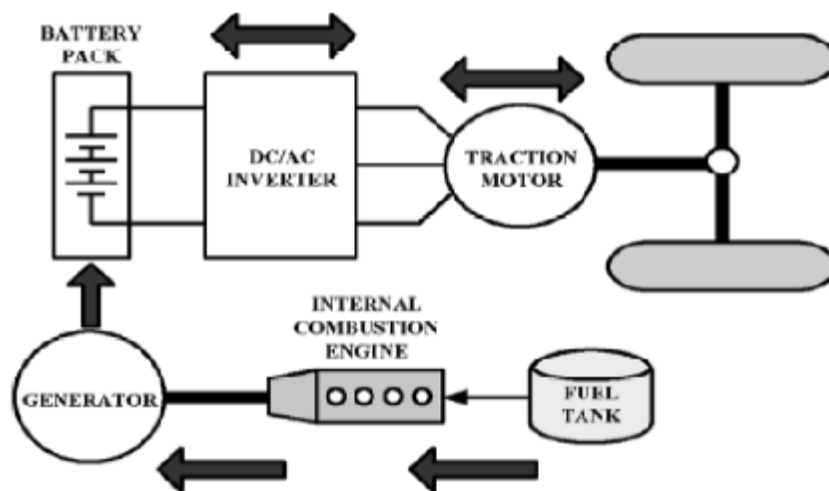


Figura 1.3 Diagrama d'un híbrid sèrie [2]

1.1.2.2.3. Sèrie/Paral·lel

Sistema on el motor tèrmic i el motor elèctric poden fer les dos funcions, és a dir, generar electricitat i proporcionar la tracció a les rodes.

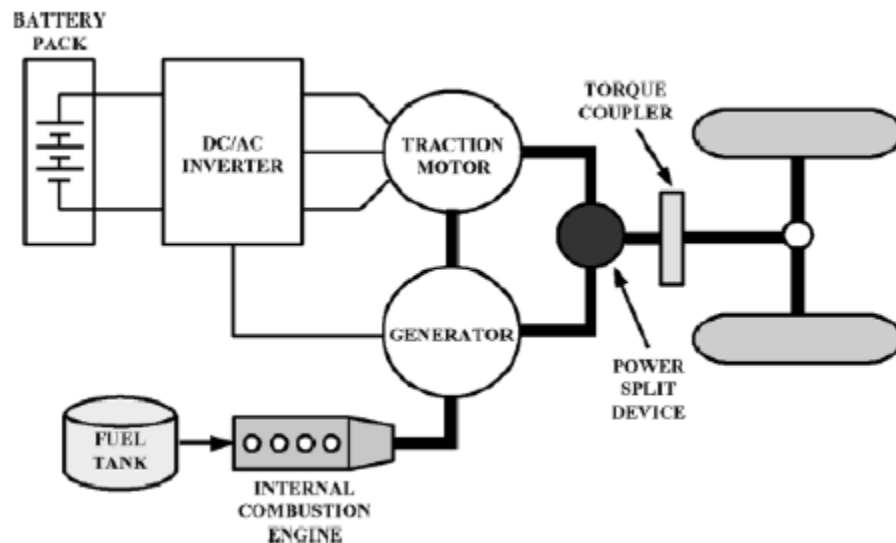


Figura 1.4 Diagrama d'un híbrid sèrie/paral·lel [2]

Per altra banda, també es pot classificar el vehicle híbrid en:

- Regulars: Utilitzen el motor elèctric com a suport.
- Endollables: També coneguts per les seves lletres en anglès PHEVs (plug-in hybrid electric vehicle) utilitzen principalment el motor elèctric i un cop s'acaba l'energia en la bateria s'utilitza el combustible. Es poden recarregar endollant-los a la xarxa elèctrica.

En el següent apartat, s'ampliarà la informació sobre el vehicles PHEV, ja que, és el tipus de vehicle que disposa de la SBOX, dispositiu que s'aplicarà el test automàtic que es dissenyarà en el projecte.

1.1.2.3. PHEV

En el vehicle híbrid elèctric endollable o PHEV (plug-in hybrid electric vehicle) les bateries poden ser recarregades endollant el vehicle a una font externa d'energia elèctrica. El vehicle híbrid endollable comparteix les característiques d'un vehicle híbrid elèctric tradicional i d'un vehicle elèctric, ja que, està dotat d'un motor de combustió interna (gasolina, dièsel o flex-fuel) i d'un motor elèctric acompanyat d'un paquet de bateries que es poden recarregar endollant el vehicle en el sistema de subministrament elèctric [3].

El primer vehicle híbrid endollable de producció comercial del món va ser el Sedan F3DM PHEV-68 (PHEV109km), fabricat per l'empresa xinesa BYD Auto, llançat el 15 de desembre de 2008 al mercat local únicament per a ús en flotes a un preu de 22.000\$. Toyota, General

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

Motors, Ford, Fisker Automotive, Aptera Motors, Volkswagen, Volvo i Hyundai van anunciar la producció d'automòbils híbrids endollables entre 2010 i 2012. El vehicle esportiu de luxe Fisker Karma PHEV-50 (PHEV-80 km) es ven desde el setembre de 2010 i el Chevrolet Volt PHEV-40 (PHEV-64 km) de General Motors va programar el seu llançament per al novembre de 2010. El vehicle esportiu utilitari Ford Escape PHEV-30 (PHEV-48 km) ja s'ha provat als Estats Units en flotes de companyies de serveis públics i està programat per ser venut al públic en general al 2012 [3].

Fins al 2010, la majoria dels híbrids endollables que van circular als Estats Units van ser conversions dels models Toyota Prius o Ford Escape Hybrid, els quals van ser modificats per utilitzar bateries recarregables i estan dotats d'un major nombre de bateries per estendre el rang d'operació en mode exclusivament elèctric. Diversos països, incloent els Estats Units i diversos països europeus, han aprovat lleis per facilitar la introducció dels vehicles híbrids endollables, incloent crèdits d'impostos, reglamentació sobre emissions permeses, i finançament de recerca i desenvolupament de bateries i altres tecnologies avançades [3].

1.1.2.3.1. Avantatges i inconvenients

Els avantatges i inconvenients d'un vehicle PHEV respecte a un cotxe elèctric convencional i a un cotxe de combustible són:

- Avantatges:

- Costos de l'operació: El cost de l'electricitat per operar un híbrid endollable en mode exclusivament elèctric s'estima que és menys de la quarta part del cost de la gasolina d'un vehicle tradicional.
- Reducció de la contaminació i emissió de carbono: Els PHEV redueixen la contaminació atmosfèrica, la dependència en el petroli i els combustibles fòssils i les emissions de gasos d'efecte hivernacle que contribueixen al escalfament global.
- Eliminació del temor d'esgotar la bateria: S'elimina aquest temor gràcies a que quan s'acaba la bateria del vehicle, s'utilitza el combustible convencional.

- Inconvenients:

- Cost, durabilitat i seguretat de les bateries: Augment del cost i de les dimensions de les bateries i disminució de la seguretat de les bateries.

- Disponibilitat d'infraestructura per recarregar: Problemes en el cas que es deixi el cotxe fora al carrer durant tota la nit, sense poder carregar-lo, ja que, actualment, no existeixen llocs sinó estàs dins d'una casa.
- Emissions de carbono degudes al tipus d'electricitat: Serà possible aquesta reducció si s'utilitza el mode elèctric i es carrega el vehicle amb electricitat que prové d'energies renovables.
- La sobrecarrega del sistema de subministrament elèctric: La xarxa elèctrica no està preparada per resistir la carga addicional que es generaria al connectar milers de vehicles.
- Disponibilitat del liti i altres metalls: La utilització d'aquest tipus de metalls per a les bateries podria crear a llarg plaç un canvi de la dependència del petroli per la dependència d'aquest materials

1.1.3. SBOX

Aquest projecte no s'especificarà totes les operacions internes de la SBOX, ja que, l'objectiu principal és realitzar un disseny amb diferents tests automàtics per tal de calibrar les tensions i la corrent que hi ha en la SBOX i també per comprovar que un cop calibrades, les tensions i la corrent compleixen amb els marges d'error especificats per BMW.

La intenció d'aquest apartat doncs, és donar una idea general de quines funcions realitza la SBOX dins d'un vehicle i de quins dispositius està compost. D'aquesta manera s'introduirà el objectiu principal del projecte definit anteriorment.

1.1.3.1. Definició

La SBOX és un dispositiu que està instal·lat en els vehicles elèctrics endollables (PHEV) de la marca BMW. És el dispositiu en el que es basarà el projecte, ja que, se l'hi aplicarà el test automàtic amb el programa LabVIEW per validar que un cop muntat, el dispositiu compleix amb totes les especificacions marcades per BMW per tal de poder instal·lar-se en el vehicle.

En la següent figura es pot observar la forma que te la SBOX. Per fer-nos una idea, les mides aproximades del dispositiu són de 40cm de llarg, 10cm d'ample i uns 12cm d'alçada.



Figura 1.5 SBOX

1.1.3.2. Funcions i característiques

La funció principal de la SBOX és controlar el subministrament de l'energia de la bateria d'alta tensió a diferents dispositius del cotxe, com l'aire condicionat, el convertidor DC/AC que controla el motor, el reductor que controla la bateria de 12V i el convertidor DC/AC que serveix per connectar-la a la xarxa elèctrica.

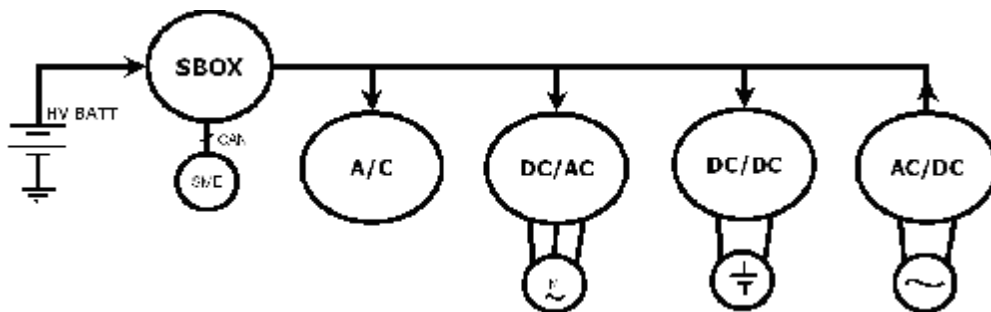


Figura 1.6 Diagrama de blocs del vehicle

El control del subministrament de la bateria d'alta tensió es farà mitjançant tres relés interns de la SBOX que tallaran o connectaran la bateria d'alta tensió.

La SBOX tindrà una PCB de control que serà la centralita d'aquest dispositiu i realitza, a grans trets, les següents funcions:

- Control dels relés. Ordena l'activació i desactivació dels relés.
- Comunicació via CAN a la SME que és una centralita del cotxe desde on es controla la SBOX. Mitjançant el protocol de comunicació CAN s'enviaran i es rebran les trames a la centralita per realitzar les mesures de les diferents tensions i la

corrent internes de la peça, per realitzar la mesura dels voltatges i el diagnòstics dels seus relés i també per realitzar el control d'aquests relés.

Per que fa a la seguretat, la SBOX té una característica important anomenada Functional Safety Asil C. Aquesta característica cataloga al dispositiu com un element important en el tema de seguretat del vehicle, ja que, significa que un possible error en la peça provocaria danys greus en els ocupants del vehicle.

En la següent figura es pot observar el esquema elèctric de la SBOX, és a dir com van situats els tres relés que es fan servir per connectar i desconectar la bateria d'alta tensió i sobretot, les mesures que serveixen per verificar la SBOX són aquestes quatre tensions i la intensitat.

- La U_b és la tensió d'entrada de la SBOX, és a dir, és la tensió que prové de la bateria d'alta tensió.
- La U_{zk} és la tensió de sortida de la SBOX, és a dir, és la tensió que arribarà a les parts del cotxe que estan connectats a la bateria d'alta tensió, com per exemple, els diferents convertidors.
- La U_{k2} és la tensió que l'hi cau al relé intern K2.
- La U_q que és la suma entre la U_{zk} i U_{k2} .
- La I és la corrent que circula pel relé K2.

Per altra banda, en el esquema elèctric també es pot observar altres components, com per exemple:

- R1 i R2. Impedància equivalent al circuit degut a la PCB de control i a tots el seus components (resistències, microprocessadors, etc.).
- Una Resistència R que es col·loca en sèrie al relé intern K3.
- R3. Impedància equivalent al circuit degut a la PCB de control i a tots el seus components (Resistències, microprocessadors, etc.) .
- Un Shunt per poder realitzar les mesures en corrent que hi ha a la SBOX.
- Un fusible en la entrada positiva de la SBOX per protegir el dispositiu de possibles sobretensions.

- C. Impedància capacitativa equivalent al circuit degut a la PCB de control i a tots els seus components (condensadors, microprocessadors, etc.).

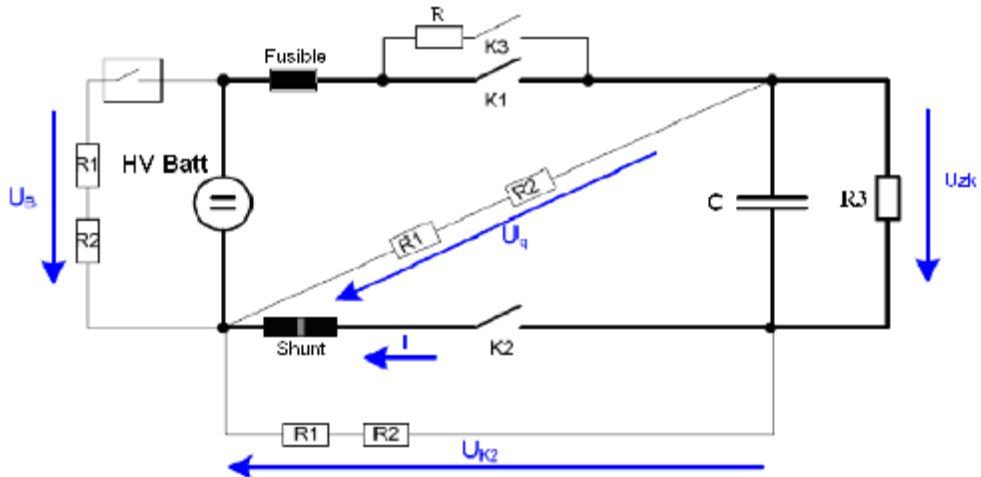


Figura 1.7 Esquema elèctric de la SBOX

En la següent figura es pot observar la situació de tots els components en una vista de planta. La connexió elèctrica és la següent:

- En la part superior del diagrama hi ha els diferents connectors d'entrada (U_{batt+} i U_{batt-}) i sortida (U_{zk+} i U_{zk-}) de la SBOX i també el connector de CAN. S'utilitzaran connectors especials que evitin possibles desconexions en cas de possibles vibracions en el funcionament normal del vehicle.
- En negre són els cables que connecten amb els connectors de sortida (U_{zk+} i U_{zk}).
- En vermell són els cables que connecten amb els connectors d'entrada (U_{batt+} i U_{batt-}).
- En taronja, blau i verd són els cables interns per connectar internament els diferents dispositius i la placa de control.
- En blau també són les connexions CAN.

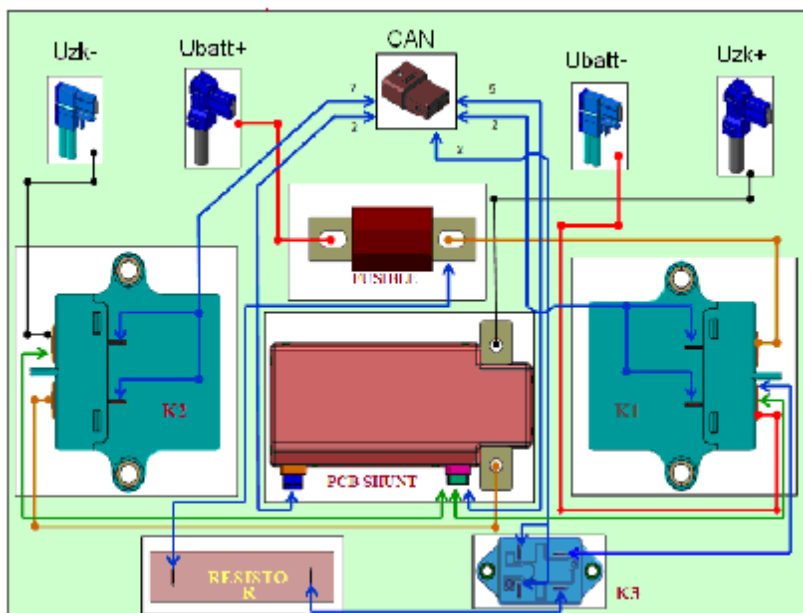


Figura 1.8 Diagrama elèctric de la SBOX

El diagrama de blocs de la estructura hardware de la placa de control de la SBOX és la següent.

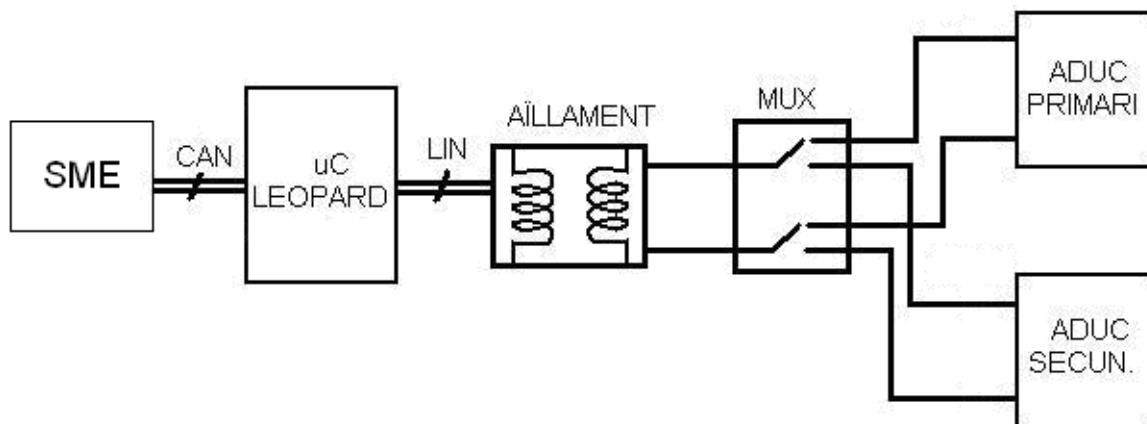


Figura 1.9 Diagrama elèctric de la SBOX

A grans trets, el esquema està format per un microprocessador principal que és el Leopard que es comunica amb els dos altres microprocessadors ADUCs (Analog Devices Microcontroller) a través de un LIN intern. El Leopard i els ADUCs estan aïllats i mitjançant un multiplexor es selecciona un dels dos ADUCs.

El funcionament del diagrama consisteix en que el Leopard rep les instruccions a través de la SME (centraleta del cotxe on es controlen totes les operacions d'aquest) mitjançant el

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

canal de comunicació CAN i aquest dona les instruccions als dos altres microprocessadors, el ADUC primari i el ADUC secundari.

Aquests tres microprocessadors són els encarregats de realitzar totes les operacions de calibratge i mesura de les tensions i de la intensitat, i també de realitzar totes les funcions que té la SBOX, com per exemple, obrir i tancar els relés.

1.2. OBJECTIUS

L'objectiu és realitzar un entorn de test automàtic per validar la SBOX i per millorar el procediment actual que hi ha instal·lat a l'empresa Lear Corporation que dissenya i fabrica el dispositiu SBOX per la marca BMW.

Actualment el procés de validació d'una SBOX dura unes 3 hores, amb la instal·lació d'aquest test es tardaria uns 20 minuts aproximadament, el que suposaria un augment de SBOXs a comercialitzar, és a dir, un augment de ingressos de Lear Corporation per aquest producte.

1.3. ESTRUCTURA DE LA MEMÒRIA

La estructura de la memòria es dividirà en dos apartats:

- L'estructura del sistema on s'explicarà el disseny d'un banc de proves (Test Bench) i el sistema de calibratge que s'implementa per verificar la SBOX.
- L'estructura software on s'explicarà el disseny dels programes LabVIEW que controlen aquest banc de proves i la SBOX.

2. ESTRUCTURA DEL SISTEMA

2.1. TEST BENCH

El Test Bench s'ha dissenyat en el transcurs del projecte on s'han seleccionat tots els dispositius i el nombre de components a utilitzar, com per exemple, el nombre de relés, entre d'altres. Un cop s'ha tingut tot el diagrama de blocs dissenyat, s'ha contactat amb una empresa (Serret Electrònica Valls) perquè dissenyés a baix nivell (drivers de relés i connexionat entre els dispositius) i construís (tota la estructura d'acer) el Test Bench amb els diferents dispositius que se l'hi han proporcionat. Cal dir que, també s'ha supervisat la construcció d'aquest Test Bench, per tal de que complís amb tots les necessitats.



Figura 2.1 Test Bench

2.1.1. DIAGRAMA DE BLOCS DEL TEST BENCH

En la següent figura és pots observar el diagrama de blocs general de l'estructura hardware del projecte amb les diferents tipus de connexions.

Tot el disseny està controlat mitjançant el programa LabVIEW a través d'un PC, amb l'objectiu de verificar de forma automàtica la SBOX. Amb el PC es controlarà via GPIB les fonts de tensió i d'intensitat, les fonts de mesura, la font de tensió auxiliar (DC POWER SUPPLY) i la carga que servirà per realitzar totes les proves de calibratge i mesura.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

El programa LabVIEW es controlarà mitjançant un panell on hi haurà diferents variables de control i un cop acabat aquestes proves es realitzarà un report amb les mesures realitzades i es comprovarà si compleixen amb les especificacions.

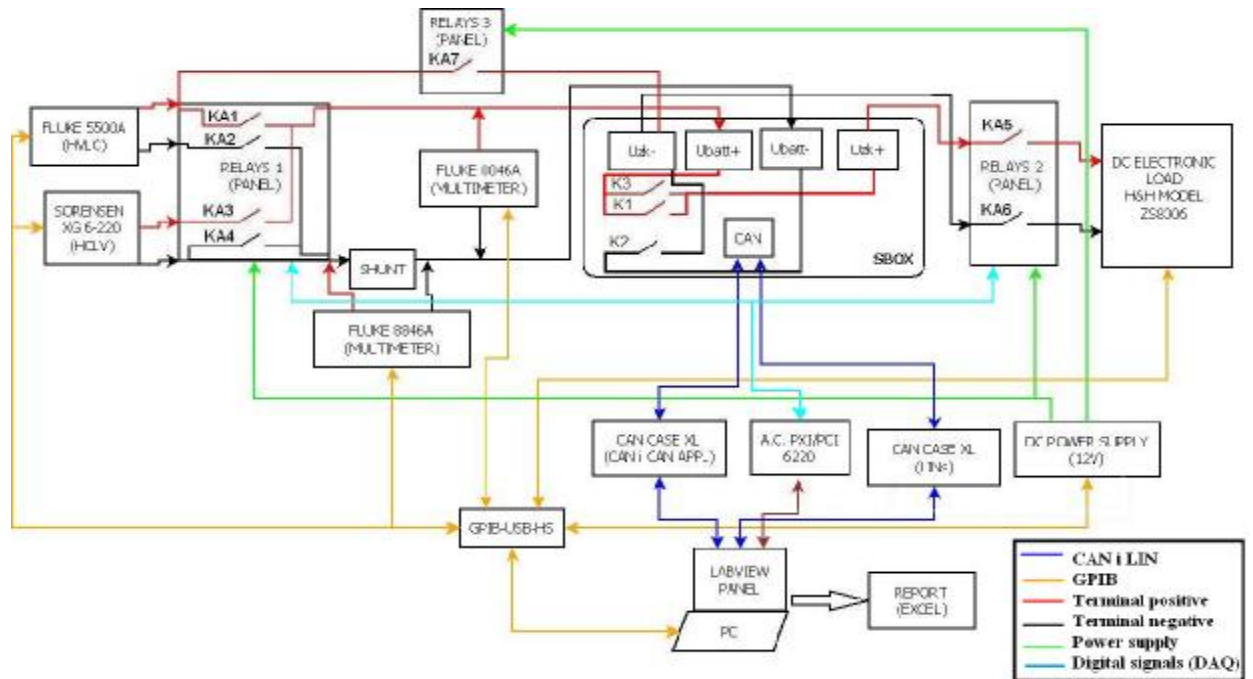


Figura 2.2 Diagrama de blocs del Test Bench

En el transcurs del calibratge i de la mesures s'anirà connectant els relés (panell de relés 1, 2 i 3) de forma automàtica mitjançant una targeta d'adquisició. Depenén de la tensió i intensitat que es vulgui calibrar i depenén de la mesura que es vulgui fer en la SBOX (U_{batt}, U_{zk}, U_{k2}, U_q i I_{batt}), es tancarà uns relés o uns altres. Aquestes condicions de connexió s'explicarà en el apartat de la targeta d'adquisició que és el dispositiu que controla les senyals digitals que permeten controlar els relés.

En les calibracions i mesures de tensió només es farà servir la font d'alta tensió i baixa corrent (HVLC) i per realitzar tots els calibratges i mesures de corrent, s'utilitzarà la font de baixa tensió i alta corrent (HCLV) i la carga.

Les calibracions i les mesures de la SBOX es realitzaran a través dels busos de comunicació CAN i LIN, és a dir, mitjançant el programa CANoe i dos targetes CANCEXSL:

- 1.- La targeta CANCEXSL amb un canal de CAN, on es realitzarà totes les mesures de les tensions i de la corrent, i un canal de CAN APPL, on es realitzaran tots els procediments de calibratge en tensió.

2.- La targeta CANCASEXL amb els dos LINs per calibrar i mesurar la corrent.

Per altra banda, pel que fa a les mesures externes del circuit, es farà mitjançant dos multímetres on es podrà mesurar la entrada de tensió que hi ha a la SBOX i la tensió que cau en el shunt. Per últim, també es disposarà d'una DC POWER SUPPLY auxiliar que servirà per alimentar els relés de la PCB i els relés del panell.

2.1.2. ESQUEMA ELÈCTRIC DEL TEST BENCH

En la següent figura es pot observar el esquema elèctric de tot el circuit i en vermell les entrades i sortides que necessita el panell de relés per connectar-se amb els dispositius i la SBOX:

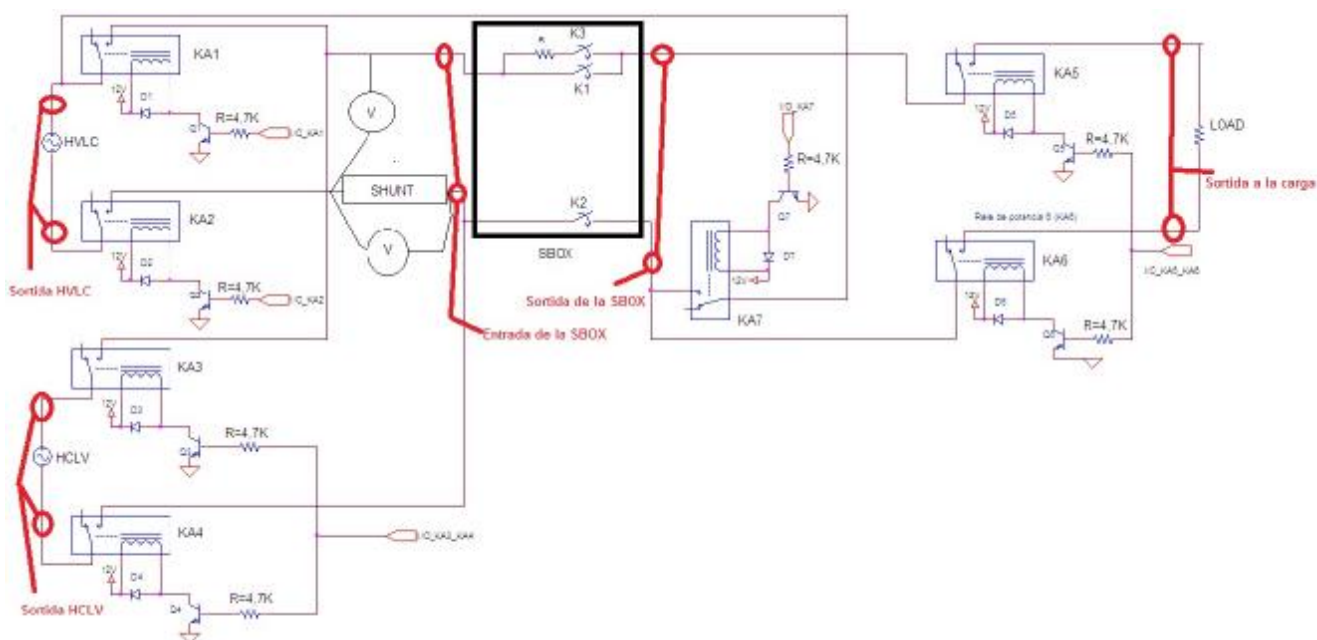


Figura 2.3 Esquema elèctric del Test Bench

2.1.2.1. Panell de relés

El panell de relés és una capsula rectangular (una mena de calaix dins del banc de proves) on només es podrà veure una tapa rectangular situada a la part superior del Test Bench. Els dispositius que hi han a la tapa són els següents:

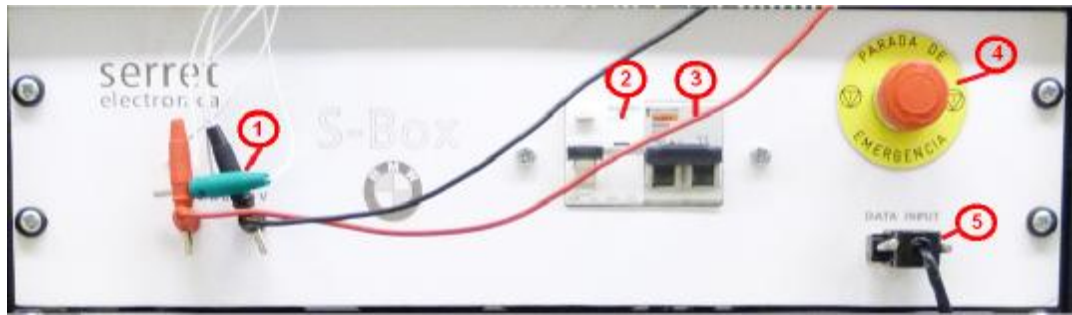


Figura 2.4 Panell de relés. Vista frontal

- 1.- Els dos terminals de 12 V serviran per a l'activació dels drivers dels relés.
- 2.- El Interruptor Diferencial protegeix als operaris de possibles derives que hi hagi en algun equip.
- 3.- El Interruptor Magnetotèrmic protegeix als equips en cas que hi hagi una sobretensió en la instal·lació elèctrica.
- 4.- El pulsador d'emergència desconnecta de forma manual tots els relés del panell independentment de les entrades digitals.
- 5.- La entrada DB9 per on es connectaran totes les entrades digitals procedents de la targeta d'adquisició.

Les entrades i sortides pertinents per connectar els dispositius del panell de relés a la resta del circuit estaran a la part posterior del panell.

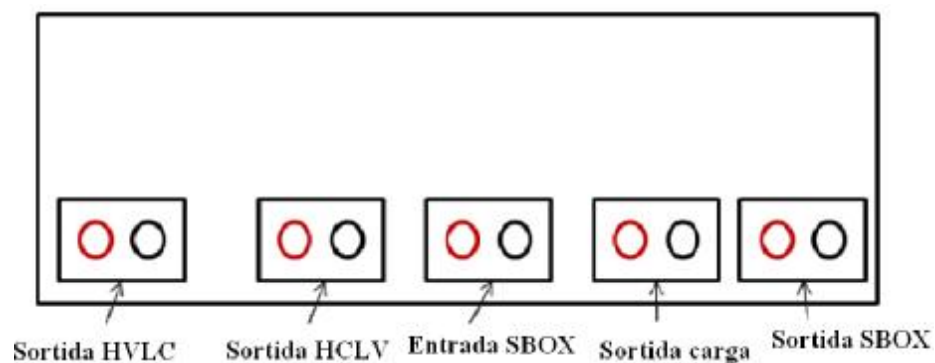


Figura 2.5 Panell de relés. Vista posterior

En el interior del panell de relés hi haurà:

- 7 relés (KA1, KA2, KA3, KA4, KA5, KA6 i KA7) amb els seus drivers.

Pel que fa al tipus de relés, els KA1, KA2 i KA7 seran els mateixos relés que s'utilitzen a la SBOX (Panasonic AEV120A i 12V) i els relés KA3, KA4, KA5 i KA6 són uns altres relés que aguanten una corrent de més de 400A (Kilovac EV200). Cal dir, que cada relé estarà format per un driver (format per un transistor darlington, un díode, una resistència i una entrada digital) que servirà per controlar-lo.

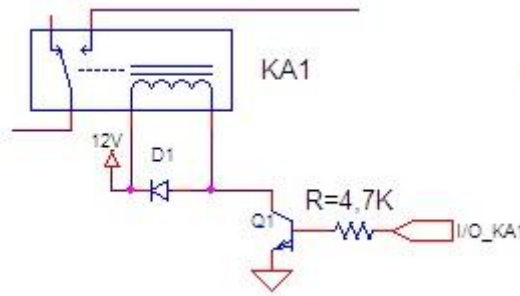


Figura 2.6 Drivers dels 7 relés

- El Shunt ($60\mu\Omega$) que es farà servir per mesurar la corrent.
- 5 Senyals digitals: I/O_KA1 (activa KA1), I/O_KA2 (activa KA2), I/O_KA3_KA4 (activa KA3 i KA4) i I/O_KA5_KA6 (activa KA5 i KA6).

Per motius de seguretat, les senyals digitals que activen els relés hauran de complir unes característiques i unes condicions de connexió:

- Els relés KA1, KA2 i KA7 connectaran la font de tensió HVLC. Aniran connectats de tal manera que, a través del software, quan es tanca KA1 es tancarà automàticament KA2 i quan es tanqui KA7 també es tancarà KA2. Per altra banda, els relés KA7 i KA1 no podran estar mai connectats al mateix moment.
- Els relés KA3 i KA4 connectaran la font de corrent HCLV i aniran connectats junts, és a dir, amb una senyal digital (HCLV) de la targeta d'adquisició es controlaran els dos relés.
- Els relés KA5 i KA6 connectaran la carga i aniran connectats junts, és a dir, amb una senyal digital (LOAD) de la targeta d'adquisició es controlaran els dos relés.
- Els relés KA1, KA2 i KA7 no poden estar al mateix instant de temps connectats que KA3 i KA4, ja que, la font de tensió i la font de corrent no poden estar mai connectats alhora.
- Els relés KA1, KA2 i KA7 no poden estar al mateix instant de temps connectats que KA5 i KA6, ja que, la font de tensió i la carga no poden estar mai connectats alhora.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- De les 5 entrades digitals, es controlaran 4 entrades digitals, ja que, KA2 depèn exclusivament de si KA1 o KA7 estan tancats.

2.1.2.2. Instruments i dispositius

En la següent figura es pot observar el Test Bench al complet. El Test Bench mesura 1,70m d'altura, 0,7m d'ample, 0,5m de profunditat. Es dividirà el banc de proves amb 9 parts.

- 1.- Tapa del Test Bench i SBOX.
- 2.- Panell de relés. Com es pot observar en la següent figura, el panell de relés té forma de calaix com s'ha dit anteriorment.
- 3.- Multímetre Fluke 8846 (National Instruments) que mesura la tensió d'entrada de la SBOX (Ubatt).
- 4.- Multímetre Fluke 8846 (National Instruments) que mesura la tensió que hi ha en el Shunt del panell de relés.
- 5.- Font de tensió HVLC Fluke 5500A (National Instruments).
- 6.- Font de corrent Sorensen XG 6-220 (Ametek) connectada en mode master.
- 7.- Font de corrent Sorensen XG 6-220 (Ametek) connectada en mode slave.
- 8.- Carga DC electronic Load H&H model ZS8006 (Höcherl & Hackl GmbH).
- 9.- Font de tensió auxiliar E3633A (Agilent).

Cal dir que, les dos fonts de corrent aniran connectades en mode master-slave, és a dir, es controlarà mitjançant GPIB el master però les dos fonts estaran connectades en paral·lel i, elles automàticament, aplicaran la meitat de la corrent que s'introdueix per tal de poder arribar a corrents de 400A.

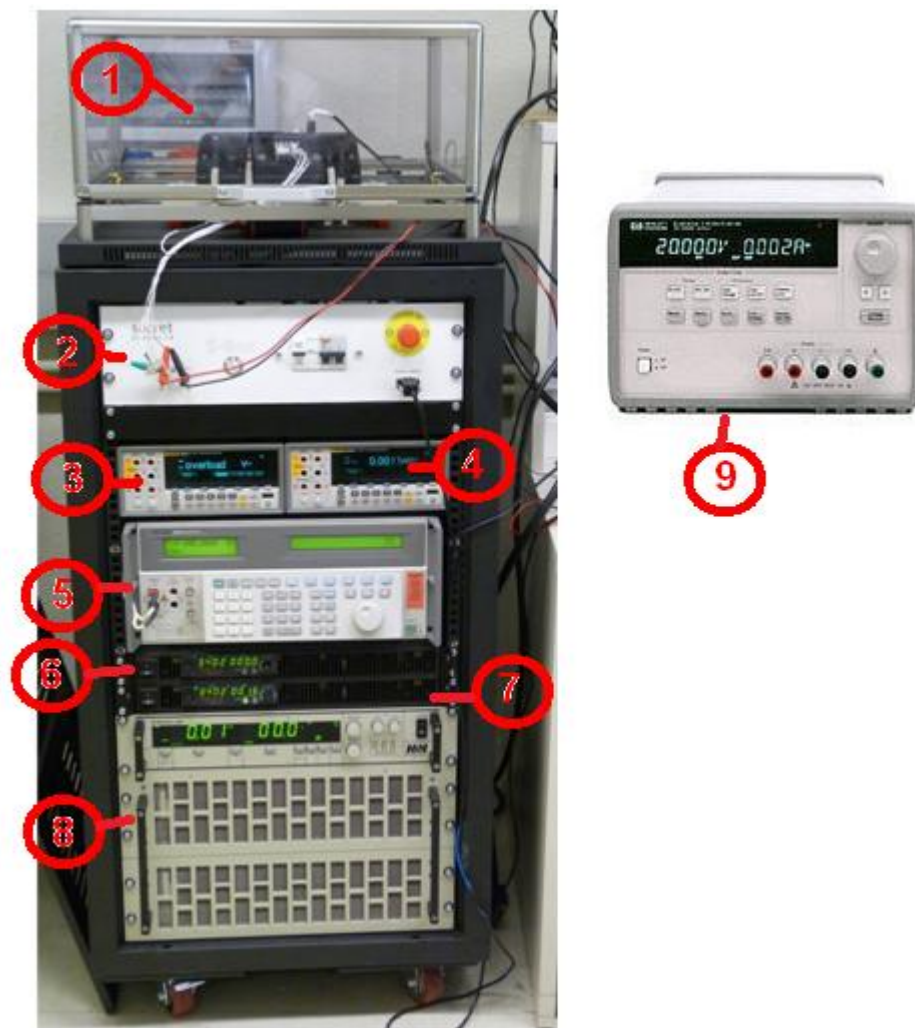


Figura 2.7 Test Bench

Per altra banda, es tindrà diferents dispositius i cablejat que no es veuen en l'anterior figura:

- Dos CANCASE XL (Vector), una amb dos canals de CAN i l'altra amb dos canals de LIN.
- Una targeta d'adquisició PCI/PXI 6220 de National Instruments
- Un cable GPIB-USB (National Instruments) per connectar els dispositius amb l'ordinador.
- Cables GPIB-GPIB per connectar els dispositius entre ells.



Figura 2.8 Can Case XL, USB-GPIB i GPIB-GPIB [8 i 14]

Les característiques i els funcionaments d'aquests dispositius estan explicats en els annexos del projecte.

2.1.2.3. Tapa de la SBOX

La SBOX s'introduirà dins de la tapa del Test Bench, es fixarà en una petita taula i es connectarà amb els connectors següents per tal que es pugi connectar amb totes les parts del circuit que s'ha citat anteriorment:

- 1.- Connector Uz k-. Connector negatiu de sortida de la SBOX.
- 2.- Connector Ubatt+. Connector positiu d'entrada de la SBOX.
- 3.- Connector de CAN. Connector CAN que uneix el dispositiu amb l'ordinador a través de les targetes CANCASEXL
- 4.- Connector Sèrie serveix per comunicar-se amb el microprocessador de forma directa. Només es realitzarà aquesta comunicació per comprovar els valors verificats però no formarà part del test de validació. És un altra manera de saber quines són les mesures del microprocessador.
- 5.- Connector Ubatt-. Connector negatiu d'entrada de la SBOX.
- 6.- Connector Uz k+. Connector positiu de sortida de la SBOX.
- 7.- Tapa de la SBOX.
- 8.- Sensor de la tapa. Detecta si està obert la tapa i obri tots els relés del panell.

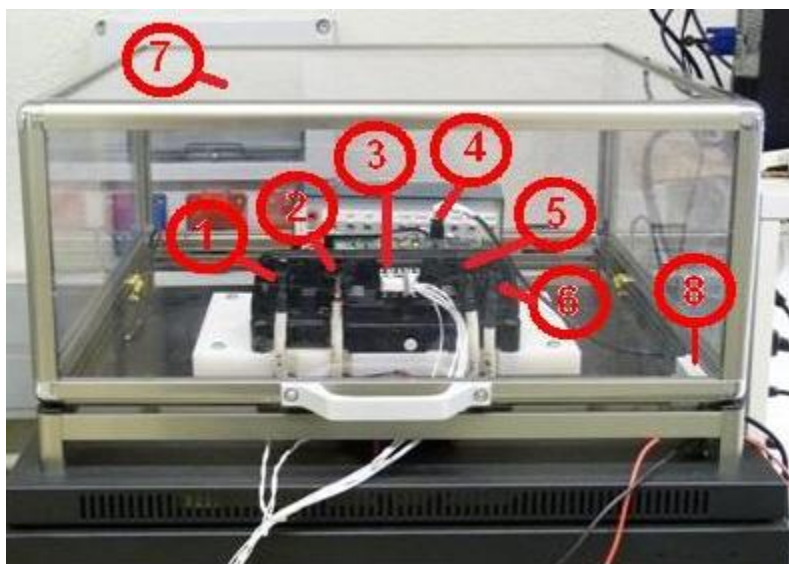


Figura 2.9 Tapa del Test Bench i connexions de la SBOX

2.2. TARGETA D'ADQUISICIÓ

La funció de la targeta adquisició PCI/PXI 6220 de National Instruments és activar i desactivar els relés del panel i els relés de la SBOX (si la SBOX és v.444) de manera automàtica, és a dir, a través de LabVIEW. Per la connexió entre la targeta i el panell de relés es necessitarà dos cables:

El cable SHC68-68-EPM que és un adaptador que es connecta al PC (connector tipus VHDCI) per tenir 68 pins a la sortida (connector tipus 68-pin 0.050 D).

Per adaptar aquest cable a la entrada del panell de relés, es necessitarà un altre adaptador perquè passi d'aquest connector de 68 pins a un connector DB9.

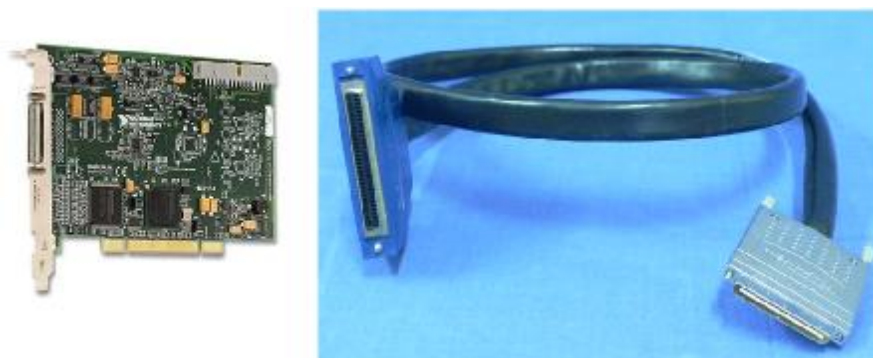


Figura 2.10 Targeta PCI/PXI 6220 i cable SHC68-68-EPM

Per l'adaptador de 68-pin 0.050 D a un DB9 s'ha cablejat els següents pins:



Figura 2.11 Adaptador pel connector DB9

- P0.0 (Pin 52- SHC68-68-EPM)=> Pin 1 (DB9 - mascle): Activa el relé KA2 del panell de relés.
- P0.1 (Pin 17 - SHC68-68-EPM)=> Pin 2 (DB9 - mascle): Activa el relé KA3 i KA4 del panell de relés, és a dir, la font de corrent.
- P0.2 (Pin 49 - SHC68-68-EPM)=> Pin 3 (DB9 - mascle): Activa el relé KA5 i KA6 del panell de relés, és a dir, la carga.
- P0.3 (Pin 47 - SHC68-68-EPM)=> Pin 4 (DB9 - mascle): Activa el relé KA7 del panell de relés, és a dir, la carga.
- P0.4 (Pin 19- SHC68-68-EPM)=> Pin 5 (DB9 - mascle): Activa el relé KA1 del panell de relés.
- PIN D GND (Pin 53 - SHC68-68-EPM)=> Pin 6 (DB9 - mascle): Pin de GND.

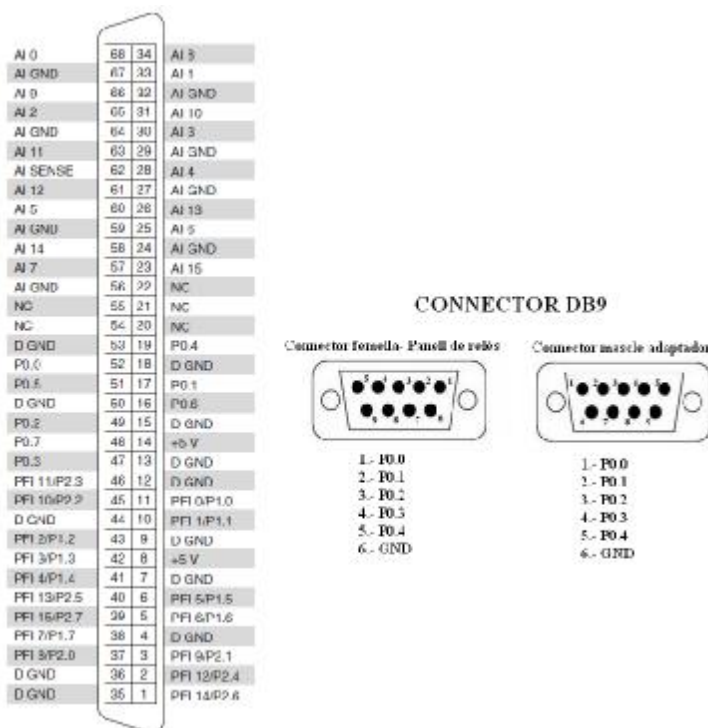


Figura 2.12 Pin out del SHC68-68-EPM i del DB9 [15]

2.2.1. CONDICIONS DE CONNEXIÓ

Per motius de seguretat, les senyals digitals que activen els relés hauran de complir unes característiques i unes condicions de connexió:

- Els relés KA1, KA2 i KA7 connectaran la font de tensió HVLC. Aniran connectats de tal manera que, a través del software, quan es tanca KA1 es tancarà automàticament KA2 i quan es tanqui KA7 també es tancarà KA2. Per altra banda, els relés KA7 i KA1 no podran estar mai connectats al mateix moment.
- Els relés KA3 i KA4 connectaran la font de corrent HCLV i aniran connectats junts, és a dir, amb una senyal digital (HCLV) de la targeta d'adquisició es controlaran els dos relés.
- Els relés KA5 i KA6 connectaran la carga i aniran connectats junts, és a dir, amb una senyal digital (LOAD) de la targeta d'adquisició es controlaran els dos relés.
- Els relés KA1, KA2 i KA7 no poden estar al mateix instant de temps connectats que KA3 i KA4, ja que, la font de tensió i la font de corrent no poden estar mai connectats alhora.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Els relés KA1, KA2 i KA7 no poden estar al mateix instant de temps connectats que KA5 i KA6, ja que, la font de tensió i la carga no poden estar mai connectats alhora.
- De les 5 entrades digitals, es controlarà 4 entrades digitals, ja que, KA2 depèn exclusivament de si KA1 o KA7 estan tancats.

A partir de les característiques i les condicions de connexió, s'ha realitzat una taula de veritat i el mètode de Karnough amb les 4 entrades per tal de que, un cop es compleixi una de les condicions es pugi començar activar els relés. D'aquesta manera s'evitarà possibles danys als dispositius del circuit, en cas de una combinació errònia del operari.

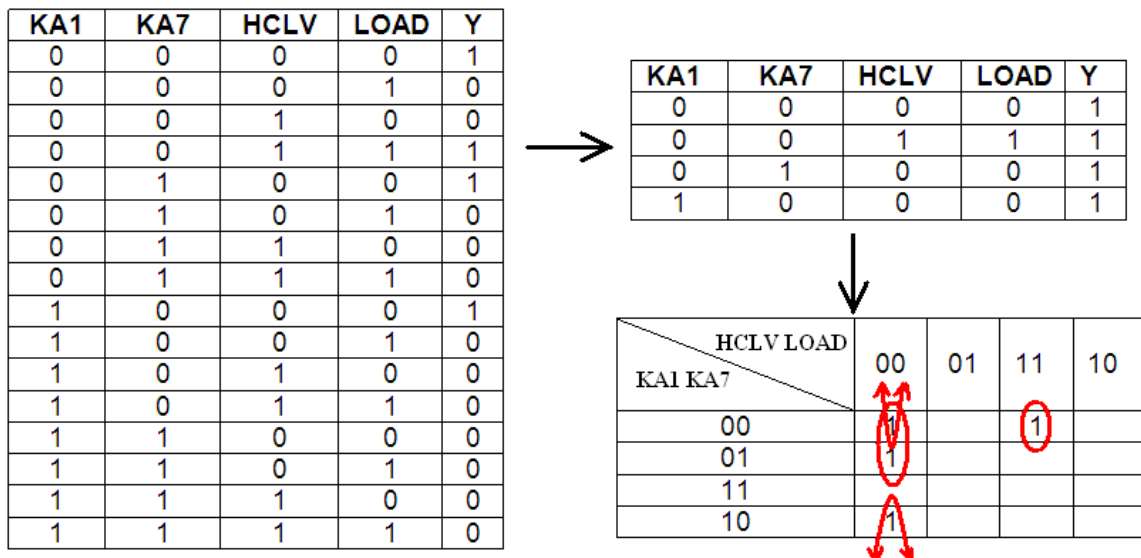


Figura 2.13 Càlculs del mètode Karnough

La sortida que complirà amb les condicions de connexió és:

$$Y = \overline{KA1} \cdot \overline{HCLV} \cdot \overline{LOAD} + \overline{KA7} \cdot \overline{HCLV} \cdot \overline{LOAD} + \overline{KA1} \cdot \overline{KA7} \cdot HCLV \cdot LOAD \quad (2-1)$$

Per la combinació booleana que es realitzarà en LabVIEW, com es pot veure en la figura, es necessitarà 4 NOTs, 2 ANDs de tres entrades, una AND de 4 entrades i una OR de tres entrades.

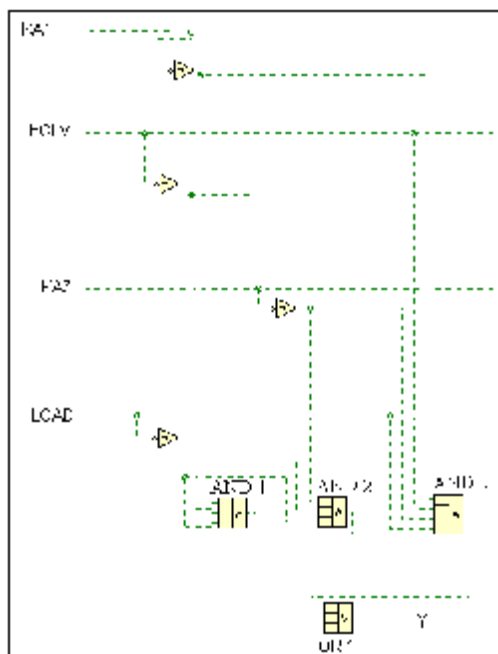


Figura 2.14 Combinació booleana de les condicions de connexió

2.3. SISTEMA DE CALIBRATGE

En aquest apartat, s'explicarà quins mètodes de calibratge s'ha seguit per tal de poder calibrar i verificar la SBOX.

En la següent figura es pot observar com és l'estructura hardware de la placa de la SBOX. El objectiu és calibrar i mesurar les tensions i les corrents dels ADUCs.

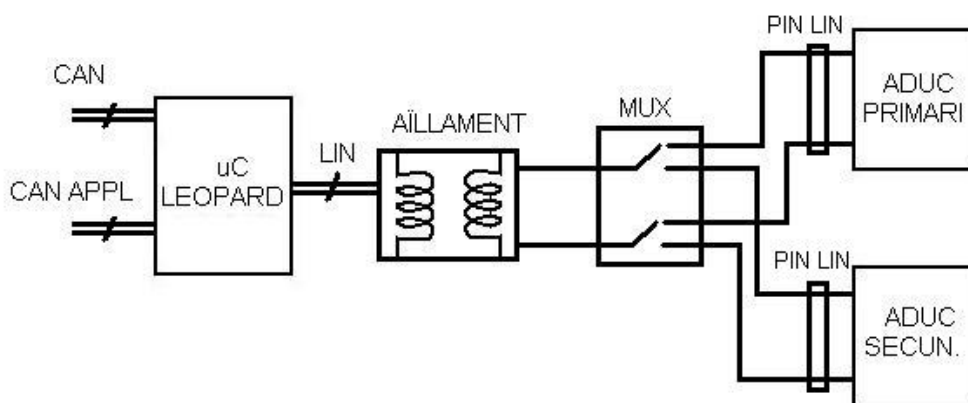


Figura 2.15 Diagrama de blocs de la placa SBOX

El esquema està format per un microprocessador Leopard que es comunica amb els dos ADUCs a través de un LIN intern. El Leopard i els ADUCs estan aïllats i mitjançant un multiplexor es selecciona un dels dos ADUCs. La comunicació amb el Leopard és a través

dels dos canals de CAN (CAN i CAN APPL) que serà una de les dos targetes CAN CASE. La comunicació amb els ADUCs és a través dels cos canals de LIN (PIN LIN) que serà l'altra targeta CAN CASE.

En el projecte, és a dir, per validar la SBOX, s'utilitzaran diferents sistemes per calibrar i mesurar tensió i corrent.

2.3.1. CALIBRATGE AMB TENSIÓ

Per calibrar amb tensió es calibrarà pel CAN APPL i es mesurarà pel CAN. El Leopard serà l'encarregat de aplicar als ADUCs els valors del guany i offset que s'han d'aplicar, ja que, inicialment els valors estaran descalibrats.

2.3.1.1. Càlcul de guany i offsets

Per realitzar el procediment per calibrar les tensions s'utilitzarà sistemes de càlcul de funcions lineals i els passos seran els següents:

- 1.- Aplicar amb el HVLC una tensió ($U_{source1}$) que serà el primer punt de mesura (PM_1).
- 2.- Realitzar la lectura per CAN o CAN APPL (U_{MEAS1}) del primer punt de mesura (PM_1) sense calibrar.
- 3.- Aplicar amb el HVLC una tensió ($U_{source2}$) que serà el segon punt de mesura (PM_2).
- 4.- Realitzar la lectura per CAN o CAN APPL (U_{MEAS2}) del segon punt de mesura (PM_2) sense calibrar.

$U_{source1}$ i $U_{source2}$ són els valors calibrats, és a dir, són els valors que es farà servir com a referència i el objectiu en aquest programa és que els valors que es mesuren per CAN (U_{MEAS1} i U_{MEAS2}), un cop calibrats, compleixin amb el marge d'error de les especificacions i s'aproximin als valors de referència.

- 5.- Calibrar U_{batt} i U_{k2} . Un cop es té els quatre valors anteriors es dibuixa dos rectes ($y(\text{recta_referència})$ i $y(\text{recta_mesura})$) on es pot observar com els valors i les rectes no són iguals, ja que, les tensions no estan calibrades. Per solucionar aquest desfase s'intentarà aplicar un guany i un offset a la recta de mesura (recta

modificada) per corregir aquest error i perquè tingui els mateixos valors que la recta de referència que és la dels valors de la font HVLC.

$$y(\text{recta_referència}) = \frac{U_{SOURCE2} - U_{SOURCE1}}{PM_2 - PM_1} x$$

$$y(\text{recta_de_mesura}) = \frac{U_{MEAS2} - U_{MEAS1}}{PM_2 - PM_1} x + b \rightarrow \quad (2-2)$$

$$\rightarrow y(\text{recta_modificada}) = \frac{U_{MEAS2} - U_{MEAS1}}{PM_2 - PM_1} \cdot x \cdot \text{Guany} + (b + \text{Offset}(mV))$$

$$y(\text{recta_referència}) = y(\text{recta_modificada})$$

Per obtenir el guany i el offset perquè les rectes siguin iguals s'haurà de realitzar els següents càlculs:

$$\text{Correcció} = \frac{U_{MEAS2} - U_{MEAS1}}{U_{SOURCE2} - U_{SOURCE1}} \rightarrow \text{Guany} = \frac{1}{\text{Correcció}} \quad (2-3)$$

Per enviar el guany que aplicarà l'ADUC per corregir l'error, s'ha d'ajustar al fons d'escala en tensió (cada bit que utilitzi el ADUC representa uns mV de tensió) i és per això que s'haurà de multiplicar pel número de comptes que hi hagi per tal de saber el guany exacte.

S'utilitza diferents bits en una Ubatt i en Uk2 perquè en Uk2 es calibra valors negatius i es necessita un bit per indicar-ho.

- Per Ubatt-> s'utilitzen 11bits -> $N^{\circ}\text{counts}=2^{11}=2048$
- Per Uk2-> s'utilitzen 10bits-> $N^{\circ}\text{counts}=2^{10}=1024$

$$\text{Guany_ajustar_fs} = \text{Guany} \cdot N^{\circ}\text{Counts} \quad (2-4)$$

Segons la teoria de les equacions de les rectes, es pot obtenir el offset de la següent manera:

$$\text{Correcció}2 = U_{MEAS1} - (U_{SOURCE1} \cdot \text{Correcció}) \rightarrow \text{Offset} = -\frac{\text{Correcció}2}{\text{Correcció}} (mV) \quad (2-5)$$

2.3.1.1.1. Calibració Ubatt

A continuació es mostra a mode de exemple, és a dir, amb valors reals, els valors dels càlculs.

$U_{SOURCE1}$ (V)	U_{MEAS1} (V)	$U_{SOURCE2}$ (V)	U_{MEAS2} (V)	Correcció	Correcció2	Guany	Guany_ajus_fs (N ^o counts=2048)	Offset (mV)
130V	147,71 V	435 V	500,83 V	0,115777	- 280,01639	8,6372	17689,17	2418, 58

Taula 2.1. Valors dels càlculs

En la següent figura es mostra en color vermell la recta de referència i en color blau la recta de les mesures de les tensions que es llegeix pel CAN. Es pot observar com a la recta de color blau se l'hi ha d'aplicar un guany (augmentar la pendent) i un offset positiu (pujar la recta) perquè sigui igual que la de color vermell.

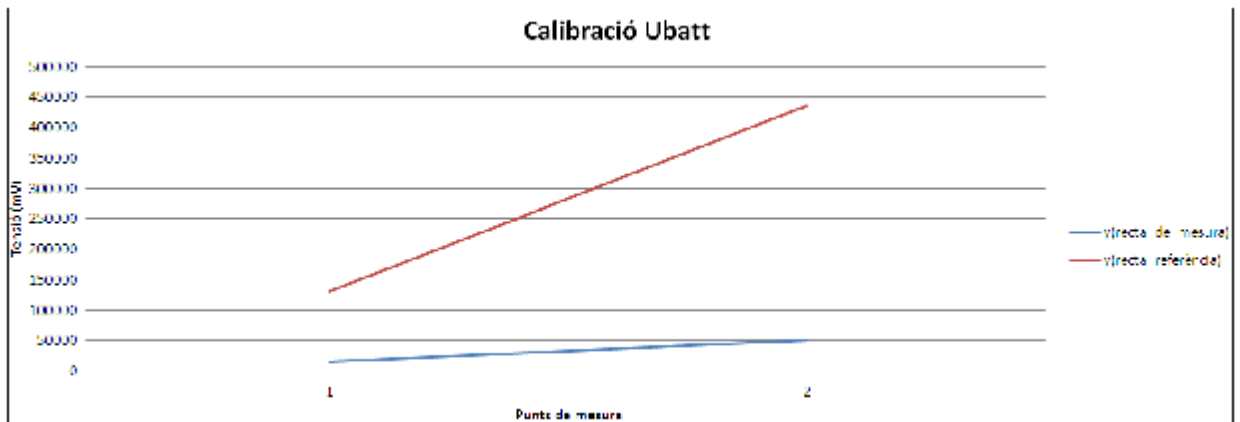


Figura 2.16 Gràfic de la calibració Ubatt amb les dos rectes

2.3.1.1.2. Calibració Uk2

A continuació es pot observar a mode de exemple, és a dir, amb valors reals, els valors dels càlculs.

$U_{SOURCE1}$ (V)	U_{MEAS1} (V)	$U_{SOURCE2}$ (V)	U_{MEAS2} (V)	Correcció	Correcció2	Guany	Guany_ajustar_fs (N ^o comptes=1024)	Offset(mV)
-100V	30,636V	435 V	60,153	0,0551	36152,3831	18,124	18559,52	-655245,1

Taula 2.2. Valors dels càlculs

En la següent figura es mostra en color vermell la recta de referència i en color blau la recta de les mesures de les tensions que es llegeix pel CAN. El objectiu és aplicar a la recta de color blau un guany (augmentar la pendent) i un offset negatiu (baixar la recta) perquè sigui igual que la de color vermell.

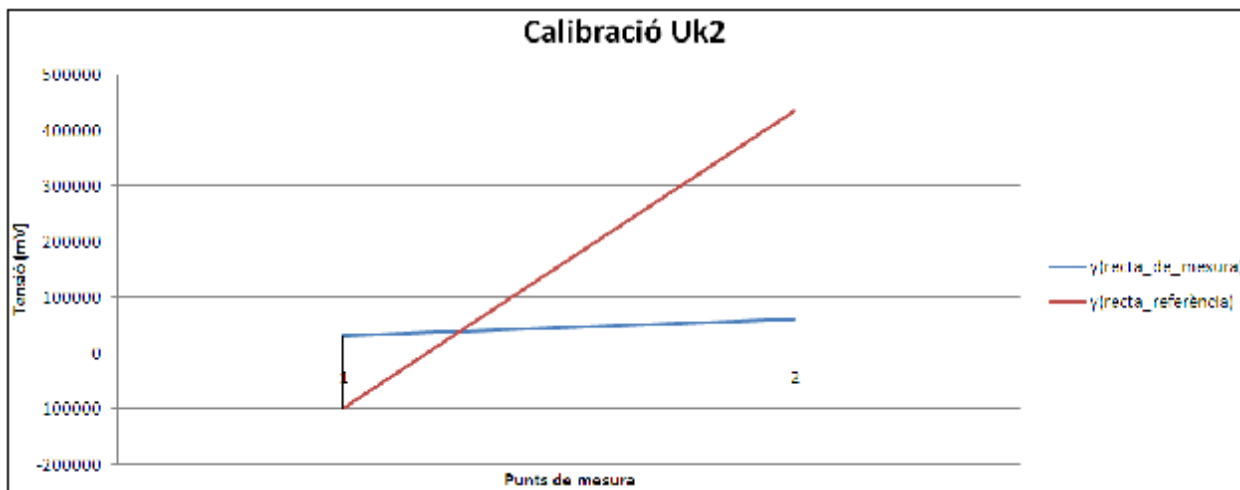


Figura 2.17 Gràfica de la calibració Uk2 amb les dos rectes

2.3.1.2. Lectura dels valors de tensió

Al software del CANoe hi haurà dos sistemes per mesurar els valors reals depenent de la tensió (U_{batt} , U_{zk} , U_q i U_{k2}) a calibrar:

- Pel canal 2 (CAN). Les lectures es faran de forma directa, a través del CAN (U_{batt} i U_{zk}).
- Pel canal 1 (CAN APPL). Un cop es té els valors reals (U_q i U_{k2}) de cada tensió dins del marge d'error especificat, s'enviarà un ordre de lectura (request) i després es llegirà el valor de senyal que correspon a la tensió real.

Un cop calibrada la recta, es realitzarà un seguit de lectures de tensió als valors especificats a les especificacions i serà les que s'enviaran al report, on es comprovarà si compleixen amb el marge d'error, i per tant, la SBOX ha estat ben calibrada.

2.3.2. CALIBRATGE AMB CORRENT

Per calibrar amb corrent es connectarà directament amb els ADUCs (sense que intervingui el Leopard) amb els pins de LIN.

El procediment de calibratge serà el següent:

- 1.- Aplicar corrent. Realitzar uns punts de mesura a diferents escales ja establerts que seran a 0A, 60A i 200A.
- 2.- Llegir corrent del LIN (per cada valor) fins que la lectura compleixi amb els errors, ja que, en aquest cas no cal aplicar guany ni offsets perquè és el mateix ADUC que fa els càlculs per tal de donar la corrent que vulgui aplicar.

Un cop calibrats els tres punts, es realitzarà un seguit de lectures de corrent als valors especificats a les especificacions i serà les que s'enviaran al report, on es comprovarà si compleixen amb el marge d'error, i per tant, es sabrà si la SBOX ha estat ben calibrada o no.

3. ESTRUCTURA SOFTWARE

En aquest apartat, s'explica com s'ha realitzat la interfície LabVIEW-CANoe per tal de mitjançant LabVIEW, controlar el programa CANoe que és el que controla tots les trames que hi ha en els busos LIN i CAN. Per altra banda, també s'explicarà el programa principal del test automàtic que controla els components del Test Bench, la SBOX i els seus subprogrames.

3.1. INTERFÍCIE LABVIEW-CANOE

Per controlar el CANoe a través del LabVIEW es necessitarà realitzar una interfície LabVIEW-CANoe. D'aquesta manera es podrà llegir les senyals que hi ha en cada un dels missatges que circulen per CAN i LIN i també escriure, és a dir, donar valor a aquestes senyals.

En aquest projecte no s'explicarà com funciona el programa CANoe per la SBOX, ja que, per temes de confidencialitat no es pot publicar i sobretot perquè és una part del projecte que no s'ha participat però en el apartat d'annex s'explicarà el funcionament del programa CANoe a mode de tutorial. En la memòria doncs, només s'explicarà els subprogrames que s'ha realitzat per tal de crear una interfície LabVIEW-CANoe.

La interfície s'explicarà mitjançant dos parts, és a dir, com es realitza l'escriptura i la lectura per CANoe a través de LabVIEW.

3.1.1. ESCRITPURA LABVIEW-CANOE

La simulació del programa tindrà la següent forma. Hi haurà la ECU que simula la SME, és a dir, la centraleta del cotxe i els dos canals de CAN on circularan els missatges de la SBOX a la SME i a la inversa.


```

void Set_Signal (void channel, void* p1)
{
  //escriptura de les variables de control
  switch(channel)
  {
    case 1:
      printf("env_ctrl_Set_Prech: p1); // estat de processa
      break;
    case 2:
      printf("env_TestComanda: p1); // estat per comprovar trames (encara no integrat)
      break;
    case 3:
      printf("env_SMS_EnableDisable_160_160s: p1); // estat per activar o desactivar els relés
      break;
    case 4:
      printf("env_SMS_StartStop_160_160s: p1); // estat per posar un valor de sincronisme en el relé de sincronisme
      break;
    case 5:
      printf("env_SMS_Stop_160_160s: p1); // estat per posar un valor de sincronisme en el relé de sincronisme
      break;
    case 6:
      printf("env_ctrl_Sema_escal: p1); // estat per calibrar els escans de les tasques i les interrupcions
      break;
    case 7:
      printf("env_ctrl_Full_Sema: p1); // estat per activar o desactivar els relés de les tasques i les interrupcions
      break;
  }
}

```

Figura 3.2 Funció Set_Signal

En el exemple de la següent figura es pot observar l'estructura del diagrama de LabVIEW. En el diagrama, el primer valor (el "p1") significa que es vol modificar la variable que pertany al valor del case de la funció del Set_Signal (per exemple: serà env_ctrl_Set_Prech si p1=1 que és una variable que modifica el estat dels relés de la SBOX). El segon valor (El "p2") significa el valor que ha de tenir aquesta variable perquè es pugui manipular els relés.

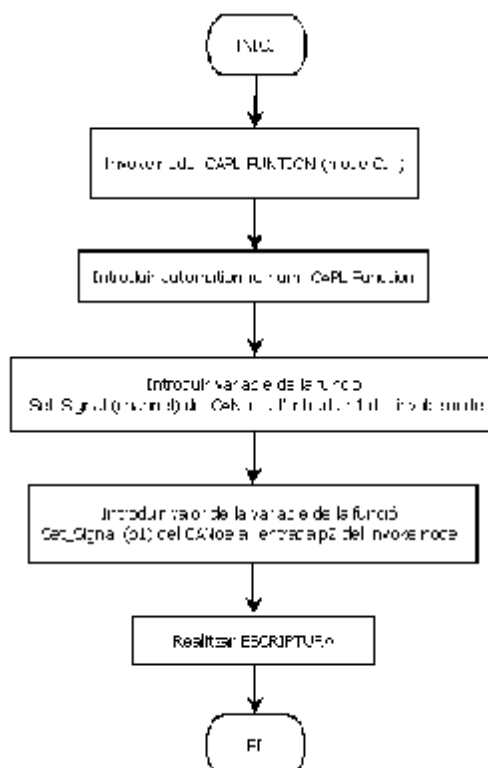


Figura 3.3 Diagrama de flux de l'escriptura de senyals

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- SendConfig(): Envia la configuració de variables de control a la SBOX a través de les senyals. Enviarà la configuració que s'ha realitzat en la funció Set_Signal.

En la següent figura es pot observar el exemple de com s'envia la senyal que pertany a al control de relés de la SBOX.

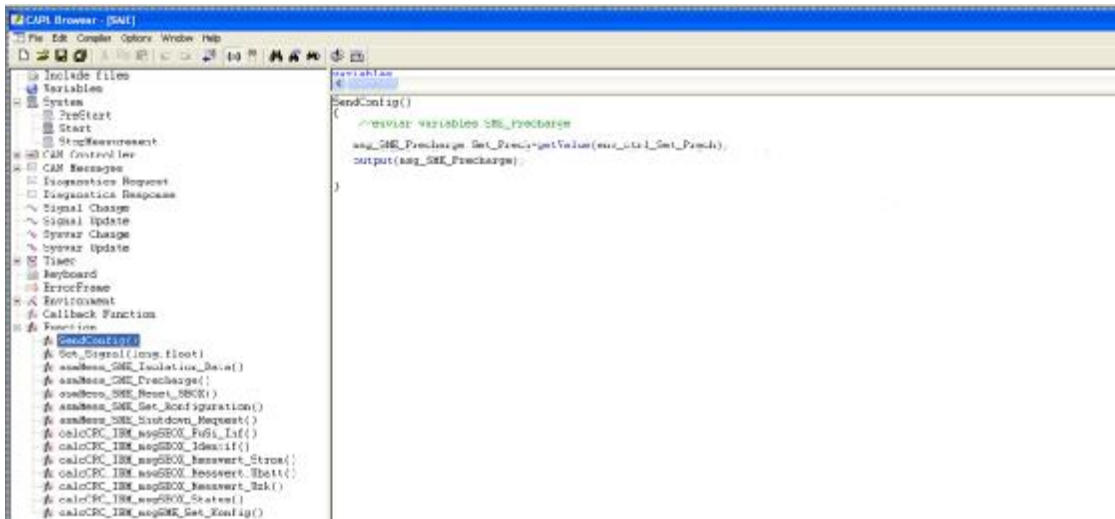


Figura 3.4 Funció SendConfig

Aquestes dos funcions es criden en un dels timers exactament en el timer msTimer100, és a dir, cada 100ms aquestes dos funcions s'executaran.

El timer tindrà la següent forma:

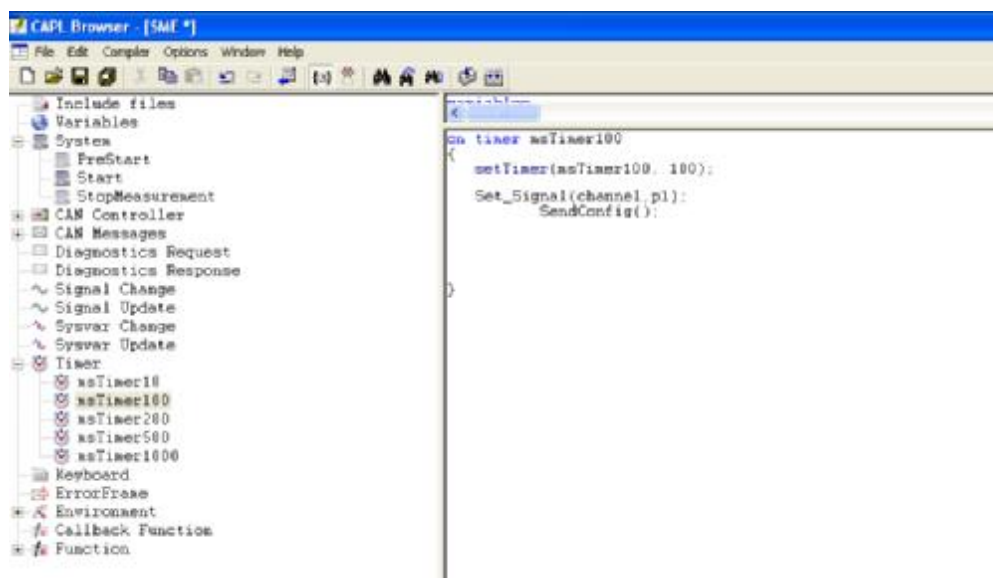


Figura 3.5 Timer 100ms

Per últim, s'haurà d'activar els timers que es fan servir en el apartat de start. S'activarà mitjançant la comanda: `setTimer(msTimer100, 100);`

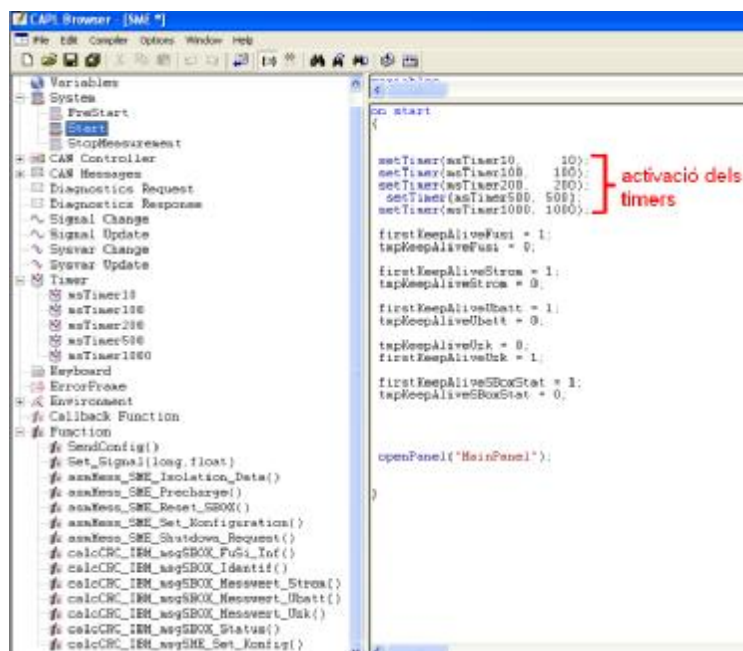


Figura 3.6 Start

3.1.2. LECTURA LABVIEW-CANOE

3.1.2.1. CAPL

La lectura de les senyals del CANoe a través del LabVIEW és molt més directe i senzill, ja que només cal fixar-se en el nom de la senyal i en el nom del missatge que pertany aquesta senyal. Els missatges i les senyals estan en el CAPL de la SME i en l'apartat de CAN Messages.

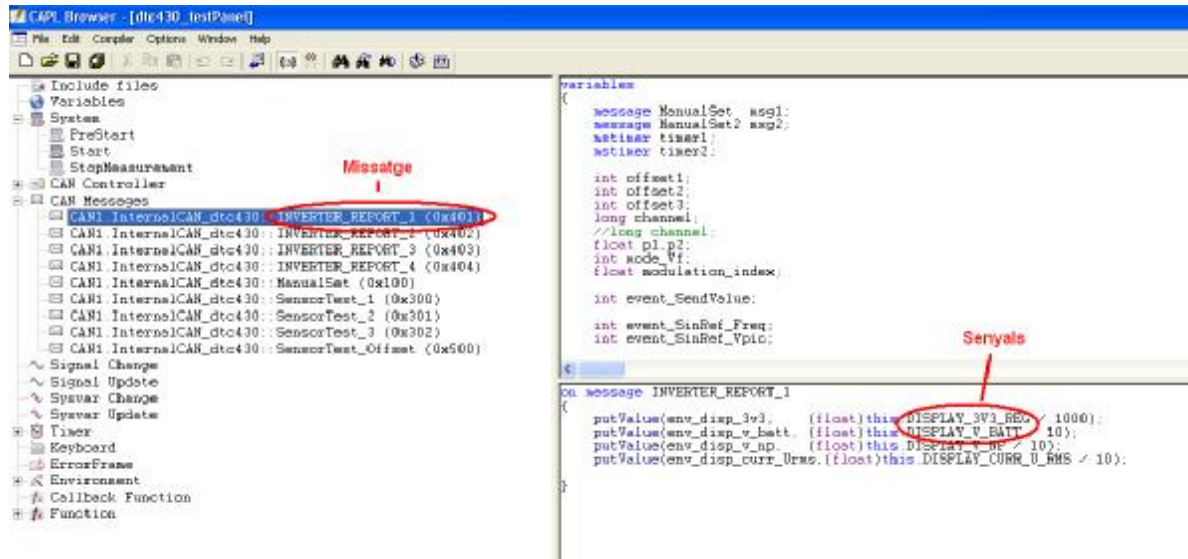


Figura 3.7 Missatgeria en el CAPL

El programa per tal de poder llegir les senyals del CANoe a través del LabVIEW és el següent:

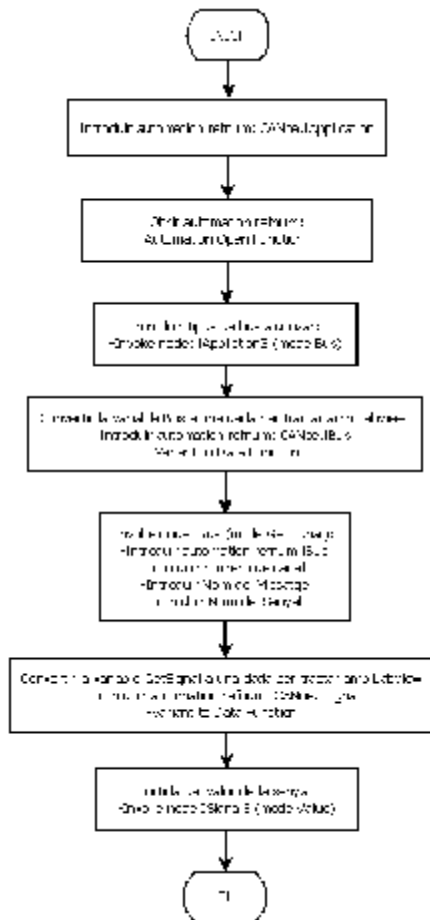


Figura 3.8 Diagrama de flux de la lectura de senyals

3.2. PROGRAMA PRINCIPAL

L'objectiu del programa com ja s'ha comentat anteriorment es comprovar si la SBOX compleix amb les especificacions marcades per BMW. Aquest programa es poden comprovar dos tipus de SBOX que s'anomenen Baukasten i v.444 on es necessiten programes diferents per cada una, ja que, el funcionament i la validació és diferent.

Per a cada validació de la SBOX, la estructura del programa principal seguirà el següent procediment:

- Primer que tot, s'introduirà dos codis que identifiquen cada peça com són el part number i el sample number.

El part number és un codi que utilitzen Lear i BMW on s'identifica quin tipus de SBOX es tracta, és a dir, si és una Baukasten o és una v.444, i quina versió és, és a dir, la peça té diferents versions fins a tenir una forma definitiva, la versió identifica l'estat actual pel que fa al hardware i el software de la peça.

El numero de prototip de la peça que és un codi específic de Lear per identificar-la.

Un cop introduït s'inicialitzarà el programa LabVIEW.

- Configurar tots els dispositius que funcionen mitjançant GPIB, és a dir, es comprovarà que estan correctament connectats i que funcionen correctament.
- Comprovar el funcionament del CANoe, és a dir, connectar el programa i mirar si les trames, és a dir, els missatges que circulen pel bus CAN i LIN entre la SBOX i la SME són les correctes.
- Comprovar que els relés interns de la SBOX, com per exemple, obrir i tancar els relés i mirar que funcionen.
- Un cop comprovat que tots els dispositius, els programes i la SBOX funcionen correctament i que no hi ha cap error es començarà el procés de calibratge i mesura.

Primer que tot, es mirarà a partir del part number, quin tipus de SBOX s'ha de validar.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Independentment del tipus de SBOX que es valida es realitzaran un seguit de proves per comprovar el consum de la SBOX que consistirà en aplicar diferents tensions amb la font auxiliar per observar el consum net de la PCB.
- En funció del tipus de SBOX es realitzaran els diferents calibratges i mesures en tensió i corrent, ja que, per cada tipus de SBOX s'ha de calibrar i mesurar tensions diferents i el funcionament varia una mica, per tant, els programes hauran de ser diferents.
- Per últim, s'introduirà totes les mesures i els codis de la SBOX (part number i sample number) en un report (un arxiu Excel). En el arxiu Excel serà on mitjançant formules Excel es comprovarà que les mesures que s'han introduït compleixen amb els marges d'error especificat per BMW.

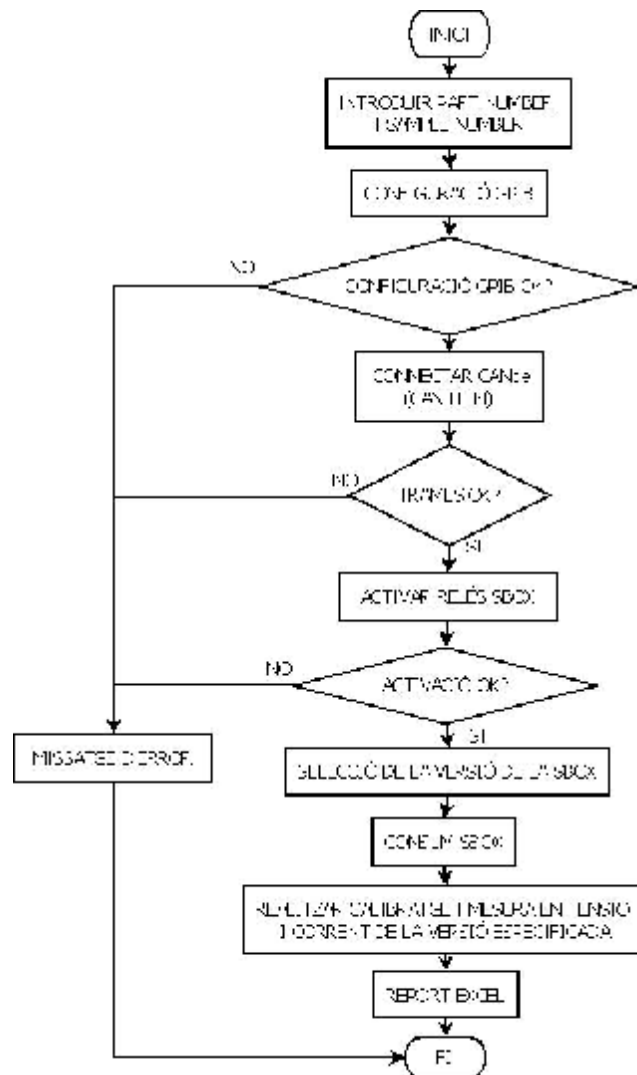


Figura 3.9 Diagrama de flux del programa principal

3.2.1. SETUP GPIB

Primer que tot, abans de començar la calibració de la SBOX, es realitzarà una configuració per comprovar de que tots els instruments que s'utilitza en aquesta calibració, estan connectats correctament i són els que realment s'han configurat.

Tots els instruments estan interconnectats entre si mitjançant busos GPIB, i per tant, es realitzarà amb LabVIEW, un programa basat exclusivament amb comandes i llibreries d'aquest tipus de comunicació anomenat VISA.

Els instruments que estan connectats mitjançant GPIB en les diferents adreces són els següents:

- 1.- Adreça 1: Multímetre Fluke 8846a que mesura la tensió Ubatt.
- 2.- Adreça 2: Multímetre Fluke 8846a que mesura la tensió que hi ha en el Shunt.
- 3.- Adreça 3: Font HVLC Fluke 5500.
- 4.- Adreça 4: Font HCLV Sorensen XG6-220.
- 5.- Adreça 5: Load H&H model ZS8006.
- 6.- Adreça 5: Font d'alimentació DC Agilent E3633A.

En el següent diagrama de flux es pot seguir de manera molt específica tots els passos per obtenir una configuració GPIB correcta però es pot simplificar en 6 passos com es realitza la configuració:

- 1.- Buscar adreça GPIB.
- 2.- Mostrar número d'adreces trobades.
- 3.- Comparar si són correctes els números d'instruments trobats.
- 4.- Buscar nom d'adreces trobades.
- 5.- Mostrar nom de cada instrument.
- 6.- Comparar si els noms són els correctes.

3.2.2. COMPROVACIÓ DE PROGRAMES

En aquest subprograma es realitzarà si els dos programes de CANoe que controlen els canals de LIN i de CAN funcionen correctament, és a dir, que hi ha les trames corresponents per cada targeta CANCE XL.

Primer que tot, s'obrirà el programa per als LINs i s'esperarà un temps de 5s per observar que el programa s'ha carregat correctament i el LabVIEW no dona cap senyal d'error. Tot seguit, es realitzarà el mateix pel programa per als CANs.

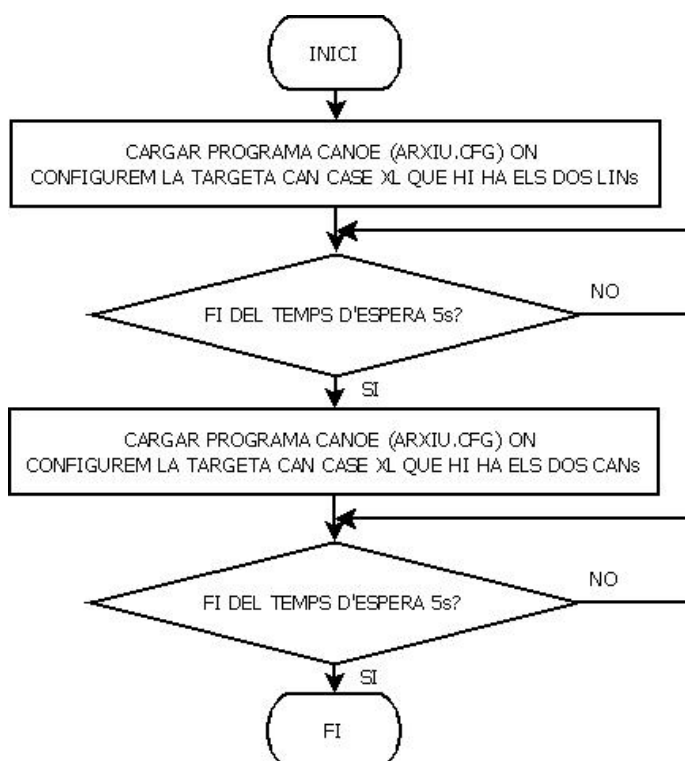


Figura 3.11 Diagrama de flux del programa comprovació dels programes de CANoe

3.2.3. COMPROVACIÓ DELS RELÉS

En aquest subprograma es realitzarà la comprovació dels relés, és a dir, s'obrirà i tancarà, un per un, els tres relés de la SBOX. En cada obertura i tancament de cada relé s'esperarà un temps prudencial de 1s. En aquest temps es podrà observar com el consum de la placa de la SBOX (ho mesura la font de tensió auxiliar) augmenta, ja que, hi ha un relé actiu i també es podrà escoltar el soroll típic de quan un relé es tanca. D'aquesta manera es podrà comprovar que els relés funcionen correctament.

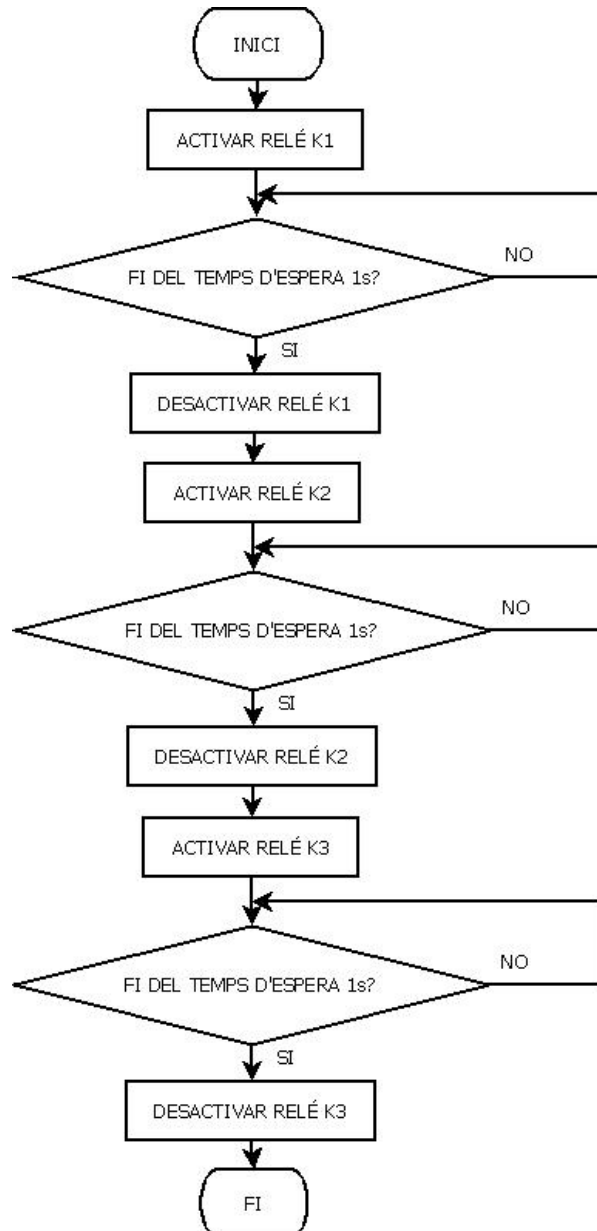


Figura 3.12 Diagrama de flux del programa comprovació del estat dels relés de la SBOX

3.2.4. SELECCIÓ TIPUS DE SBOX

Després de comprovar que tots els dispositius GPIB, i els programes CANoe funcionen correctament s'haurà de seleccionar quin tipus de SBOX s'ha de validar. A partir del part number es sap si la peça que es vol validar és una Baukasten o una v.444.

En la següent taula es mostra les diferències que afecten alhora de calibrar i mesurar una peça o una altra, ja que, el procediment canvia i com es pot observar les calibracions i les mesures que es faran seran diferents.

En parèntesi s'indica el canal del protocol de comunicació que es fa servir alhora de calibrar i mesurar la peça.

	BAUKASTEN	v.444
Control de relés	CAN	Targeta d'Adquisició
Calibració	Ubatt (CAN)	Ubatt (CAN)
	Uq (CAN APPL)	Uq (CAN)
	Uk2 (CAN APPL)	Uk2 (CAN)
	Ibatt (LIN)	Ibatt (LIN)
Mesura	Consum de la SBOX	Consum de la SBOX
	Ubatt (CAN)	Ubatt (CAN)
	Uzk (CAN)	Uq (CAN)
	Ibatt (CAN)	Uk2 (CAN)
		Ibatt (CAN)

Taula 3.1 Diferències entre Baukasten i v.444

En el següent figura es pot observar el diagrama de flux que pertany al programa selecció del tipus de SBOX.

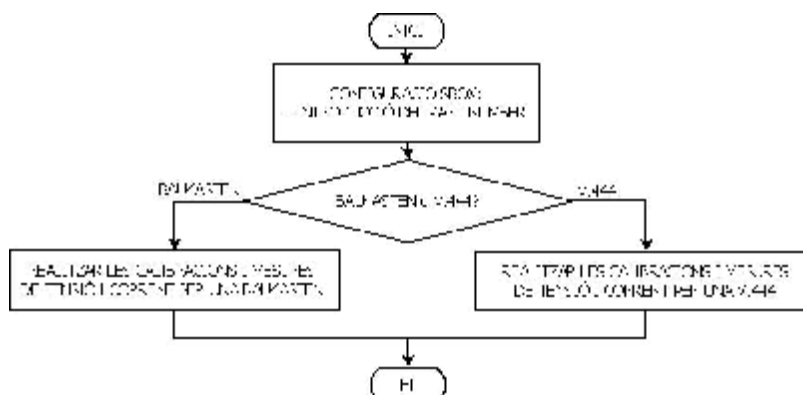


Figura 3.13 Diagrama de flux del programa selecció del tipus de SBOX

3.2.5. CONSUM DE LA SBOX

Abans de calibrar en tensió i corrent es realitza la prova de consum de la PCB de la SBOX. Aquesta prova consistirà en aplicar diferents tensions i mesurar quin és el seu consum per cada tensió, amb la font DC Agilent E3633A. Tot el procediment es realitzarà mitjançant el bus GPIB i es realitzarà sense cap font activada (només la font DC) i sense cap relé del panell i de la SBOX activat, ja que, només es vol veure el consum net de la placa. Aquests valors mesurats es reportaran al Excel, on es comprovarà que les intensitats compleixen amb les sensibilitats acordades per BMW.

Les tensions aplicades seran:

- 13,5V: <350mA.
- 5,2V: <350mA.
- 16V: <350mA.
- 18V: <350mA.

El programa de càlcul de consum de la SBOX (com es pot observar en el següent diagrama de flux) seguirà sempre el mateix procediment. S'aplicarà la tensió i es mesurarà la corrent, després s'esperarà un segon perquè s'estabilitzi la font, i així fins passar per les quatre tensions de prova. Al final, es deixarà activada la font a 12V que serà la tensió que es farà servir per realitzar totes les calibracions i mesures de corrent i de tensió.

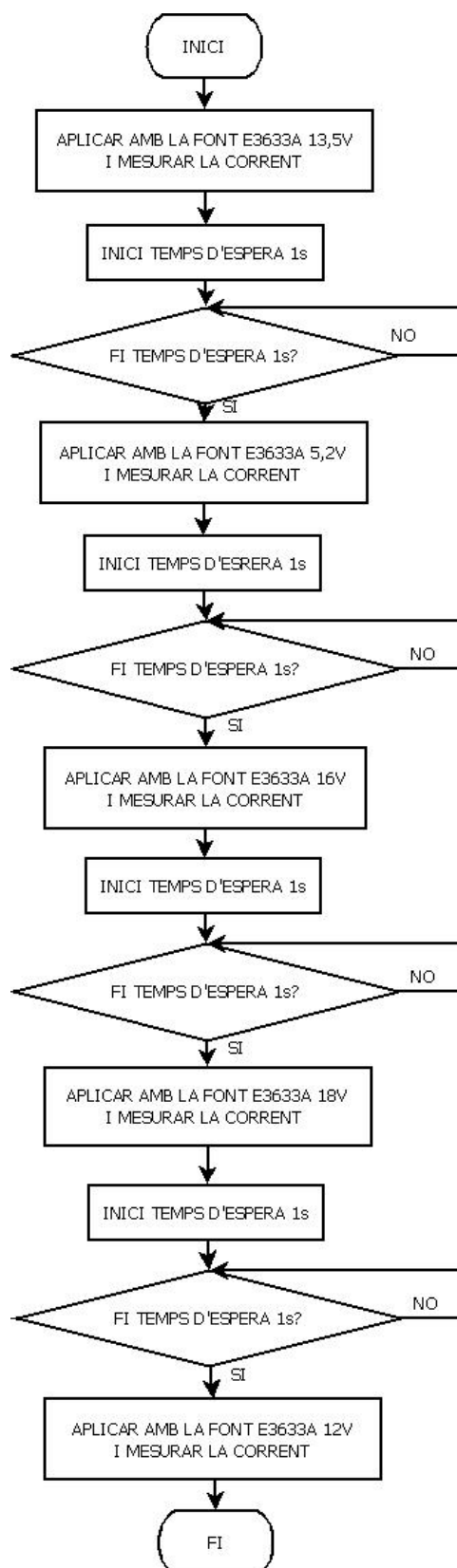


Figura 3.14 Diagrama de flux del programa consum de la SBOX

3.2.6. CALIBRACIÓ I MESURA DE TENSIONS

Per la calibració i mesura en tensió dels dos tipus de SBOX (Baukasten o v.444) s'haurà de tancar i/o obrir els relés del panell i de la SBOX, per tal de poder realitzar les calibracions i les mesures desitjades.

Les diferents configuracions de la estructura hardware que s'utilitzarà per realitzar les mesures i les calibracions de les diferents tensions seran les següents:

- Per mesurar i calibrar U_{batt} i U_q es tancarà els relés de la font de tensió (KA1 i KA2) i es tancarà el relé K1 de la SBOX.

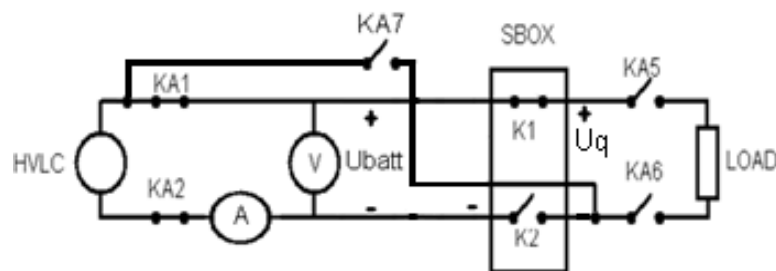


Figura 3.15 Configuració estructura hardware per mesurar i calibrar U_{batt} i U_q

- Per mesurar i calibrar U_{k2} es tancarà els relés KA2 i KA7 i es tancarà el relé K1 de la SBOX.

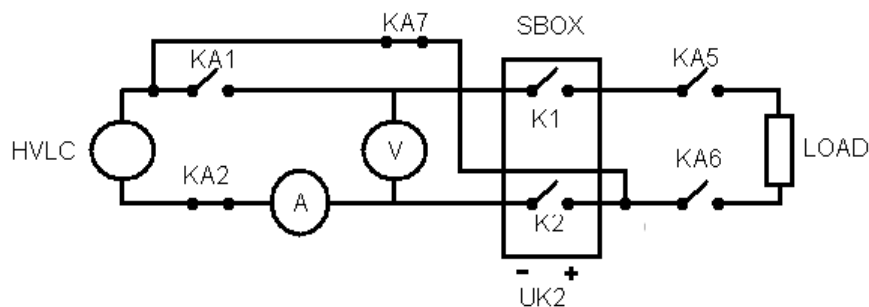


Figura 3.16 Configuració estructura hardware per mesurar i calibrar U_{k2}

- Per mesurar U_{zk} es tancarà els relés de la font de tensió (KA1 i KA2) i es tancarà els relés interns de la SBOX.

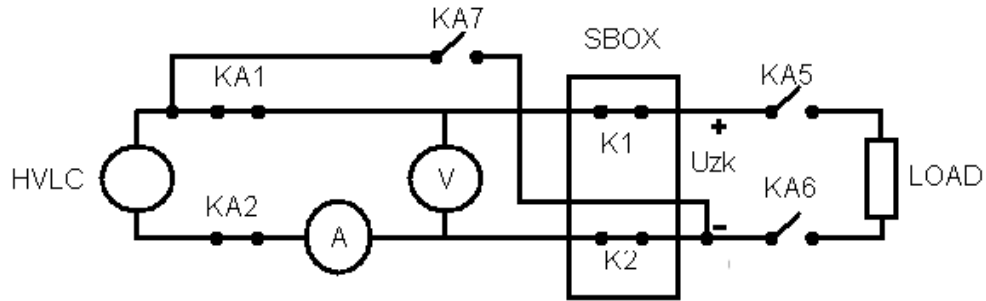


Figura 3.17 Configuració estructura hardware per mesurar U_{zk}

3.2.6.1. Calibració i mesura de tensió en una Baukasten

El programa de calibració i mesura de tensió en una Baukasten (com es pot observar en el següent diagrama de flux) seguirà sempre el mateix procediment. Un cop configurat la estructura hardware es calibrarà les tensions desitjades, es realitzarà un escalat de mesura d'aquestes tensions (U_{batt} i U_{zk}) i s'enviaran al report i així successivament fins calibrar totes les tensions que s'han de calibrar en una Baukasten (U_{batt} , U_q i U_{k2}).

Quan es configura els relés de la targeta d'adquisició i els relés de la SBOX es deixa un marge de seguretat, en forma de temps d'espera per evitar que no es configuren correctament.

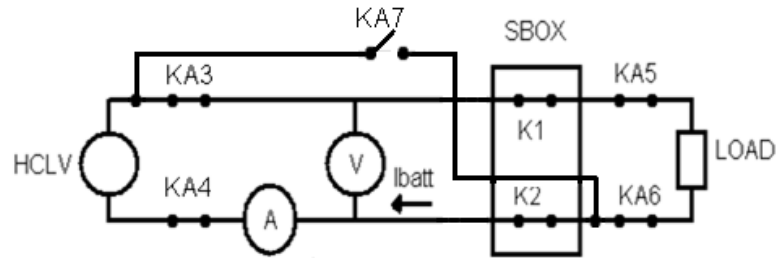


Figura 3.20 Configuració estructura hardware per mesurar i calibrar I_{batt}

3.2.7.1. Calibració i mesura de corrent en una Baukasten

La calibració de la corrent es farà per els canals de LIN (targeta CANCEASE XL LIN) i les mesures de corrent i el control de relés per CAN (targeta CANCEASE XL CAN), per tant, s'hauran d'utilitzar dos programes de CANoe, un per CAN i un per LIN. Es divideix el programa en quatre parts:

- Control de relés. Es tancarà els relés de la SBOX per la targeta de CANs, per tant, s'haurà d'obrir el programa de CANs del CANoe cada vegada que s'hagi de configurar els relés.
- Calibrar la corrent I_{batt} . Un cop configurat la estructura hardware (es configurarà amb la targeta d'adquisició els relés del panell) i obert el programa del CANoe que hi ha els dos LINS i es calibrarà la corrent en diferents punts mitjançant la targeta de LINS.
- Mesurar la corrent I_{batt} . Per últim, es tornarà a obrir el programa de CANs del CANoe i es realitzarà un escalat de mesures de corrents pel canal de CAN.
- Enviar mesures de la corrent al report Excel.

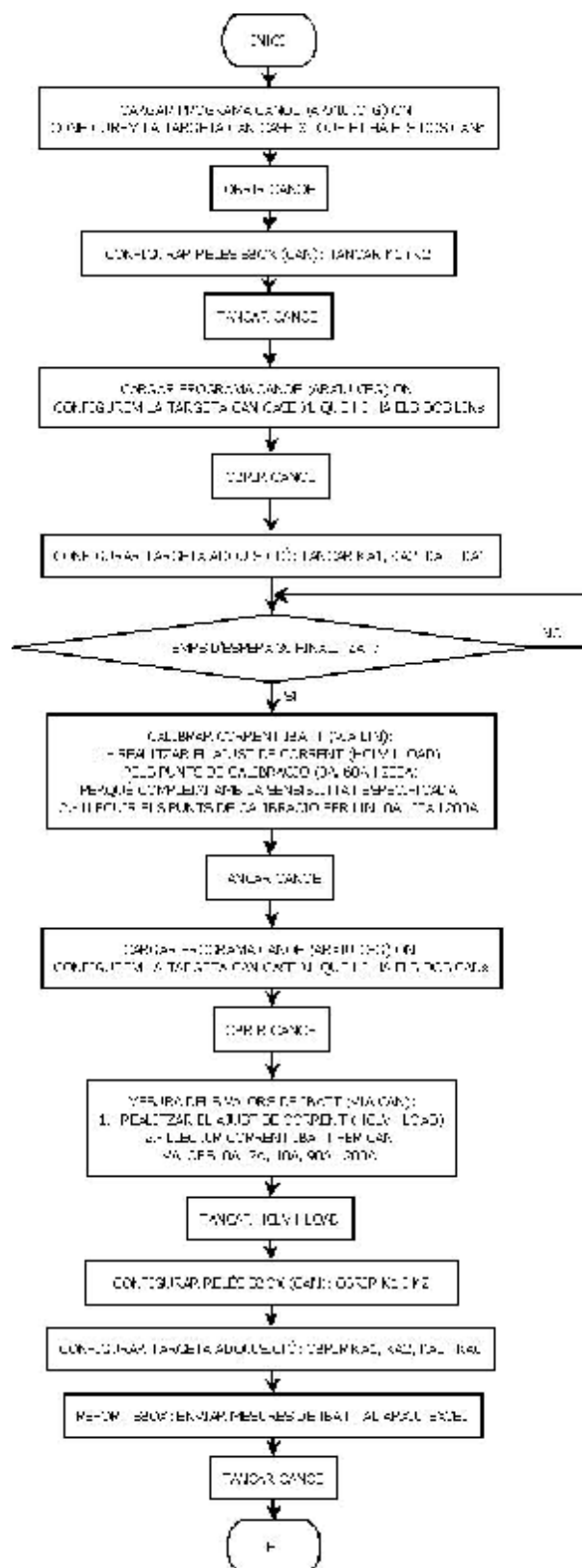


Figura 3.21 Diagrama de flux del programa calibració i mesura de corrent per una Baukasten

3.2.7.2. Calibració i mesura de corrent en una v.444

La calibració es farà per els canals de LIN i les mesures per CAN. És per aquest motiu que es carregarà dos programes de CANoe.

Un cop configurat la estructura hardware, s'esperarà un temps entre la configuració dels relés de la SBOX i els relés del panell perquè es configurin correctament i es calibrarà la corrent en diferents punts mitjançant el canal de LIN. Tot seguit s'obrirà el arxiu CANoe per realitzar l'escalat de mesures de corrents pel canal de CAN i s'enviaran al report.

Un cop acabat tot el procediment, per motius de seguretat, ja que, s'ha de deixar el circuit obert, s'obrirà tots els relés del panell i de la SBOX.

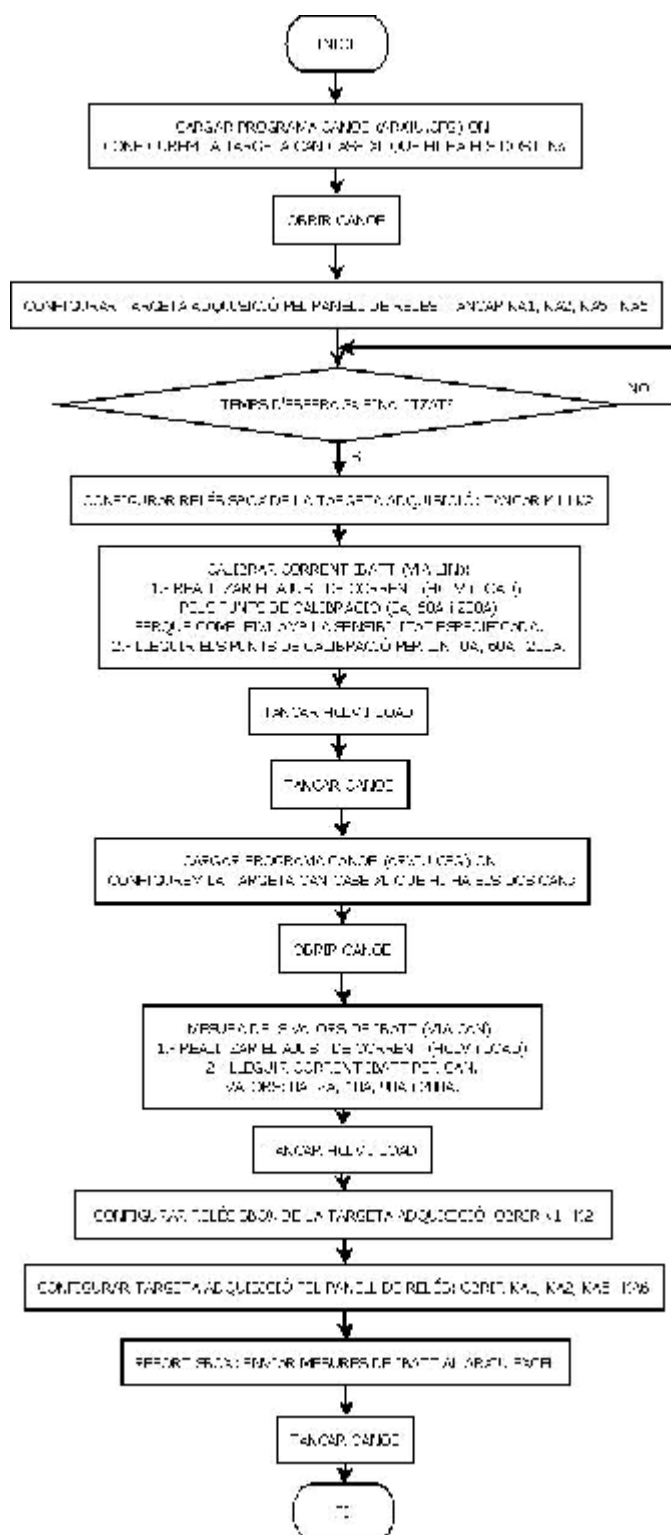


Figura 3.22 Diagrama de flux del programa calibració de corrent lbatt per una v.444

3.2.8. REPORT

Per últim, un cop finalitzat totes les calibracions es realitzarà un report, agafant totes les variables mesurades i s'introduiran a les celes especificades en un arxiu Excel. En el programa es partirà d'un arxiu Excel que servirà com a motlle i un cop finalitzat la lectura i introducció de tots els valors de tensió i corrent, es guardarà en un altre arxiu apart, conservant el arxiu que fa de motlle.

El motlle Excel serà el mateix pels dos tipus de SBOX, el que canvia és que les mesures que s'introdueixen al report són diferents, com es pot observar en la següent taula.

BAUKASTEN	V.444
Consum de la SBOX	Consum de la SBOX
Ubatt	Ubatt
Uzk	Uq
lbatt	Uk2
	lbatt

Taula 3.2 Valors que s'han d'introduir al Report

En el Excel hi ha una capçalera on hi ha tota la informació sobre la peça que s'està validant. En la següent taula es pot observar les dades que s'introdueixen a través de LabVIEW i on estan situades en el Excel.

Nom de la celda	Situació de la celda
Part Number	E6
Sample Number	H5

Taula 3.3 Valors que s'han d'introduir a la cabecera del Excel

En la següent taula es pot observar per als diferents valors que s'introdueixen al report, la cela on s'introdueix el resultat (mitjançant LabVIEW a través de CAN) i el marge d'error que ha de complir el resultat.

Valors a introduir	Resultats (Cela Excel) (mV i mA)	Marge d'error (Condicció)
Consum SBOX. Vappl=13.5V	M18	Error $\leq\pm 350\text{mV}$
Consum SBOX. Vappl=4.5V	L19	Error $\leq\pm 350\text{mV}$
Consum SBOX. Vappl=16V	L19	Error $\leq\pm 350\text{mV}$
Consum SBOX. Vappl=18V	L19	Error $\leq\pm 350\text{mV}$
Ibatt=0A	K21	Error $\leq\pm 20\text{mA}$
Ibatt=2A	K23	Error $\leq\pm 20\text{mA}$
Ibatt=10A	K24	Error $\leq\pm 100\text{mA}$
Ibatt=90A	K25	Error $\leq\pm 900\text{mA}$
Ibatt=200A	K26	Error $\leq\pm 2\text{A}$
Ubatt=100V	K32	Error $\leq\pm 1\text{V}$
Ubatt=130V	K33	Error $\leq\pm 0.265\text{V}$ (Baukasten) Error $\leq\pm 0.320\text{V}$ (v.444)
Ubatt=434.5V	K34	Error $\leq\pm 0.265\text{V}$ (Baukasten)

		Error $\leq\pm 0.320V$ (v.444)
U _{batt} =450V	K35	Error $\leq\pm 1V$
U _{zk} =0V	K37	Error $\leq\pm 1V$
U _{zk} =100V	K38	Error $\leq\pm 1V$
U _{zk} =130V	K39	Error $\leq\pm 0.265V$ (Baukasten) Error $\leq\pm 0.320V$ (v.444)
U _{zk} =434.5V	K40	Error $\leq\pm 0.265V$ (Baukasten) Error $\leq\pm 0.320V$ (v.444)
U _{zk} =450V	K41	Error $\leq\pm 1V$
U _q =-225V	K42	Error $\leq\pm 0.320V$
U _q =-100V	K43	Error $\leq\pm 1V$
U _q =100V	K45	Error $\leq\pm 1V$
U _q =270V	K46	Error $\leq\pm 0.320V$
U _q =380V	K47	Error $\leq\pm 0.320V$
U _q =450V	K48	Error $\leq\pm 1V$
U _{k2} =-225V	K49	Error $\leq\pm 0.320V$
U _{k2} =-100V	K50	Error $\leq\pm 1V$
U _{k2} =0V	K51	Error $\leq\pm 1V$
U _{k2} =100V	K52	Error $\leq\pm 1V$
U _{k2} =270V	K53	Error $\leq\pm 0.320V$

Uk2=380V	K54	Error $\leq\pm 0.320V$
Uk2=450V	K55	Error $\leq\pm 1V$

Taula 3.4 Valors que s'han d'introduir al Report

En la següent taula es pot observar per als diferents valors que s'introdueixen al report, la formula per obtenir el error absolut en mV (un cop introduït el resultat per CAN), la seva cela en el Excel, la formula per obtenir si el error que hi ha realment compleix o no amb les especificacions i la seva cela en el Excel.

Valors a introduir	Valor absolut del error en mV	Cela d'error (mV i mA)	Condió de compliment de les especificacions	Cela de compliment
Consum SBOX. Vappl=13.5V	-	-	=SI(M18<=350;"OK";"NOK")	I18
Consum SBOX. Vappl=4.5V	-	-	-	I19
Consum SBOX. Vappl=16V	-	-	-	I19
Consum SBOX. Vappl=18V	-	-	-	I19
Ibatt=0A	=0-K21	M21	=SI(ABS(M21)<=20;"OK";"NOK")	I21
Ibatt=2A	=2000-K23	M23	=SI(ABS(M23)<=20;"OK";"OK, RESTRICTED")	I23
Ibatt=10A	=10000-K24	M24	=SI(ABS(M24)<=100;"OK";"NOK")	I24

Ibatt=90A	=90000-K25	M25	=SI(ABS(M25)<=100;"OK";"NOK")	I25
Ibatt=200A	=200000-K26	M26	=SI(ABS(M24)<=100;"OK";"NOK")	I26
Ubatt=100V	=100000-K32	M32	=SI(ABS(M32)<=1000;"OK";"NO K")	I32
Ubatt=130V	=130000-K33	M33	=SI(\$E\$6=4611419;SI(ABS(M33))<=320;"OK";"NOK");SI(ABS(M3 3)<=256;"OK";"NOK"))	I33
Ubatt=434.5 V	=434500-K34	M34	=SI(\$E\$6=4611419;SI(ABS(M34))<=320;"OK";"NOK");SI(ABS(M3 4)<=256;"OK";"NOK"))	I34
Ubatt=450V	=450000-K35	M35	=SI(ABS(M35)<=1000;"OK";"NO K")	I35
Uzk=0V	=SI(\$H\$37="<N/A >";"";(0-37))	M37	=SI(\$E\$6=4611419;"<N/A>";SI(M37<=1000;"OK";"NOK"))	I37
Uzk=100V	=SI(\$H\$37="<N/A >";"";(0-38))	M38	=SI(\$E\$6=4611419;"<N/A>";SI(M38<=1000;"OK";"NOK"))	I38
Uzk=130V	=SI(\$H\$37="<N/A >";"";(0-39))	M39	=SI(\$E\$6=4611419;"<N/A>";SI(M39<=265;"OK";"NOK"))	I39
Uzk=434.5V	=SI(\$H\$37="<N/A >";"";(0-40))	M40	=SI(\$E\$6=4611419;"<N/A>";SI(M40<=265;"OK";"NOK"))	I40
Uzk=450V	=SI(\$H\$37="<N/A >";"";(0-41))	M41	=SI(\$E\$6=4611419;"<N/A>";SI(M41<=265;"OK";"NOK"))	I41
Uq=-225V	=-225000-K42	M42	=SI(\$E\$6=4611419;SI(ABS(M42))<=1000;"OK";"<N/A>");"NOK")	I42
Uq=-100V	=-100000-K43	M43	=SI(\$E\$6=4611419;SI(ABS(M43	I43

)<=1000;"OK";"<N/A>");"NOK")	
Uq=100V	=100000-K45	M45	=SI(\$E\$6=4611419;SI(ABS(M45))<=1000;"OK";"NOK");"NOK")	I45
Uq=270V	=270000-K46	M46	=SI(\$E\$6=4611419;SI(ABS(M46))<=320;"OK";"NOK");"NOK")	I46
Uq=380V	=380000-K47	M47	=SI(\$E\$6=4611419;SI(ABS(M47))<=320;"OK";"NOK");"NOK")	I47
Uq=450V	=450000-K48	M48	=SI(\$E\$6=4611419;SI(ABS(M48))<=1000;"OK";"NOK");"NOK")	I48
Uk2=-225V	=-225000-K49	M49	=SI(\$E\$6=4611419;SI(ABS(M49))<=320;"OK";"NOK");"<N/A>")	I49
Uk2=-100V	=-100000-K50	M50	=SI(\$E\$6=4611419;SI(ABS(M50))<=1000;"OK";"NOK");"<N/A>")	I50
Uk2=0V	=0-K51	M51	=SI(\$E\$6=4611419;SI(ABS(M51))<=1000;"OK";"NOK");"<N/A>")	I51
Uk2=100V	=100000-K52	M52	=SI(\$E\$6=4611419;SI(ABS(M52))<=1000;"OK";"NOK");"<N/A>")	I52
Uk2=270V	=270000-K53	M53	=SI(\$E\$6=4611419;SI(ABS(M53))<=320;"OK";"NOK");"<N/A>")	I53
Uk2=380V	=380000-K54	M54	=SI(\$E\$6=4611419;SI(ABS(M54))<=320;"OK";"NOK");"<N/A>")	I54
Uk2=450V	=450000-K55	M55	=SI(\$E\$6=4611419;SI(ABS(M55))<=1000;"OK";"NOK");"<N/A>")	I55

Taula 3.5 Valors que s'han d'introduir al Report

En les tres anteriors taules s'ha pogut observar les dades que es treballa en el projecte però en el report hi ha més dades i més especificacions que s'han de complir per tal de validar cadascuna de les SBOX.

El diagrama de flux que fa referència al programa Report és el següent. Es seguirà tres passos:

- Inicialitzar els drivers corresponents i obrir el arxiu que farà de motlle (s'haurà de ficar l'adreça on es troba l'arxiu).
- Realitzar uns bucles que estaran escrivint en el report nou, fins que totes les dades ja s'hagin introduït en el Excel. S'introduirà diferents tensions i la corrent depenén de si es Baukasten o v.444.
- Un cop estiguin totes les dades apuntades es guardar l'arxiu on el nom serà la SBOX més el numero de prototip del dispositiu.

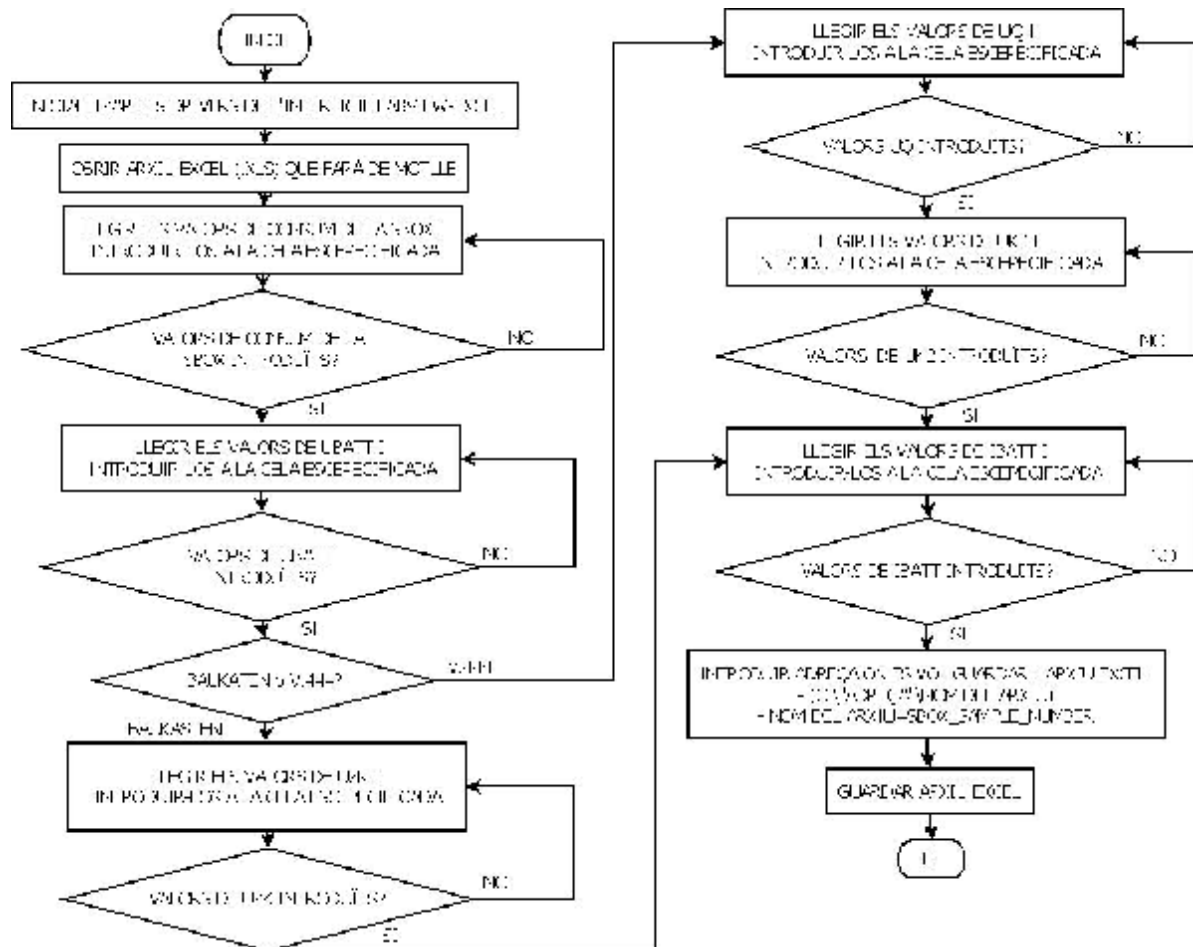


Figura 3.23 Diagrama de flux del programa Report

3.2.9. ALTRES PROGRAMES

En aquest apartat s'explicarà els subprogrames més important que s'han fet servir en el programa principal, com ara, el control de fonts i el control de la targeta d'adquisició.

3.2.9.1. Control de les fonts

Per realitzar el control de les fonts es realitzarà mitjançant dos subprogrames, un per controlar la font de tensió (HVLC) i l'altre per controlar la font de corrent (HCLV). En els dos programes s'utilitzarà el bus GPIB per configurar els dispositius.

3.2.9.1.1. Control de la font HCLV

Per controlar les fonts de HCLV s'utilitzarà també la carga que es configurarà en mode corrent, és dir, se l'hi aplicarà corrent i automaticament es configurarà amb el valor en ohms que faci falta perquè pugui passar la corrent introduïda.

En aquest cas, la font HCLV es controlarà indirectament, ja que, s'introduirà la corrent que es vol que passi per la carga i de forma automàtica, les dos fonts de corrent s'ajustaran i aplicaran la meitat de la corrent. Com que la font de corrent i la carga no són dos dispositius molt precisos, s'utilitzarà el multímetre i d'un programa LabVIEW per assegurar de que la corrent que passa és l'especificada i pot complir amb el marge d'error especificat.

El programa LabVIEW per controlar la HCLV, la carga i el multímetre que controla el Shunt és el següent:

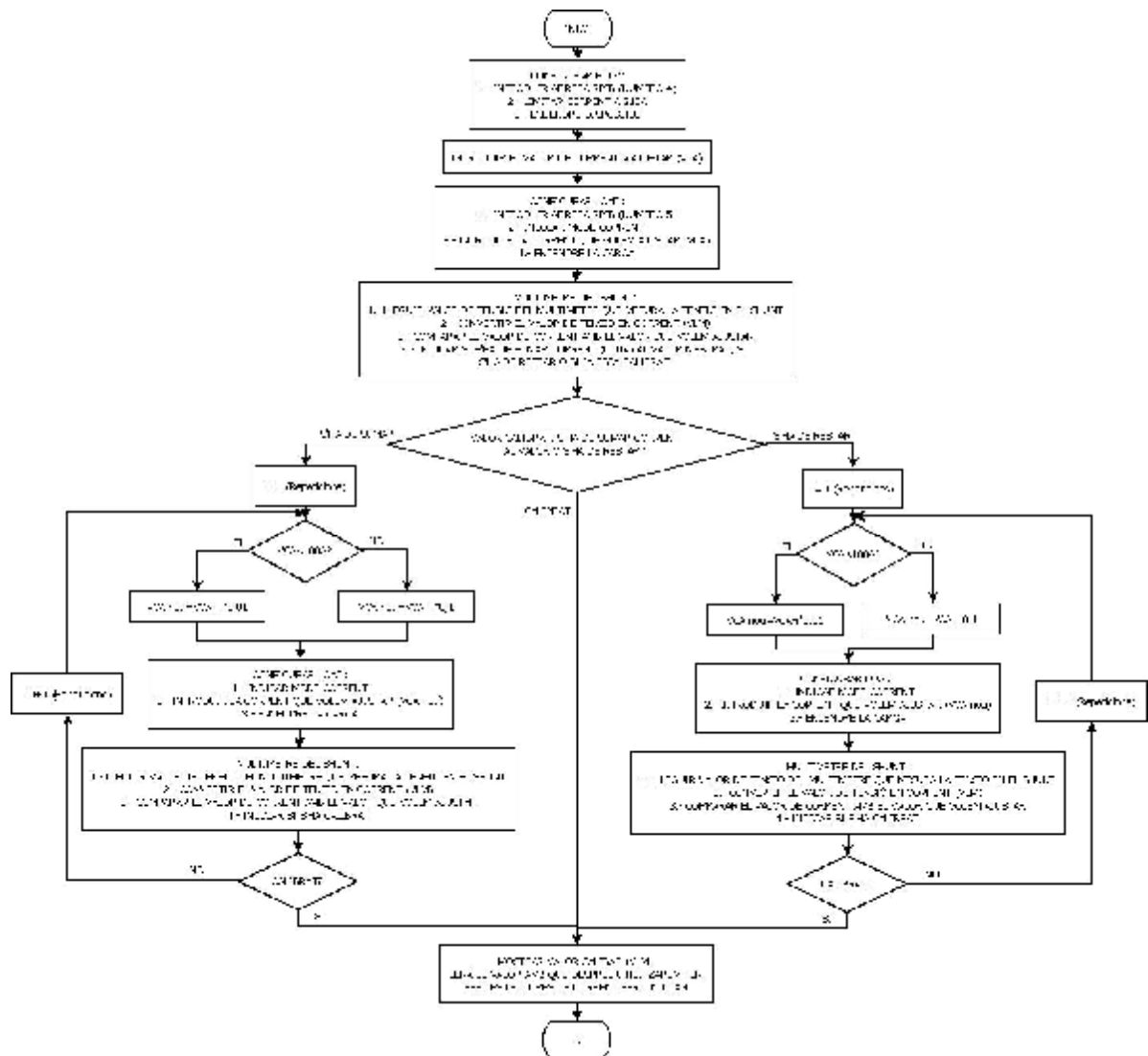


Figura 3.24 Diagrama de flux del programa control de HCLV

3.2.9.1.2. Control de la font de tensió

El control de la font de tensió HVLC es farà de manera directa, és a dir, sense multímetres que mesurin la tensió aplicada i la carga, ja que, la font de tensió és suficientment precisa, ja que, té un error de micro parts per milió (u.p.m).

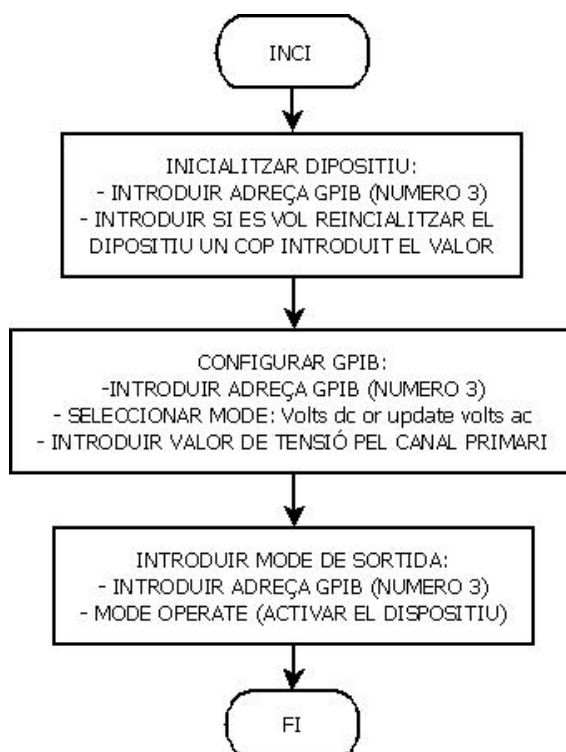


Figura 3.25 Diagrama de flux del programa control de HVLC

3.2.9.1.3. Control de la font auxiliar

Per controlar la font auxiliar (E3633A) que alimentarà a la placa de la SBOX i als relés del panell es seguirà el següent diagrama de flux.

Primer que tot, s'introduirà l'adreça GPIB que s'ha configurat, el canal del dispositiu, la tensió a aplicar i s'activarà la sortida per poder alimentar la placa i els relés. En mode de seguretat es seleccionarà el límit de corrent que es deixa passar en aquest cas es deixarà en 1A. Per últim, es mesura la tensió i la corrent que hi ha.

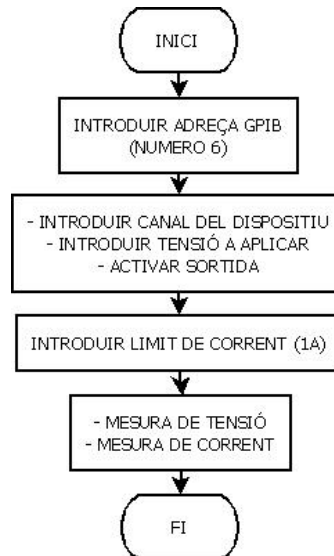


Figura 3.26 Diagrama de flux del programa control de la font auxiliary

3.2.9.1.4. Programa de la targeta d'adquisició

En el següent diagrama de flux es pot observar de forma més clara com es realitza les condicions de connexió i, com i quan, s'activen els relés.

Per evitar que el programa es quedi en el bucle de forma indefinida, en cas de no introduir una combinació booleana correcta, s'ha afegit un altra condició que comprovarà varies vegades les condicions de connexió dels relés, i si no compleix amb la condició, quan ja s'han fet tots els intents, es sortirà del bucle i es posarà totes les sortides de la targeta d'adquisició a zero. Per altra banda, entre una activació de relé i una altra, es realitzarà un temps d'espera per donar temps als relés a tancar-se o a obrir-se.

A continuació es seguirà els passos per a realitzar aquest projecte i també el executable:

- 1.- Crear un nou projecte: Obrir el LabVIEW i és prem sobre "Empty Project".
- 2.- Un cop dins del projecte s'afegeix el arxiu .vi on hi ha tot el programa principal i el arxiu .vi on hi ha totes les variables global d'aquest programa principal. Un arxiu s'afegeix polsant amb el botó dret damunt de "My Computer" i seguint els passos que es pot observar en la següent figura.

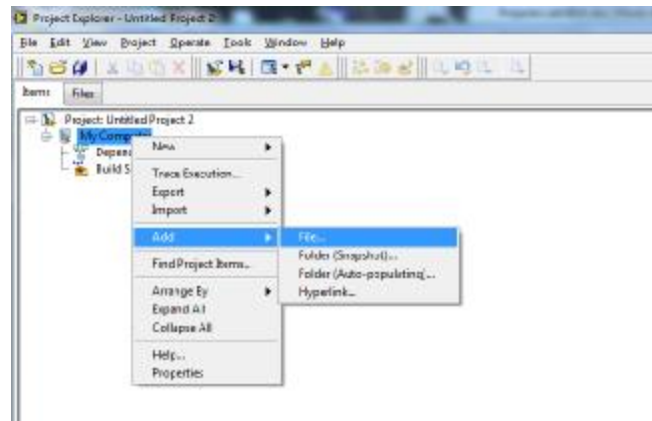


Figura 3.28 Procediment per afegir un arxiu .vi

- 3.- Un cop afegit aquests dos arxius, directament també s'afegiran tots els subprogrames del programa principal. Com es pot observar a la figura el projecte tindrà la següent forma.

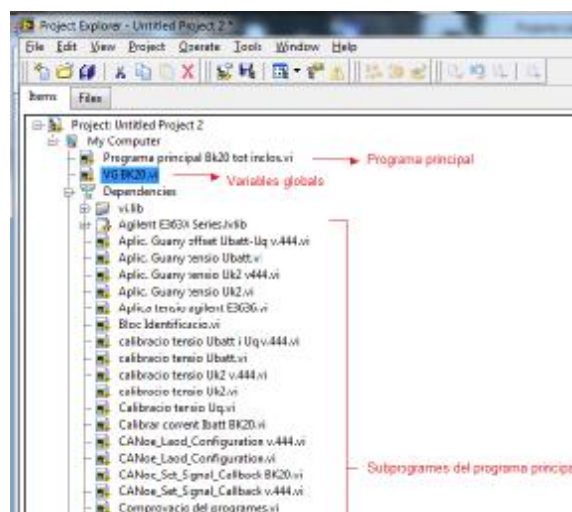


Figura 3.29 Vista de tot el projecte

4.- Quan s'ha acabat d'afegir tots els arxius es hora de realitzar el executable (.exe). S'ha de prémer amb el botó dret damunt de "Build Specifications" i seguint els passos que es pot observar en la següent figura.

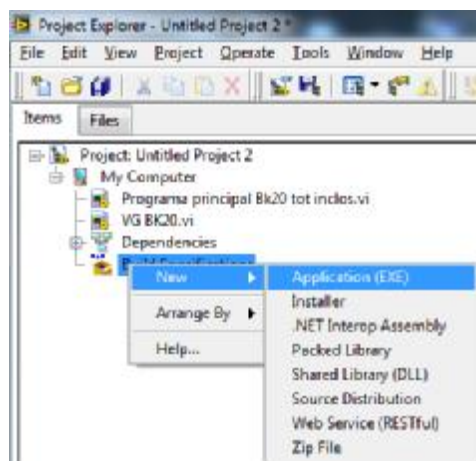


Figura 3.30 Procediment per crear un executable

Cal dir que, s'hi no s'ha guardat el projecte, LabVIEW obliga a fer-ho.

5.- En aquest apartat, s'ompli les propietats del executable, com per exemple, el nom del executable, el nom de la carpeta on es guardarà i la direcció on es vol guardar-lo entre d'altres propietats.

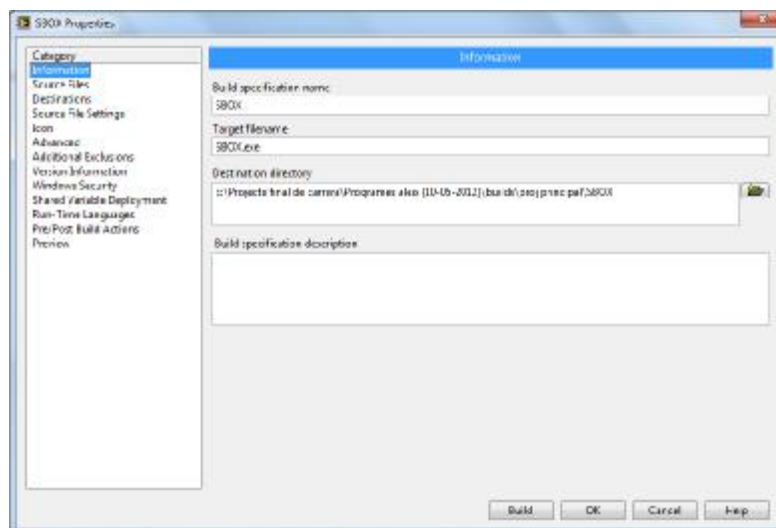


Figura 3.31 Propietats del executable

6.- S'afegeix el arxiu .vi que formarà part del executable al "Startup VIs" seleccionant l'arxiu i la fletxa de direcció a la dreta en l'apartat de "Source Files". En aquest cas

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

com que les úniques variables de control estan en l'arxiu de les variables globals, l'únic arxiu que formarà part del executable és el de les variables globals.

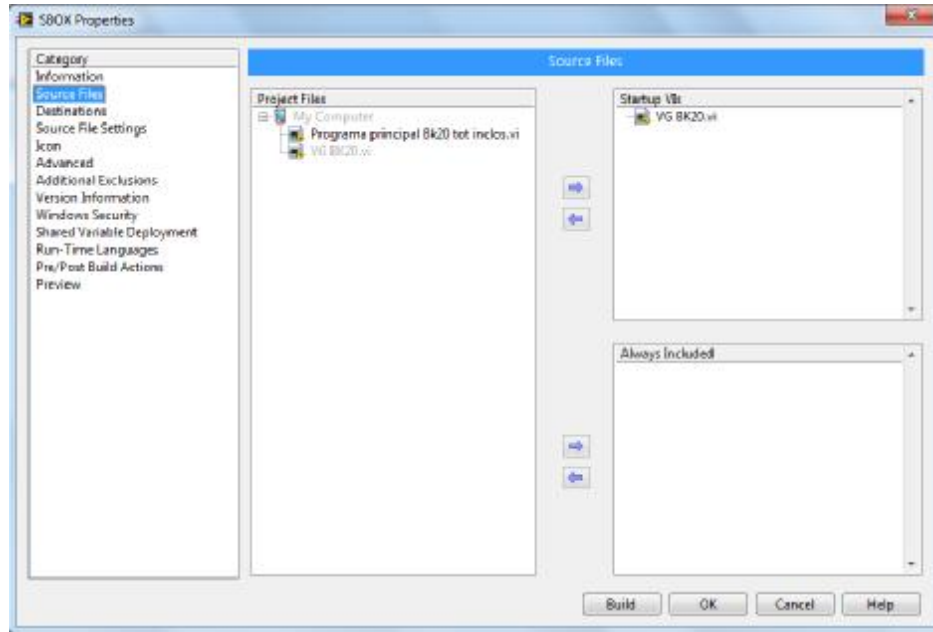


Figura 3.32 Propietats del executable

7.- Per últim, un cop creat l'executable és prem al botó dret i a "Build" i ja es podrà utilitzar-lo.

Existeixen dos maneres de fer funcionar l'executable, polsant el botó dret i Run, i també buscant la carpeta on s'ha guardat l'executable i polsant a l'executable. En l'últim cas no faria falta obrir el projecte ni el programa LabVIEW que és el que realment es busca, és a dir, que el operari només realitzi el doble clic al executable i que el programa ja funcioni sense obrir el LabVIEW.

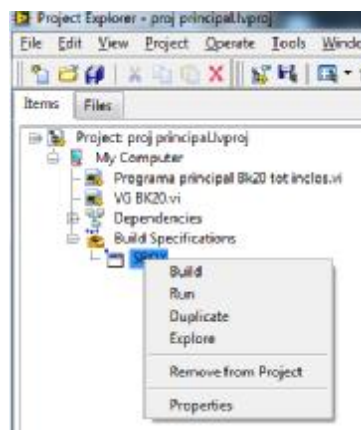


Figura 3.33 Propietats del executable

3.2.10.1. Executable

En la següent figura es mostra la forma que tindrà el executable. Hi ha dos variables de control (variables globals) que són els dos codis (numero de prototip i el part number) de la SBOX i la variable booleana ON/OFF que el operari activarà un cop hagi introduït els codis.

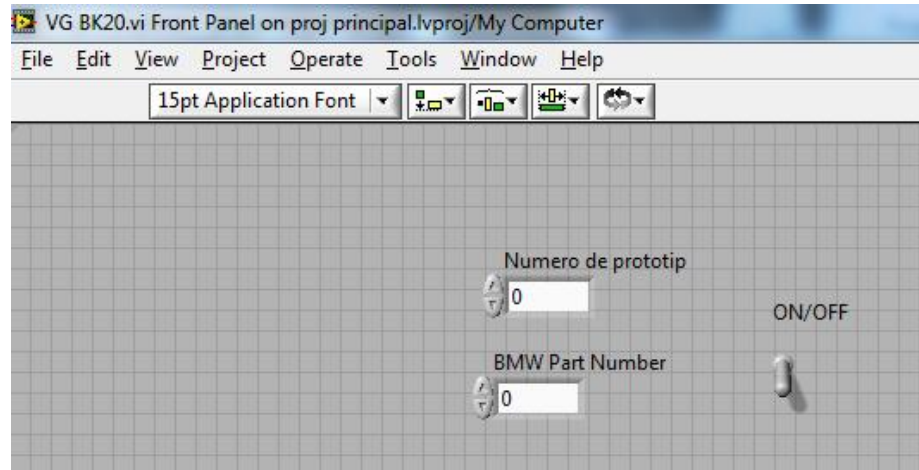


Figura 3.34 Vista del executable

3.3. ALTRES PROGRAMES

En aquest apartat s'explicarà els diferents programes que s'han fet en el transcurs del projecte però que no s'han introduït al programa principal, ja que, no formen part de la validació de la SBOX que s'ha fet però forma part de la feina que s'ha realitzat.

3.3.1. ESCALA DE TENSIONS

Un dels tests que s'ha dissenyat en el transcurs del projecte per comprovar que el software de CANoe (que han realitzat els del departament de software de Lear) funciona correctament, és el de dissenyar un programa LabVIEW, on la font de tensió del Test Bench estigui aplicant de forma continua una escala de tensions, per tal de comprovar que, les mesures que llegeix la SBOX per CAN de forma continua i durant un temps molt llarg (uns dies) són les correctes. D'aquesta manera, es comprovarà que les trames de CAN no es perden en el llarg del temps.

Aquest test no forma part del calibratge i mesura del programa principal del projecte, però s'ha volgut introduir-lo a la memòria, ja que, ha sigut una de les proves importants per tal de verificar que el software que s'ha fet per la PCB de la SBOX funcionés correctament.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

En aquest test, s'ha dissenyat un programa per configurar la tarjeta d'adquisició i per configurar la font de tensió (HVLC) i no s'intervindrà amb la interfície CANoe/LabVIEW, és a dir, es mesurarà els valors de U_{batt} , U_q i U_{k2} directament del programa CANoe a través de les seves gràfiques.

Es realitzarà dos configuracions hardware del Test Bench, a través de la tarjeta d'adquisició, per tal de poder mesurar les tensions:

L'esquema del circuit per mesurar U_{batt} i U_q és el mateix:

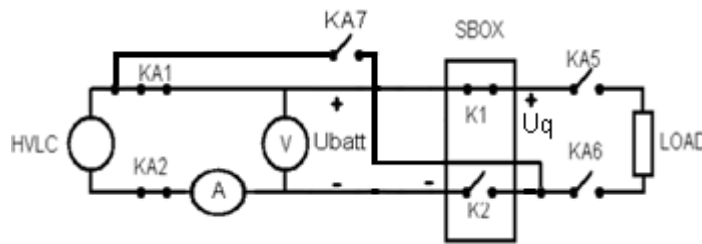


Figura 3.35 Configuració estructura hardware per mesurar U_{batt} i U_q

Per activar i desactivar els relés de la SBOX, com que no hi ha cap interfície CANoe/LabVIEW o s'ha realitzat mitjançant un panell de interruptors connectats per CAN de forma manual.

L'esquema per mesurar U_{k2} és:

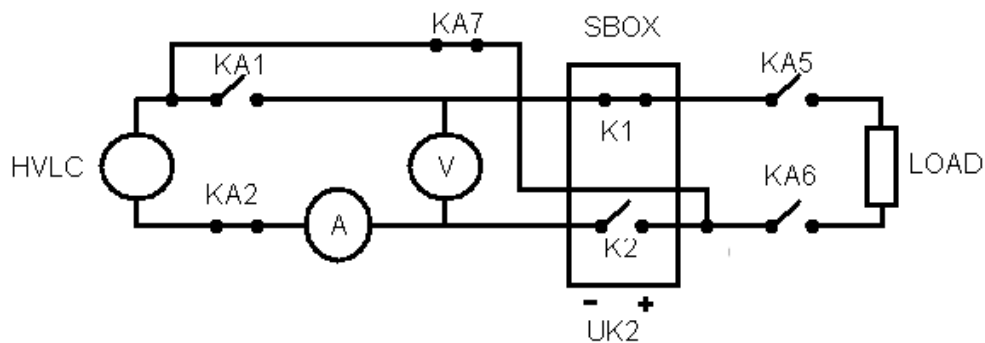


Figura 3.36 Configuració estructura hardware per mesurar U_{k2}

Com que els relés de la SBOX s'activen de forma manual i el procediment és un test que dura dies es deixarà tancat el relé K1 tant per mesurar U_{k2} com U_{batt} i U_q , ja que, per la mesura de U_{k2} no faria falta però tampoc no modifica el valor de U_{k2} .

Per configurar la targeta adquisició i la font de tensió s'utilitzarà el mateix programa que s'ha explicat en aquests apartats i que també s'utilitzarà per al programa principal de calibratge i mesura.

Els valors que s'han de introduir en el programa són:

- VIUBATT i VIUK2: Són els valors desde on s'inicialitza la gràfica i també desde on s'acaba.
- EUBATT i EUK2: Indica cada quants volts s'aplica tensió a la font de tensió (VAPL).
- VFUBATT i VFUK2: Són els valors finals de la recta ascendent de valors i els valors inicials de les rectes de valors descendents.

L'objectiu del programa és aplicar tensió a la font HVLC desde una tensió inicial (VIUBATT i VIUK2) i anar realitzant una pendent de valors aplicat per l'escala de valors (EUBATT i EUK2) fins que sigui igual o superior al valor final (VFUBATT i VFUK2). Després es realitzarà la mateixa pendent però a la inversa fins al valor inicial (VIUBATT i VIUK2).

En la següent figura es pot observar com és la gràfica en el temps, dels valors de tensió que s'aplicaran a la font.

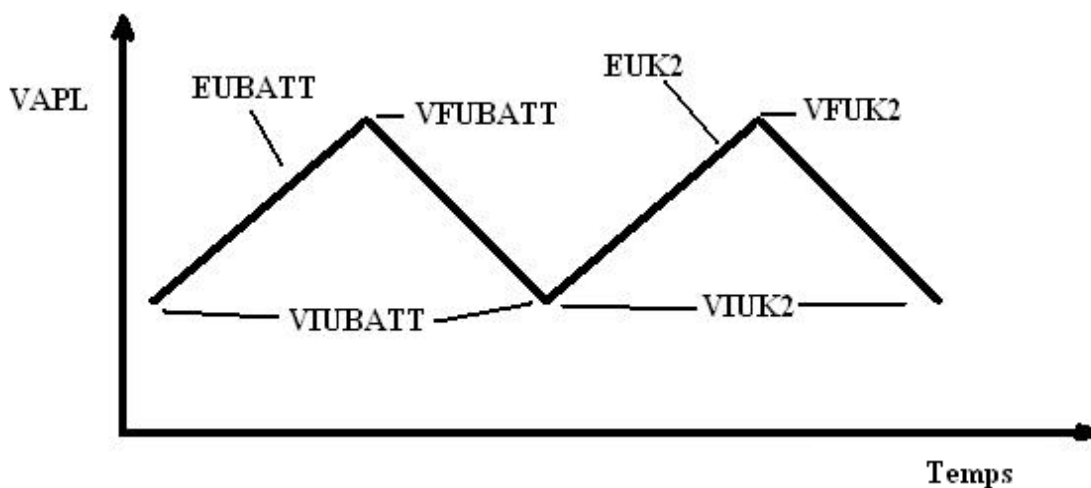


Figura 3.37 Gràfica en el temps de la tensió aplicada per la font HVLC

El programa CANoe realitzarà la lectura dels valors de forma repetitiva i ininterrompuda, i per tant, la gràfica anterior haurà de ser molt semblant a la gràfica del programa CANoe.

3.3.3. ISOLATION MONITORING

En aquest programa es calcularà la resistència d'aïllament de la SBOX. Per calcular aquesta resistència es realitzarà un procediment amb cinc variables del programa CANoe:

- Enable Isolation: Variable (bit) que ha d'estar actiu perquè el procediment de càlcul de la resistència estigui actiu.
- Start/Stop Isolation: Variable (bit) per activar i desactivar el procediment, amb un zero es desactiva i amb un 1 s'activa.
- Sync_Iso_SME: Variable (byte) de sincronisme amb la SME. Variable de control que dona la SME per realitzar el càlcul de la resistència.
- Sync_Iso_SBOX: Variable (byte) de sincronisme amb la SBOX. Variable que es llegeix de la SBOX mentre s'està realitzant els càlculs de la resistència. El procediment de càlcul acaba quan les dos variables de sincronisme són igual.
- Resistència d'aïllament: Valor de la resistència d'aïllament.

El funcionament comença quan s'activa els dos bits de control com són el Enable Isolation i el Start/stop Isolation i es mantindrà en un bucle fins que no sigui així. Un cop els dos bits estiguin a 1, s'haurà de donar un valor a la variable que controla el sincronisme de la SME i s'esperarà que el valor de sincronisme de la SBOX sigui igual. Quan aquests dos bytes són iguals, vol dir que el procediment que ha realitzat la SBOX per al càlcul de la resistència ja ha acabat, i per tant, ja es pot llegir el valor de la resistència d'aïllament de la SBOX.

Aquest programa, com ja s'ha comentat anteriorment, no s'ha adaptat al programa principal i per tant s'ha comprovat que funciona a nivell de subprograma, és a dir, actualment, el programa està dissenyat de tal manera que les variables que es controlen són les de Enable Isolation, Start/stop Isolation i la Sync_Iso_SME i les que es llegeix són les de Sync_Iso_SBOX i la Resistència d'aïllament i l'objectiu, en un futur, serà realitzar un procediment que quan s'adapti al programa principal, el operari (el que valida la SBOX i manipula el programa) no li calgui controlar cap variable.

Cal dir, que el procediment de càlcul de la resistència d'aïllament no s'ha realitzat en aquest projecte sinó que l'han realitzat l'equip de software de Lear i el que s'ha fet en aquest projecte és que, a partir del procediment, adaptar-lo a la interfície LabVIEW-CANoe. En aquest apartat s'ha explicat aquest procediment de forma general per donar una mica una

idea general del càlcul de la resistència però cal dir que el procediment és molt més complex.

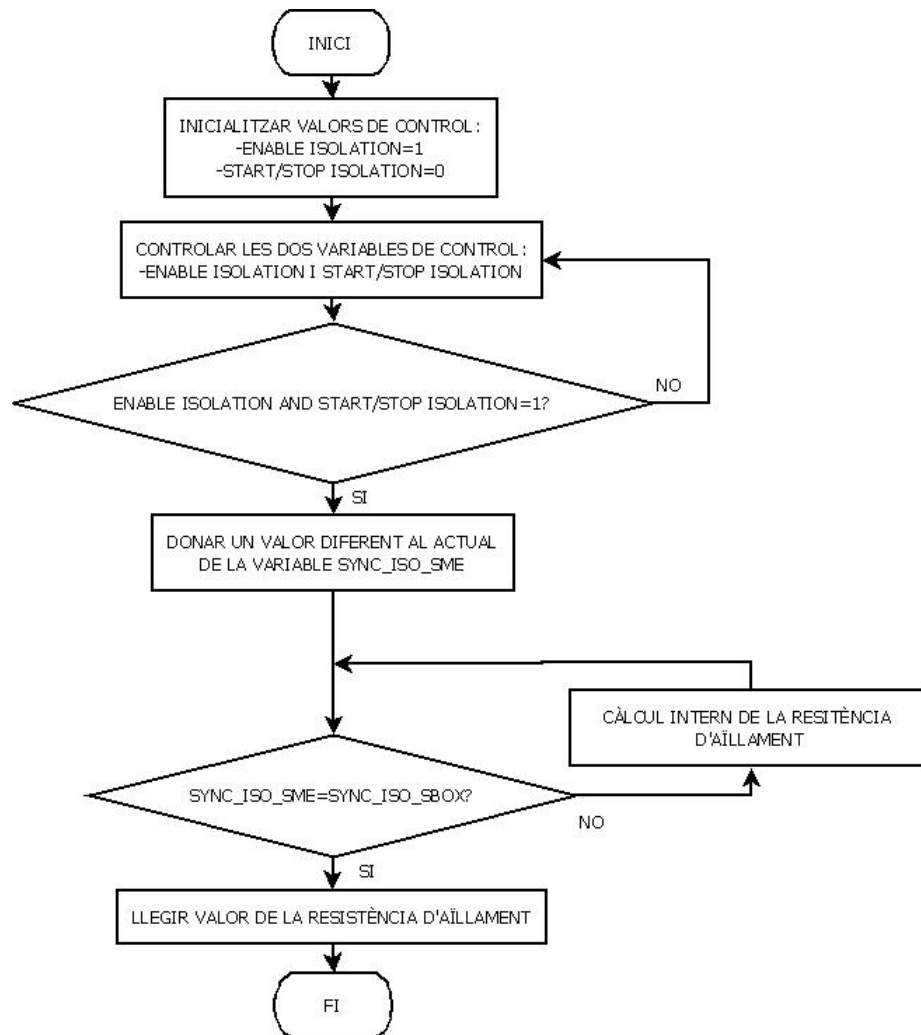


Figura 3.39 Diagrama de flux del programa Isolation Monitoring

3.3.4. CRASH

El Crash és una senyal que reporta la SBOX quan els seus relés estan alimentats a menys de 3V això indicarà un possible error greu en el funcionament del dispositiu (com per exemple, un accident), ja que, com a funcionament normal, els relés estaran alimentats a 12V.

Aquesta senyal no es validarà en el programa principal però s'ha realitzat en mode de subprograma, ja que, en un futur, és a dir, quan s'ampliï la validació serà un dels programes a introduir.

En aquest programa es comprovarà que aquesta senyal funciona correctament, desconnectant la alimentació del dispositiu i llegint a través del CAN la senyal de Crash (senyal: Sta_Crash del missatge: SBOX_Status del programa CANoe) durant 100ms. Si durant aquest temps, la senyal de Crash s'activa significarà que funciona correctament, i sinó voldrà dir que hi ha un error en la generació d'aquesta senyal.

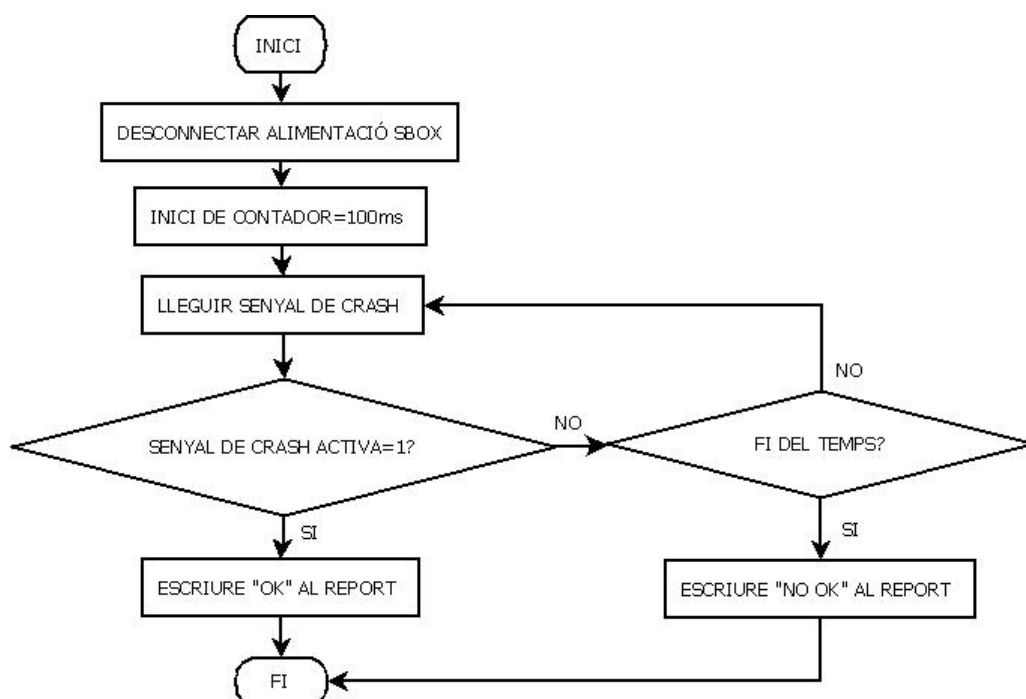


Figura 3.40 Diagrama de flux del programa Crash

Conclusions

En el transcurs d'aquests mesos a Lear Corporation i en la realització d'aquest projecte, he pogut treure diferents conclusions:

- He adquirit alts coneixements sobre programes importants en el món de l'automoció com el programa CANoe i softwares tan importants en el món de la programació gràfica per desenvolupar sistemes de mesura, proves i control com és el programa LabVIEW.
- He adquirit coneixements sobre protocols i busos de comunicació, ja que s'ha estudiat i s'ha treballat amb el bus GPIB, LIN i CAN. Actualment, aquests tres busos són dels més importants en el món de la indústria i de l'automoció.
- He après com es realitza diferents entorns en la realització de tests automàtics per a la validació de diferents prototips, és a dir, he après a seleccionar els dispositius de calibratge i mesura pertinents per a les necessitats que es necessitaven.
- S'ha desenvolupat un test automàtic per Lear Corporation que valida les parts més importants de la SBOX que es fabrica a la mateixa empresa. Per tant, es pot dir que s'ha millorat el procediment de validació que estava instal·lat abans de realitzar el projecte, ja que abans és tardava unes 3 hores en validar una peça i ara es tardà entre 10 i 20 minuts.

4. REFERÈNCIES

- [1] Informació sobre els vehicles híbrids. Disponible a <http://www.slideshare.net/tecnologiajorge/coches-elctricos-7655457>
- [2] Dr James Gover. A tutorial on Hybrid Electric Vehicles: EV, HEV, PHEV and FCEV [arxiu .pdf]. Informació sobre vehicles híbrids.
- [3] Informació sobre els vehicles híbrids. Disponible a http://es.wikipedia.org/wiki/Veh%C3%ADculo_h%C3%ADbrido_el%C3%A9ctrico_enchufable
- [4] Informació sobre els vehicles híbrids. Disponible a http://ca.wikipedia.org/wiki/Vehicle_h%C3%ADbrido
- [5] Muhammad H.Rashid. Power Electronics Handbook. University of West Florida 2001.
- [6] Erickson, Robert W.). Fundamentals of Power Electronics. Second Edition. Secaucus, NJ, USA: Kluwer Academic Publishers, 2000.
- [7] PSpice. Manual Pspice [arxiu .pdf]
- [8] Escuela de Ingenieros de Sevilla. Información sobre el bus GPIB [arxiu .pdf]
- [9] Elgar Electronics Corporation. PowerTEN 3300I Series DC Power Supplies, IEEE 488.2 Programming Manual [arxiu .pdf] (Comandes per la norma IEEE 482).
- [10] Agilent Technologies. Comandes de la norma SCPI per la font auxiliar (E3363A). Disponible a http://na.tm.agilent.com/pna/help/latest/Programming/GPIB_Command_Finder/System.htm#PSensorType
- [11] Protocol de comunicació CAN. Disponible a <http://canbus.galeon.com/electronica/canbus.htm>
- [12] Protocol de comunicació LIN. Disponible a <https://sites.google.com/site/sistemasdemultiplexado/protocolos-de-comunicacin/lin-bus>
- [13] Victor Jiménez. Estudio del nuevo bus de automoción Flexray y diseño de un prototipo ilustrativo [arxiu .pdf] (protocol comunicació LIN)

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

[14] Vector. Información sobre la tarjeta CANCASE XL. Disponible a <http://es.edaboard.com/topic-3956732.0.html>

[15] National Instruments. LabVIEW, Data Acquisition Basics Manual [arxiu .pdf]

[16] José Rafael Lajara Vizcaíno i José Pelegrí Sebastià. LaVIEW 8.2, Entorno gráfico de programación. Marcomobo S.A. 2007.

[17] Antonio Manuel Lázaro i Joaquim Rio Fernández. LabVIEW 7.1, Programación gráfica para el control de Instrumentación. Ediciones PARANINFO S.A. 2005.

[18] Pere J.Riu Costa, Javier Rosell Ferrer i Juan Ramos Castro. Sistemas de Instrumentación. Edicions UPC 1995.

[19] Material de suport per als programes LabVIEW. Disponible a <http://www.ni.com/downloads/idnet/esa/>

[20] National Instruments. Informació sobre LabVIEW. Disponible a la eina d'ajuda del programa.

[21] Juan Albornoz Carrasco y Pablo Lagos Norambuena. Guía de iniciación en LabVIEW 7.1 [arxiu .pdf]. Universidad de la Frontera 2008.

[22] Vector. CANoe Tutorial Version 2.0 [arxiu .pdf]. Tutorial del programa CANoe

[23] Informació sobre el projecte VERDE. Disponible a <http://cenitverde.es/>

5. APÈNDIX

A. DIPOSITIUS DEL SISTEMA

En aquest apartat s'explicarà els diferents busos de comunicació utilitzats en el projecte.

A.1. BUS GPIB

S'explicarà el bus GPIB a través de les tres normes que defineixen el funcionament i l'estructura d'aquest tipus de protocol de comunicació tant important en el projecte, ja que, tots els dispositius de mesura i fonts de tensió estaran interconnectats mitjançant aquest bus.

La norma ANSI/IEEE 488.2 de 1987 (revisada al 1992) va introduir una estandardització de molts aspectes que no estaven inclosos a la norma IEEE 488.1 (1975). Així doncs, aquesta norma estableix algunes especificacions del software que no estaven inclosos a la norma original. Tot i així, es deixa obert el format i el tipus de comandes que s'han d'enviar als instruments. Al 1991, apareix la norma SCPI (Standard Commands for Programmable Instruments) per aconseguir una estandardització de les comandes de control i el format de les dades dels instruments. El objectiu és que, independentment del fabricant, equips que tenen la mateixa funcionalitat responen d'igual forma a un conjunt estàndard de comandes.

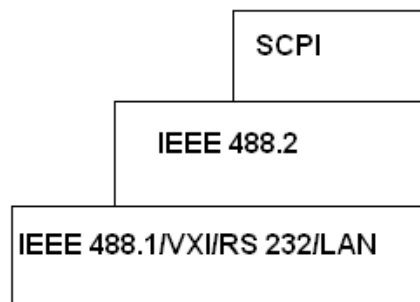


Figura A.1 Jerarquia de les norma per a instruments controlables

A.1.1. LA NORMA IEEE-488.1

Aquesta norma especifica les diferents característiques físiques i les especificacions funcionals (estructura del bus, capacitat fundamentals en la comunicació, etc).

Així doncs, els objectius d'aquesta norma són:

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Crear un sistema de comunicació de instrumentació on els instruments estiguin pròxims.
- Especificar els requisits mecànics, elèctrics i funcionals per poder interconnectar instruments.
- Permetre la comunicació directa entre instruments sense necessitat de que les dades passin pel controlador.
- Definir un sistema que no introdueix restriccions sobre les característiques que tenen els instruments.
- Definir un sistema fàcil d'utilitzar, de baix cost i que permeti comunicacions asíncrones en un ampli marge de velocitats de transmissió de dades.

A.1.2. CARACTERÍSTIQUES DEL BUS GPIB

El bus GPIB està format per 16 línies de senyal i 8 línies de terra. Les 18 línies de senyal estan dividides en tres grups (8 línies de dades, 3 línies de handshake i 5 línies de control).



Figura A.2 Controlador GPIB amb interfase USB i el pinout del bus GPIB [16]

- Es poden connectar fins a 15 dispositius en el sistema dels quals un d'ells ha de ser el controlador que estableixi la funció que ha de realitzar cada un dels altres.
- Un dispositiu connectat al bus, pot enviar o rebre informació fins a o desde qualsevol dels altres 14 equips. A vegades, la pròpia naturalesa d'un equip fa que solament estigui capacitat per rebre (per exemple una impressora), o solament estigui capacitat per enviar (per exemple un comptador), o qualsevol de les dos funcions (per exemple un oscil·loscopi o un computador).

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- El límit pràctic de velocitat de intercanvi de dades és de 500Kbytes/s (o 4Mbits/s).
- La interconnexió entre equips es realitza utilitzant cables de 25 fils, finalitzant en connectors de doble boca (mascle per un costat i femella per l'altre), que permet la interconnexió dels equips en qualsevol configuració (estrella, línia, o qualsevol combinació de les dos). Per a la connexió entre l'ordinador i el instrument, a part del de doble boca, també es pot utilitzar el controlador GPIB amb interfase USB que és el que es farà servir per al projecte i que es pot veure en l'anterior figura.
- Les longituds màximes permeses en els cables és de 20 metres. Els cables que es comercialitzen són de 1,2,4 i 8 metres.

Per al projecte, s'utilitzarà la configuració dels dispositius GPIB (en el Test Bench seran 6 dispositius) que es pot observar en la següent figura i, d'aquesta manera, es complirà amb les característiques definides anteriorment per la norma IEEE-488.1.

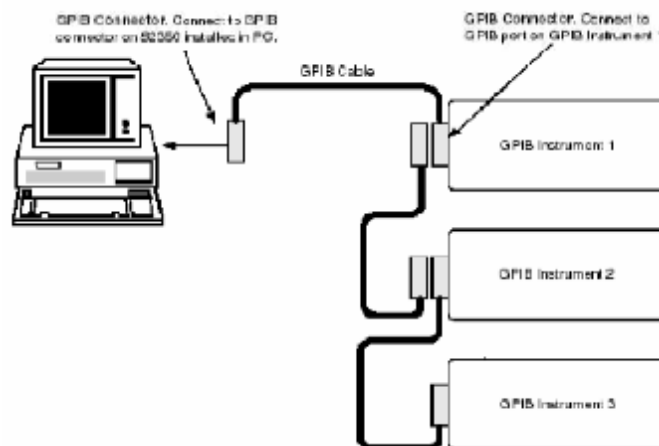


Figura A.3 Estructura GPIB [8]

A.1.3. LA NORMA IEEE-488.2

La creació d'aquesta norma va suposar un pas més en la compatibilitat d'instruments, introduint solucions com:

- Definir el mínim conjunt de capacitats que ha de tenir un instrument.
- Especificar la forma de presentar les dades a través del bus.
- Definir un protocol per enviar missatges de dispositius i la forma d'enviar varis missatges en una sola cadena de caràcters.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Donar un conjunt de comandes comuns a tots els instruments.
- Definir el model estàndard de bytes de informació sobre els estats del dispositiu.

En la següent taula hi ha les comandes i preguntes més comunes que la norma obliga a incloure en un instrument.

COMANDA	DESCRIPCIÓ
*CLS	Clear Status Command
*ESE	Standard Event Status Enable Command
*ESE?	Standard Event Status Enable Query
*ESR?	Standard Event Status Register Query
*IDN?	Identification Query
*OPC	Operation Complete Command
*OPC?	Operation Complete Query
*RST	Reset Command
*SRE	Service Request Enable Command
*SRE?	Service Request Enable Query
*STB?	Read Status Byte Query
*TST?	Self-test Query
*WAI?	Wait-to-Continue Command

Taula A.1 Comandes norma IEEE 488.2 [16]

De l'anterior taula es pot destacar per al projecte el *CLS que inicialitza el estatus del instrument, *IDN? que s'utilitza quan es vol que un instrument s'identifiqui i el *RST que inicialitza el propi instrument col·locant en un estat conegut.

A.1.4. NORMA SCPI

La norma SCPI és el escaló més alt dins de la jerarquia normativa per al control de sistemes d'instrumentació. La norma SCPI s'assenta sobre la IEEE 488.2 i aquesta, al mateix temps, es basa en la norma IEEE 488.1. Tot i que, existeix aquesta jerarquia, les comandes i la estructura de dades basats en la norma SCPI es poden utilitzar en sistemes de instrumentació que no estan basats en IEEE 488, per exemple, en sistemes basats en VXI, RS-232 o LAN.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

El objectiu general de la norma SCPI es reduir els costos de desenvolupament i manteniment de programes de control de sistemes d'instrumentació per a proves automàtiques. Aquest objectiu s'aconsegueix, ja que, el compliment de la normativa:

- Facilita el aprenentatge i utilització de les comandes i les dades.
- Facilita el desenvolupament i manteniment dels programes.
- Possibilita la substitució d'equips amb els mínims canvis de software.

Per aconseguir aquests objectius la norma estableix la sintaxis i els formats dels missatges per tal de que instruments amb la mateixa funcionalitat o instruments del mateix tipus utilitzen els mateixes comandes. Per exemple, les comandes per mesurar una freqüència utilitzant freqüencímetres de diferents fabricants seran els mateixos. A més a més, la mesura de la freqüència amb altres instruments que ho permetin, com per exemple, un oscil·loscopi digital o un multímetre, també utilitzaran les mateixes comandes.

La norma estableix tres tipus de compatibilitat entre instruments:

- Vertical: equips d'un mateix tipus tenen els mateixos controls. Exemple: en dos multímetres diferents la selecció d'escala es realitzarà de la mateixa manera.
- Horitzontal: equips que realitzen la mateixa mesura, tot i que, siguin de maneres diferents, utilitzant les mateixes comandes. Exemple: un oscil·loscopi que pot mesurar períodes i un freqüencímetre-comptador utilitzaran les mateixes comandes per mesurar el període d'una senyal.
- Funcional: dos equips diferents que poden realitzar la mateixa funció ho faran amb les mateixes comandes. Exemple: un analitzador d'espectres que poden realitzar escombrats de freqüència i un generador de senyal seran funcionalment compatibles si el escombrat de freqüència es programa amb les mateixes comandes.

En la següent figura es pot observar els diferents diagrames de blocs de sistemes que es pot aplicar als instruments SCPI.

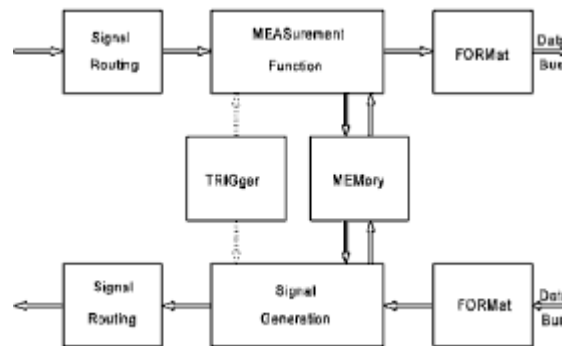


Figura A.4 Model d'instrument definit en la norma SCPI [10]

Els blocs de cada model són:

- Signal routing (encaminament de la senyal): S'ocupa de dirigir la senyal (una senyal física) des de el exterior al bloc de mesura de senyals o des de l'interior del instrument en el bloc de senyal al exterior.
- El bloc de mesura (measurement function) converteix la senyal a un format que admet el processat. Aquest bloc s'ha dividit en tres subsistemes:
 - INPut: Condiciona la senyal d'entrada abans de ser convertida a format digital, mitjançant filtres, escalats o atenuacions.
 - SENSE: Transforma la senyal a un format digital que es pot manipular fàcilment. Així es controla la resolució, el rang de mesura, el rebuig en mode sèrie, etc.
 - CALCulate: Amb aquest subsistema es transforma les dades adquirides a un format més útil que facilita l'aplicació; com per exemple, la conversió d'unitats, freqüència.
- El bloc de generació de senyal (signal generation) transforma dades internes en senyals que es poden utilitzar en el món exterior. Aquest bloc es subdivideix amb tres subblocs:
 - OUTPut: Una vegada la senyal ha sigut generada, es pot condicionar-la mitjançant filtres o sumes de offsets abans d'extraure-la al exterior.
 - SOURce: Es genera la senyal segons certes característiques que s'indiquen. Els exemples són la modulació en amplitud, la corrent a proporcionar, la freqüència, etc.

- CALCulate: Es converteix en unitats o es canvia de domini.
- El bloc de memòria: reté la informació interna necessària per arribar a la resta dels processos.
- El bloc de format adapta la informació que es genera en el equip o es recull del exterior per a que puguin ser transmesa pel bus, verificant convenientment la norma.
- El bloc de trigger o dispar sincronitza els diversos blocs i accions del instrument amb successos externs o amb altres instruments.

A.1.4.1. Sintaxis i estil

Les comandes en la norma SCPI s'agrupen jeràrquicament en forma de arbre. A l'arrel del arbre es troba a les comandes que fan referència als blocs que apareixen en el model d'instrument que s'ha comentat en l'apartat anterior. Cada un d'aquests comandes es divideix en un conjunt de rames identificades per paraules clau que al mateix temps, identifiquen a funcions subordinades al bloc de l'arrel i així successivament

En la següent figura es representa l'arbre del bloc del INPut. La notació (també serveix per un bloc diferent a INPut) que segueix és la següent:

: => Indica el pas de un nivell jeràrquic inferior.

[] => Paraules clau opcionals.

<> => Tanquen el tipus de paràmetre.

| => Separa paràmetres opcionals (solament es poden posar un d'ells).

; => Separa les comandes que estan en la mateixa línia (no canvia el nivell de l'última comanda).

, => S'utilitza per separar diferents paràmetres dins del mateix nivell.

? => Indica que és una comanda de consulta i que es s'espera una resposta del equip al que s'envia. Es molt important llegir la dada sol·licitada; de no ser així, al enviar un altra comanda s'apropa una situació d'error en el instrument.

KEYWORD	PARAMETER FORM	NOTES
INPut		
:ATTenuation	<numeric_value>	<tipo de variable>
:AUTO	<Boolean> ORCE	separa paràmetres opcionales
:BIAS		
[:STATE]	<Boolean>	opcional
:VOLTage		
[:DC]	<numeric_value>	opcional
:AC	<numeric_value>	
:TYPE	CURRent VOLTage	otras opciones
:	.	.
:COUpling	AC DC ORound	
:GAIN	<numeric_value>	
:AUTO	<Boolean> ORCE	otras opciones
.	.	.
.	.	.

Figura A.5 Estructura jeràrquica de comandes per al bloc de INPut d'un instrument [16]

A.1.4.2. Comandes

A continuació, en la següent figura, es pot observar un seguit d'exemples de possibles comandes en un multímetre digital per als blocs de mesura i configuració. Existeixen molt exemples per a diferents aparells però per simplificar l'explicació només es cita aquests, ja que, en el projecte s'utilitzarà dos multímetres i s'aplicarà alguna d'aquestes comandes.

```

MEASURE
:VOLTage:DC? {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:VOLTage:DC:RATio? {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:VOLTage:AC? {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:CURRent:DC? {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:CURRent:AC? {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:RESistance? {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:FRESistance? {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:FREquency? {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:PERiod? {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:CONTinuity?
:DiODE?

CONFIGure
:VOLTage:DC {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:VOLTage:DC:RATio {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:VOLTage:AC {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:CURRent:DC {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:CURRent:AC {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:RESistance {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:FRESistance {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}
:FREquency {<range>|MIN|MAX|DEF}, {<resolution>|MIN|MAX|DEF}

```

Figura A.6 Exemple de comandes possibles en un multímetre digital pels blocs de mesura i configuració [16]

A.1.5. DRIVERS PER A GPIB

La programació d'un instrument de mesura a través del bus GPIB es realitza mitjançant el enviament de cadenes de caràcters (ASCII). Cada instrument respon a un conjunt particular de comandes, per lo que el programador es veu obligat a aprendre'ls per a realitzar el

control d'un dispositiu. Això implica un cost elevat en temps, per lo que sorgeix la necessitat de crear codi reutilitzable anomenat driver.

Un driver d'un instrument és una llibreria d'alt nivell per controlar dispositius GPIB, VXI, RS232 o altres interfícies. Amb un driver es possible el control d'un instrument sense conèixer la sintaxis de les comandes entrada/sortida de baix nivell.

En definitiva, un driver és un conjunt de funcions, que es criden desde dins d'una aplicació i que realitza una o més accions sobre el instrument. Aquestes accions van desde enviar un missatge SCPI a través del bus, fins a el tractament de les dades rebudes del instrument (FFT, series de Fourier, etc).

S'han desenvolupat una sèrie de estàndards per la implementació dels drivers però, en aquest projecte, s'utilitzarà l'arquitectura anomenada VISA (Virtual Instrument Software Architecture) per a la construcció dels nostres programes en LabVIEW.

Els tipus de dades definides en VISA permeten la portabilitat de drivers a diferents sistemes operatius i llenguatges de programació. Cada funció del dispositiu es genèrica i independent del tipus de interface (GPIB,VXI,etc) utilitzat per controlar el instrument.

A.2. BUS CAN (Controller area network)

El CAN-Bus és un protocol de comunicació en sèrie desenvolupat per Bosch per a l'intercanvi d'informació entre unitats de control electròniques del automòbil.

Aquest és un sistema que permet compartir una gran quantitat d'informació entre les unitats de control abonades al sistema, el que provoca una reducció important tant del número de sensors utilitzats com de la quantitat de cables que formen la instal·lació elèctrica.

D'aquesta manera augmenten considerablement les funcions en els sistemes del automòbil on s'utilitza el CAN-Bus sense augmentar els costos, a més a més, de que aquestes funcions poden estar repartides entre unitats de control.

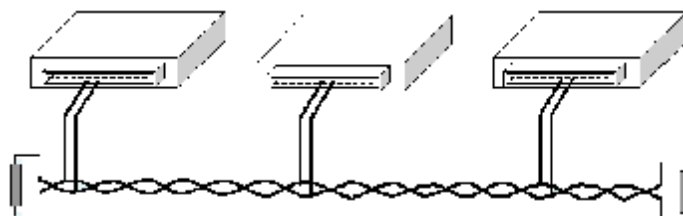


Figura A.7 Connexió CAN [11]

A.2.1. CARACTERÍSTIQUES PRINCIPALS

La informació que circula entre les unitats de control a través dels dos cables (bus) són paquets de 0 i 1 (bit) amb una longitud limitada i amb una estructura definida de camps que formen part del missatge.

Un d'aquest camps actua d'identificador del tipus de dada que es transporta, de la unitat de control que el transmet i de la prioritat per transmetre respecte a altres. El missatge no va dirigit a cap unitat de control en concret cada un d'elles reconeixerà mitjançant aquest identificador si el missatge l'interessa o no.

Totes les unitats de control poden ser transmissores o receptors i la quantitat de les mateixes abonades al sistema pot ser variable (dins dels límits).

Si la situació ho exigeix, una unitat de control pot sol·licitar a una altra una determinada informació mitjançant un dels camps del missatge (trama remota o RDR).

Qualsevol unitat de control introdueix un missatge en el bus amb la condició de que està lliure, si una altra ho intenta al mateix temps el conflicte es resol per la prioritat del missatge indicat pel seu identificador.

El sistema està dotat d'una sèrie de mecanismes que assegurin que el missatge es transmet i repel·lida correctament. Quan un missatge presenta un error es anul·la i torna a transmetre de forma correcta, de la mateixa manera, una unitat de control amb problemes avisa a les demes mitjançant el propi missatge, si la situació és irreversible, aquesta unitat queda fora del servei però el sistema continua funcionant.

A.2.2. COMPOSICIÓ DEL SISTEMA CAN-BUS

A.2.2.1. Cables

La informació circula pels dos cables trenats que uneixen totes les unitats de control que formen el sistema. Aquesta informació es transmet per diferència de tensió entre els dos cables, de manera que un valor alt de tensió representa un 1 i un valor baix de tensió representa un 0. La combinació de uns i zeros formen el missatge que s'ha de transmetre.

En un cable, els valors de tensió oscil·len entre 0 i 2.25V, i per això hi ha el cable L (Low) i el cable H (High) ho fa entre 2.75V i 5V. S'hi ha alguna interrupció en la línia H o es deriva a massa, el sistema treballa amb la senyal de Low amb respecte a massa, en el cas de que hi

hagi una interrupció en la línia L el sistema treballa amb la senyal H amb respecte a massa. Aquesta situació permet que el sistema continuï treballant amb un dels cables tallats o comunicats a massa i inclús amb els dos comunicats també seria possible el funcionament, és a dir, només queda fora de servei quan queden fora de servei els dos cables.

Es important tenir en compte que el trenat entre les dos línies serveix per anular els camps magnètics pel que no s'han de modificar en cap cas el pas ni la longitud d'aquests cables.

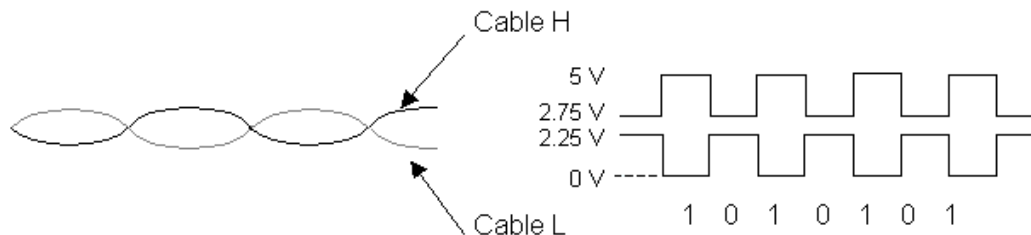


Figura A.8 Connectors i senyals del bus CAN [11]

A.2.2.2. Element de tancament

Són resistències connectades als extrems dels cables H i L. Els seus valors s'obtenen de forma empírica i permeten adequar el funcionament del sistema a diferents longituds de cables i número de unitats de control abonades, ja que, impedeixen fenòmens de reflexió que poden pertorbar el missatge.

Per qüestions de economia i seguretat de funcionament, aquestes resistències estan situades en el interior d'algunes de les unitats de control del sistema.

A.2.2.3. Controlador

És el element encarregat de la comunicació entre el microprocessador de la unitat de control i el transmissor-receptor. Treballa acondicionant la informació que entra i surt entre els dos components.

El controlador està situat en la unitat de control, per tant, existeixen tantes unitats control com unitats connectades al sistema. Aquest element treballa amb nivells de tensió molt baixos i és el que determina la velocitat de transmissió dels missatges que serà més o menys elevada segons el compromís del sistema. Així doncs, per exemple, en la línia CAN-Bus del motor-frenos-canvi automàtic, la velocitat és de 500K baudis i en el sistema de confort és de 62.5K baudis. Aquest element també intervé en la necessària sincronització entre les diferents unitats de control per la correcta emissió i recepció dels missatges.

A.2.2.4. Transmissor/Receptor

El transmissor-receptor és el element que té la missió de rebre i de transmetre les dades i també te la funció de condicionar i preparar la informació per que pugi ser utilitzada pels controladors. Aquesta preparació consisteix en situar els nivells de tensió de forma adequada amplificant la senyal quan la informació s'introdueixi a la línia i reduint-la quan es recollida de la mateixa i subministrada al controlador.

El transmissor-receptor és bàsicament un circuit integrat que està situat en cada una de les unitats de control abonades al sistema, treballa amb intensitats pròximes a 0.5A i en cap cas intervé modificant el contingut del missatge. Funcionalment està situat entre els cables que formen la línia CAN-Bus i el controlador.

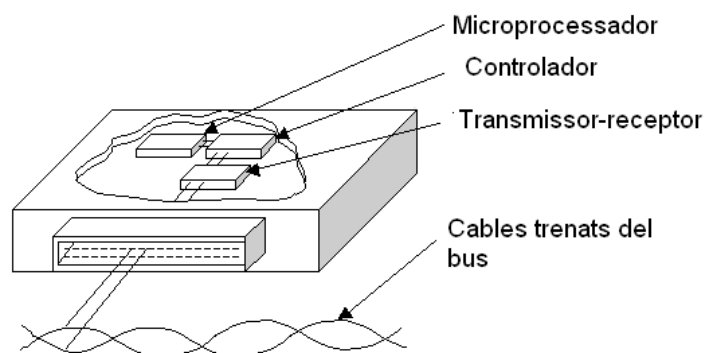


Figura A.9 Parts d'una unitat de control [11]

A.2.3. FUNCIONAMENT DEL SISTEMA CAN-BUS

Les unitats de control que es connecten al sistema CAN-Bus són les que necessiten compartir informació, tot i que, no formen part del mateix sistema. En automoció generalment estan connectades a una línia les unitats de control del motor, del ABS i del canvi automàtic i a una altra línia (de menys velocitat) les unitats de control relacionades amb el sistema de confort.

El sistema CAN-Bus està orientat al missatge i no al destinatari. La informació en la línia es transmesa en forma de missatges estructurats en la que una part del mateix és un identificador que indica la classe de dades que conté. Totes les unitats de control reben el missatge, el filtren i solament ho apliquen les que necessiten aquesta dada. La totalitat de unitats de control abonades al sistema són capaces tant de introduir com de rebre missatge de la línia. Quan el bus està lliure, qualsevol unitat connectada pot començar a transmetre un nou missatge.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

En el cas de que una o varies unitats vulguin introduir un missatge al mateix temps, ho farà el que tingui una major prioritat. Aquesta prioritat ve indicada pel identificador.

El procés de transmissió de dades és el següent:

- Subministrament de dades: Una unitat de control rep informació dels sensors que tenen associats (r.p.m. del motor, velocitat, temperatura del motor, etc). El seu microprocessador passa la informació al controlador on es gestiona i acondicionada per ser passada al transmissor-receptor on es transforma en senyals elèctriques.
- Transmissió de dades: El controlador que pertany a aquesta unitat, transfereix les dades i el seu identificador juntament amb la petició de inici de transmissió i assumint la responsabilitat de que el missatge sigui correctament transmès a totes les unitats de control associades. Per transmetre el missatge hi ha d'haver el bus lliure i en cas de col·lisió amb una altra unitat de control que transmet simultàniament, tenen una prioritat més alta. A partir del moment en que passa això, la resta de unitats de control es converteixen en receptores.
- Recepció del missatge: Quan la totalitat de les unitats de control reben el missatge verifiquen el identificador per determinar si el missatge serà utilitzat. Les unitats de control que necessiten les dades del missatge ho processaran i si no ho necessiten, el missatge serà ignorat.

El sistema CAN-Bus disposa de mecanismes per detectar errors en la transmissió de missatges, de manera que, tots els receptors realitzen una revisió del missatge analitzant una part del mateix, aquest camp s'anomena CRC. En les unitats de control s'apliquen altres mecanismes de control que monitoritzen el nivell del bus, la presència de camp de format fix en el missatge (verificació de la trama), anàlisis estadístics per part de les unitats de control dels seus propis errors entre d'altres.

Aquestes mesures fan que les probabilitats d'error en la emissió i recepció de missatges siguin molt baixes i això fa que sigui un sistema molt segur.

El plantejament del CAN-Bus permet disminuir el cablejat en el automòbil, ja que, per exemple, si una unitat de control disposa de una informació, com per exemple, la temperatura del motor, aquesta dada pot ser utilitzada per la resta de unitats de control sense que sigui necessari que cada una de elles rebí la informació del sensor de temperatura.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

Una altre avantatge es que les funcions poden ser repartides entre diferents unitats de control i s'hi es vol incrementar les funcions tampoc no suposa cap cost addicional.

En la següent figura es pot observar un exemple del comportament de 4 unitats de control en el tractament de la senyal de velocitat del motor.

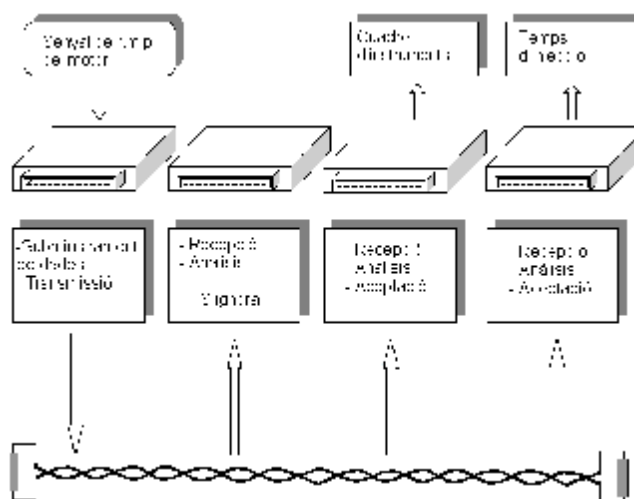


Figura A.10 Funcionament del sistema CAN-Bus [11]

A.2.3.1. Estructura del missatge

El missatge és una successió de “0” i “1” que com s’ha explicat al principi, estan representats per diferents nivells de tensió en els cables del CAN-Bus i que s’anomenen “bit”.

El missatge té una sèrie de camps de diferents dimensions (número de bits) que permeten portar a cap el procés de comunicació entre les unitats de controls segons el protocol definit per Bosch pel CAN-Bus que faciliten identificar la unitat de control, indicar el principi i el final del missatge, mostrar les dades, permetre diferents control, entre d’altres.

Els missatges són introduïts en la línia amb una cadència que oscil·la entre els 7 i els 20 milisegons depenent de la velocitat del àrea i de la unitat de control que els introdueixi.

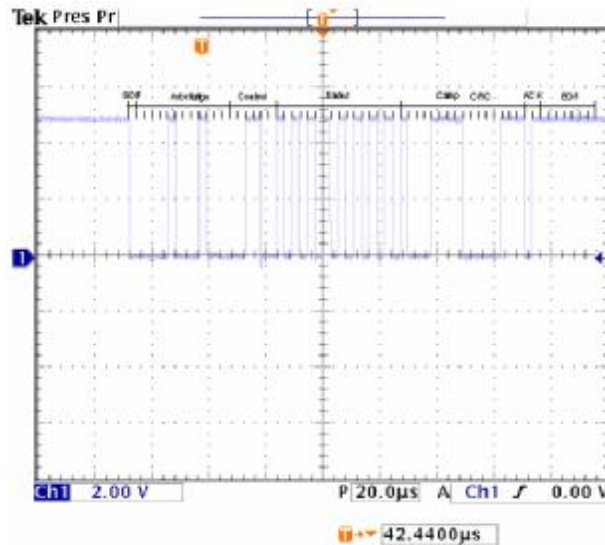


Figura A.11 Exemple d'una trama CAN enviant un byte

- Camp d'inici del missatge (SOF): El missatge s'inicia amb un bit dominant, el qual el flanc descendent es utilitza per les unitats de control per sincronitzar-se entre elles.

- Camp d'arbitratge: Els 11 bits d'aquest camp s'utilitzen com a identificador que permet reconèixer a les unitats de control, la prioritat del missatge. Quan més baix sigui el valor del identificador més alta és la prioritat i per tant determina l'ordre en que seran introduïts els missatges en la línia.

El bit RTR indica si el missatge conté dades (RTR=0) o si es tracta d'una trama remota sense dades (RTR=1). Una trama de dades sempre té una prioritat més alta que una trama remota.

La trama remota s'utilitza per a sol·licitar dades a altres unitats de control o simplement perquè es necessiten o per realitzar una revisió.

- Camp de control: Aquest camp informa sobre característiques del camp de dades. El bit IDE indica quan és un "0" que es tracta d'una trama estàndard i quan és un "1" que és una trama extensa. Els quatre bits que componen el camp DLC indiquen el nombre de bytes que hi ha en el camp de dades.

La diferència entre una trama estàndard i una trama extensa és que la primera té 11 bits i la segona 29 bits. Les dues trames poden coexistir eventualment i la raó de la seva presència és l'existència de dos versions de CAN.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Camp de dades: En aquest camp apareix la informació del missatge amb les dades que la unitat de control corresponen introdueix a la línia CAN-Bus. Pot contenir entre 0 i 8 bytes (de 0 i 64 bits).
- Camp de seguiment (CRC): Aquest camp té la longitud de 16 bits i es utilitza per la detecció d'errors pels 15 primers, mentre que el últim sempre és un bit recessiu (1) que delimita el camp CRC.
- Camp de confirmació (ACK): El camp ACK està format per dos bits que són sempre transmesos com a recessius (1). Totes les unitats de control que reben el mateix CRC modificant el primer bit de camp ACK per un de dominant (0), de forma que, la unitat de control que encara està transmetent reconeix que al menys alguna unitat de control a rebut un missatge escrit correctament. De no ser així, la unitat de control transmissora interpreta que el seu missatge presenta un error.
- Camp de final de missatge (EOF): Aquest camp indica el final de missatge amb una cadena de 7 bits recessius.

Pot passar que en determinats missatges es produeixin llargues cadenes de zeros o uns i que aquest provoqui una pèrdua de sincronització entre unitats de control. El protocol CAN resol aquesta situació introduint un bit de diferent polaritat cada cinc bits iguals: cada cinc "0" s'introdueix un "1" i viceversa. La unitat de control que utilitza el missatge, descarta un bit posterior a cinc bits iguals. Aquests bits reben el nom de bit stuffing.

En la següent taula es pot observar un exemple real de un missatge CAN.

SOF	Identificador	RTR	DE	DLC	DADA (Byte1)	DADA (Byte2)	CRC	ACK	FN
0	1100010000	0	000	0010	00010110	00000000	0	0	11111

Taula A.2 Missatge CAN [11]

A.2.3.2. Diagnòstic del CAN-Bus

El sistema de seguretat que incorpora el CAN-Bus permeten que les probabilitats d'error en el procés de comunicació siguin molt baixes, però segueixen sent possible que cables, contactes i les pròpies unitats de control presentin algun error.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

Per l'anàlisi d'una averia, s'han de tenir present que una unitat de control avariada abonada al CAN-Bus en cap cas impedeix que el sistema treballi amb normalitat. Lògicament no serà possible realitzar les funcions que impliquin el ús d'informació que proporcioni la unitat avariada però sí totes els altres unitats.

Per exemple, si es quedés fora de servei la unitat de control d'una porta, no funcionaria el tancament elèctric ni es podrien accionar el de la resta de les portes.

S'hi hagués un averia en els cables del bus, seria possible accionar elèctricament el pany de la porta, però no les altres (només passaria si els dos cables es tallen o es curtcircuiten a massa).

També es possible localitzar errors en el CAN-Bus consultant el sistema de auto diagnòstic del vehicle, on es podran esbrinar des de el estat de funcionament del sistema fins les unitats de control associades al mateix, però necessàriament s'ha de disposar del equip de revisions apropiat.

Un altra alternativa es utilitzar el programa informàtic que s'utilitzarà en el projecte que s'anomena CANoe on es podrà visualitzar el tràfic de dades en el CAN-Bus, indicar el contingut dels missatges, visualitzar rendiment i visualitzar errors.

A.3. BUS LIN (Local Interconnect Network)

El bus LIN és un protocol de comunicació sèrie de baix cost que està format per un master i múltiples esclaus. Pertany a la classe A, ja que, la seva velocitat és limitada, d'uns 20Kbps. Es sol utilitzar per ECUs de control senzilles, com per exemple, en el sector del automòbil, per al tancament de les portes o el control dels seients.

Les xarxes LIN solen ser subxarxes d'una xarxa principal CAN. El seu ús està orientat a aplicacions no crítiques. Algunes característiques importants de LIN són la autosincronització dels esclaus per una seqüència de clock generada pel master, així com, el master disposa d'una taula de programació (schedule) per determinar quan i quina trama ha de ser transmesa. També es pot posar els nodes en mode sleep per estalviar energia, en el cas de no utilitzar-se en aquest moment.

A.3.1. COMPOSICIÓ DEL SISTEMA LIN-BUS

En aquest apartat s'explicarà les diferents parts i el funcionament del LIN-Bus.

A.3.1.1. Unitat de control LIN mestra

La unitat de control que va connectada al CAN-Bus és la que executa les funcions de la mestra al LIN-Bus.

Les funcions que realitza aquesta unitat són:

- Controlar la transmissió de dades i la seva velocitat. La unitat de control LIN mestra transmet l'encapçalament del missatge (header).
- En el programari es defineix un cicle, segons el qual s'han de transmetre missatges al LIN-Bus i quins són.
- Assumeix la funció de traducció entre les unitats de control LIN abonades al sistema de LIN-Bus local i el CAN-Bus de dades. D'aquesta manera és l'única unitat de control del LIN-Bus que va connectada al seu torn al CAN-Bus.
- El diagnòstic de les unitats de control LIN esclaves que porta connectades es realitza a través de la unitat de control LIN mestra.

A.3.1.2. Transmissió de dades

Com s'ha dit anteriorment la transmissió de dades és lenta, ja que, la velocitat de transmissió és de 1-20kbps en LIN. Aquesta velocitat equival a una cinquena part de la velocitat de transmissió del CAN.

A.3.1.3. Senyal

Existeixen dos tipus de nivells:

- Nivell recessiu: Si a través del LIN-Bus no es transmet cap missatge o es transmet un bit recessiu, el cable del bus té aplicada una tensió equivalent pràcticament a la de bateria.
- Nivell dominant: Per transmetre un bit dominant sobre el LIN-Bus, un transceptor en la unitat de control que efectua la transmissió connecta el cable del bus de dades a massa.

A.3.1.4. Seguritat de transmissió

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

Amb la determinació de les toleràncies per a la transmissió i recepció en la gamma dels nivells recessius i dominants s'ha de donar una transmissió estable.

La senyal ha d'estar compresa entre el 80% de la tensió i la tensió màxima per estats alts i el 20 % de la tensió i zero per estats baixos.

Per poder rebre senyals vàlides tot i existir interferències paràsites s'han configurat més amplies les gammes de tensions admissibles pel costat de la recepció. La senyal ha d'estar compresa entre el 60% de la tensió i la tensió màxima per estats alts i el 40 % de la tensió i zero per estats baixos.

A.3.2. FUNCIONAMENT DEL SISTEMA LIN-BUS

En aquest apartat s'explicarà tot el que fa referència a l'estructura del missatge i al seu funcionament.

A.3.2.1. Estructura del missatge

El missatge està dividit amb encapçalament del missatge i el contingut del missatge.

A.3.2.1.1. Encapçalament del missatge

L'encapçalament es transmès de forma cíclica per la unitat de control LIN mestra. Aquest es divideix en quatre camps:

- Pausa de sincronització (Synch break) té una longitud mínima de 13 temps per bit. Es transmet a nivell dominant. És necessària la longitud de 13 bit per indicar de manera exacta el començament d'un missatge a totes les unitats de control LIN esclaves.
- La limitació de la sincronització (Synch delimiter) té una longitud mínima d'1 bit i és recessiva.
- El camp de sincronització (Synch field) està compost per la cadena binària 0101010101. Amb aquesta seqüència de bits es poden ajustar (sincronitzar) totes les unitats de control LIN esclaves al ritme del sistema de la unitat de control LIN mestra.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- El camp de l'identificador té una longitud de 8 temps per bit. En els primers 6 bits hi ha la identificació del missatge i el nombre de camps de dades que formen la resposta.

El nombre de camps de dades on la resposta pot ser de entre 0 i 8.

Els dos últims bits reben la suma de verificació dels 6 primers bits per al detecció d'errors de transmissió. La suma de verificació es necessita per evitar que es produeixin assignacions a missatges equivocats en haver errors de transmissió de l'identificador.

A.3.2.1.2. Contingut del missatge

En el cas d'un missatge amb resposta d'esclau, una unitat de control LIN esclava afegeix informació a la resposta obeint al que especifica l'identificador.

En la següent figura es pot observar un exemple basat en el sector de l'automòbil en el que es pot observar com funciona el contingut del missatge quan LIN esclava afegeix informació a la resposta.

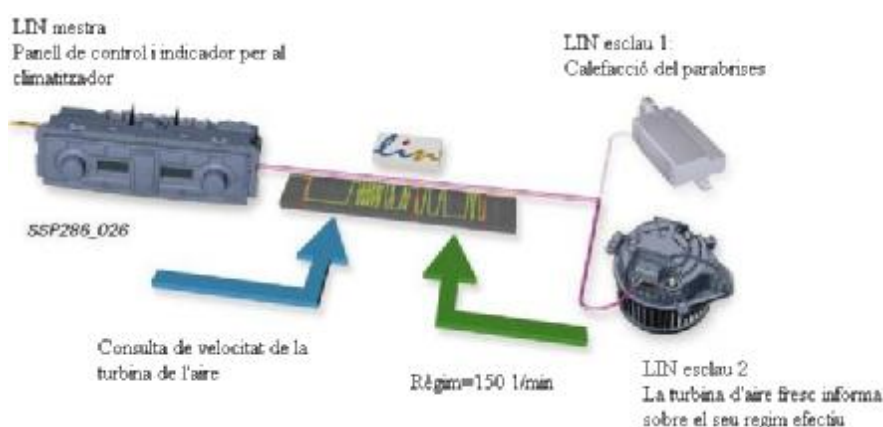


Figura A.12 Exemple d'un exemple de sistema LIN-Bus [12]

En un missatge amb sol·licitud de dades per part de la LIN mestra, aquesta última és la que afegeix la resposta.

En funció de l'especificat en l'identificador, les unitats de control LIN esclaves processen les dades i les utilitzen per a l'execució de funcions.

En la següent figura es pot observar un exemple basat en el sector de l'automòbil en el que es pot observar com funciona el contingut del missatge quan la LIN mestra es la que realitza la sol·licitud de dades.

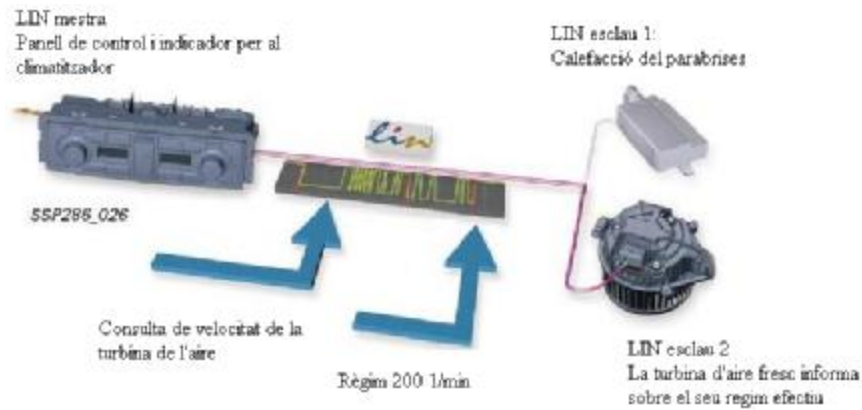


Figura A.13 Exemple d'un exemple de sistema LIN-Bus [12]

La resposta consta de 1 a 8 camps de dades (data fields). Un camp de dades consta de 10 bits. Cada camp de dades està format per un bit d'arrencada dominant, un byte de dades que conté la informació i un bit de parada. Els bits d'arrencada i parada s'utilitzen per a la resincronització i, per tant, per evitar errors de transmissió.

A.4. TARGETA CAN

Actualment, el estat del CAN el converteix en una tecnologia habitual en la indústria i nombroses firmes fabriquen i distribueixen productes compatibles amb aquest protocol de comunicació.

Entre els productes existents es troben:

- Controladors de CAN que gestionen les comunicacions a través d'aquest protocol. Es subdivideixen en:
 - Mòduls CAN integrats en el mateix chip del microcontrolador. Existeixen versions CAN amb els microcontroladors més populars del mercat.
 - Controladors CAN independents que permeten als microcontroladors no inclosos en l'anterior categoria, comunicant-se a través del CAN.
 - Targetes de connexió amb el PCs.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Transceivers (transceptors) de CAN, que funcionen amb interfase entre el controlador i els cables del bus físic a una xarxa de CAN.
- Software i altres eines diverses de monitorització de sistemes CAN, útils tant en la fase de disseny i simulació com en la fase de test.

A.4.1. CANCASE XL

Per a la connexió entre el PC i la PCB s'utilitzarà la targeta CANCASE XL que és dels tipus de controlador de CAN que s'ha explicat en el apartat anterior

En el següent apartat es realitzarà una breu introducció sobre les característiques, funcions i aplicacions de la targeta.



Figura A.14 CANcaseXL [14]

A.4.1.1. Característiques generals

Les característiques de que disposa la targeta són les següents:

- Interfase USB 2.0.
- Robusta carcassa de metall.
- Microcontrolador de 32 bits con 64 MHz.
- 2 canals completament independents.
- Els canals poden tenir dos tipus de protocol: CAN 2.0B i LIN.
- Transceptor de autobusos amb targetes plug-in (CAN-/LINpiggies).

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Font d'alimentació via USB, font d'alimentació externa opcional.
- Indicació d'estat del canal mitjançant 3 leds.

A.4.1.2. Funcions

El àmbit funcional de la CANcaseXL:

- Enviament i recepció de dades i marcs de control remot.
- Reproducció d'un bus CAN, sense influir en ell (mode silenciós).
- El reconeixement i la generació de marcs d'error.
- Buffering de missatges al dispositiu.
- Sincronització de temps.
- El funcionament simultani de diversos dispositius.

A.4.1.3. Aplicacions

Gràcies al microcontrolador d'alt rendiment i l'USB 2.0 interfície, el CANcaseXL és adequat per a aplicacions de gamma alta, és a dir, per a temps ràpids de reacció, dades d'alt rendiment, etc.

La varietat dels conductors d'autobús (CAN-/LINpiggies) permet a la CANcaseXL ser utilitzat en moltes àrees d'aplicació de CAN i LIN:

- Tecnologia per l'automòbil.
- La tecnologia de vehicles comercials.
- Tecnologia d'automatització.
- Tecnologia de l'aire i espacial.
- Tecnologia marina.

El CANcaseXL es pot utilitzar amb tots els programes de Vector, com ara CANalyzer, CANoe, i CANape. Es pot utilitzar amb qualsevol CANcaseXL adicional i amb qualsevol altre vector CAN o LIN interfície. A més, és possible utilitzar la "Biblioteca CAN Driver" per

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

crear aplicacions individuals. Per descomptat, aquestes aplicacions també es poden utilitzar amb totes les interfícies d'un altre vector.

B. TEORIA SOFTWARE

En aquest apartat s'explicarà el funcionament teòric dels dos programes principals utilitzats per a realitzar el test automàtic.

B.1. LABVIEW

LabVIEW és un entorn de programació gràfica (anomenat també llenguatge G) utilitzat per mils d'enginyers i investigadors per desenvolupar sistemes sofisticats de mesura, proves i control utilitzant icones gràfiques i cables que s'assemblen a un diagrama de flux. Ofereix una integració incomparable amb mils de dispositius de hardware i conté moltes biblioteques integrades per l'anàlisi avançat i visualització de dades, tot això, es realitza per crear una instrumentació virtual. La plataforma LabVIEW és escalable a través de múltiples objectius i sistemes operatius.

Aquest programa va ser creat per National Instruments (1976) per funcionar sobre màquines MAC i va sortir al mercat per primera vegada al 1986. En aquest moment està disponible per les plataformes Windows, UNIX, MAC i GNU/Linux. La última versió és la de 2011.

Els programes desenvolupats amb LabVIEW s'anomenen Instruments Virtuals, o VIs i el seu origen prové del control d'instruments, tot i que, avui en dia s'ha estès àmpliament no sols al control de tot tipus d'electrònica (instrumentació electrònica) sinó també a la seva programació rocosa. Alguns dels objectius és la de reduir el temps de desenvolupament d'aplicacions de tot tipus (no sols en àmbit de probes, control i disseny) i permetre la entrada a la informàtica de qualsevol altre camp.

Un dels grans avantatges que té LabVIEW és que aconseguix combinar tot tipus de software i hardware, tant del propi fabricant (com per exemple, la targeta d'adquisició que es farà servir al projecte) com d'altres fabricants (com per exemple, el software CANoe que s'utilitzarà en el projecte).

B.1.1. CARACTERÍSTIQUES PRINCIPALS

La seva principal característica és la facilitat d'ús, vàlid per programadors professionals com per a persones amb pocs coneixements de programació on poden arribar a realitzar programes relativament complexes, impossible de realitzar amb llenguatges tradicionals.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

També és molt ràpid fer programes amb LabVIEW i qualsevol programador per experimentat que sigui pot beneficiar-se d'ell.

Amb els instruments virtuals (VIs) es poden arribar a realitzar programes molt complexos que equivaldrien a milers de pàgines de codi de text amb programes tradicionals. A partir de LabVIEW 7.0 s'introdueix un nou tipus de subVI interactius anomenat VIs Exprés que tenen una configuració de caixa de diàleg que permet al usuari personalitzar la funcionalitat del VI Express. Com es podrà observar en el nostre projecte s'utilitzarà en moltes ocasions aquest tipus de VI, ja que, d'aquesta manera el programa serà més modular i serà més fàcil d'entendre.

El LabVIEW presenta múltiples facilitats en la:

- Utilització de interfícies de comunicacions, com per exemple, Port sèrie, Port paral·lel, GPIB, PXI, VXI, TCP/IP, UDP, Datasocket, Irda, Bluetooth, USB, OPC.
- Capacitat d'interactuar amb altres llenguatges i aplicacions, com per exemple, DLL (llibreries de funcions), Matlab/Simulink, AutoCAD, Solidworks, etc.
- Eines gràfiques i textuais pel processat digital de senyals.
- Visualització i maneig de gràfiques amb dades dinàmiques.
- Adquisició i tractament d'imatges.
- Control de moviment.
- Programació de FPGAs pel control o validació.
- Sincronització entre dispositius.

B.1.1.1. Estructura LabVIEW

LabVIEW al ser una eina de programació gràfica, els programes no s'escriuen sinó que es dibuixen i quan s'inicialitza el programa ja es té una gran quantitat de blocs predeterminats construïts. Aquests blocs estan en el apartat de control del programa i es poden seleccionar quan és prem el botó dret.

Cada VI consta de dos parts diferencials:

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Panel Frontal: El Panell Frontal és la interfície amb l'usuari, s'utilitza per interactuar amb l'usuari quan el programa s'està executant. Els usuaris podran manipular les dades i observar les dades del programa actualitzades en temps real.
- Diagrama de blocs: És el programa pròpiament dit, on es defineix la seva funcionalitat, aquí es col·loquen icones que realitzen una determinada funció i s'interconnecten entre elles. En aquest panell es pot trobar tot tipus de controls i indicadors, on cada un d'aquests elements queda assignat en el diagrama de blocs un terminal, és a dir, el usuari podrà dissenyar un projecte en el panel frontal amb controladors i indicadors, on aquests elements seran les entrades i sortides que interactuïn amb el terminal del VI.

En el panell frontal es pot trobar tot tipus de controls i indicadors, on cada un d'aquests elements queda assignat en el diagrama de blocs un terminal, és a dir, el usuari podrà dissenyar un projecte en el panel frontal amb controladors i indicadors, on aquests elements seran les entrades i sortides que interactuïn amb el terminal del VI.

En la següent figura es pot observar quina forma té un programa Labview. En aquest exemple, es pot observar un programa on es pot definir l'adreça del instrument, la tensió i la intensitat (per a un diferent tipus de rang) que es vol aplicar a l'aparell Agilent E3633a (font de tensió auxiliar).

Com es pot observar en el diagrama de blocs, s'utilitzarà la interfície de comunicació GPIB i per tant, s'utilitzarà diferents tipus de blocs que ja estan definits al programa i que s'utilitzen per interconnectar amb l'aparell mitjançant aquesta interfície.

LabVIEW té una eina d'ajuda molt important i molt útil per saber quina funció realitza cada bloc i quines sortides i entrades té cada bloc. Aquesta eina es casi imprescindible alhora de realitzar un programa una mica complex.

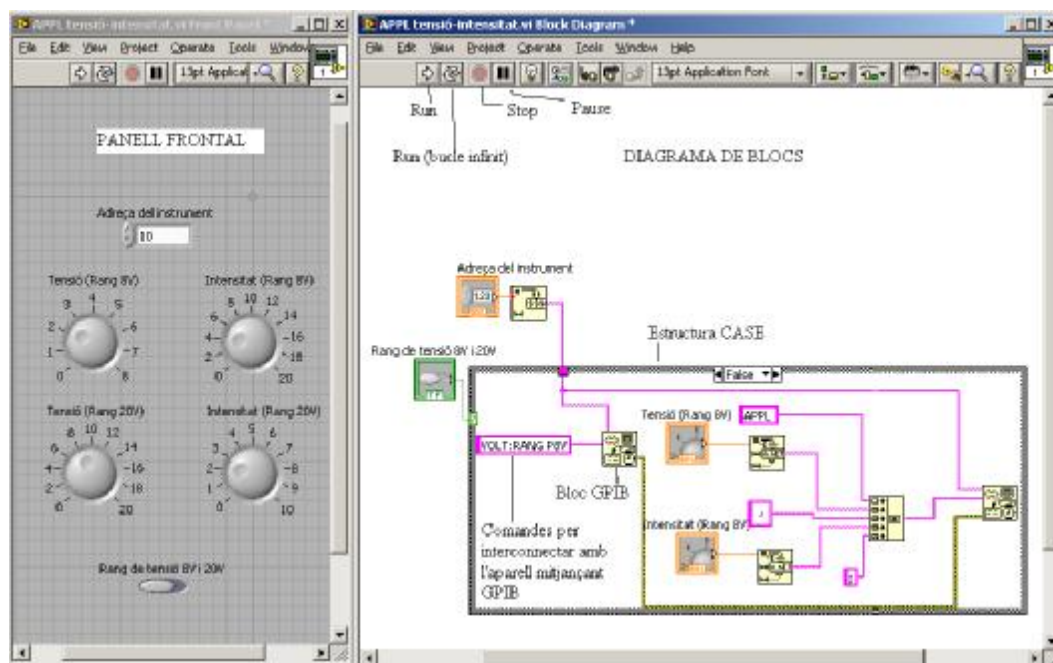


Figura B.1 Esquema LabVIEW

B.1.2. APLICACIONES

El programa LabVIEW es utilitzat en moltes i diferents taques però les més generals són:

- Adquisició de dades i anàlisi matemàtic.
- Comunicació i control d'instruments de qualsevol fabricant.
- Disseny de controladors: simulació i validació.
- Disseny de microprocessadors i chips.
- Control i supervisió de procés.
- Visió artificial i control de moviment.
- Robòtica.
- Domòtica i xarxes de sensors inalàmbrics.

B.1.3. DRIVERS LabVIEW

Per la comunicació dels PC amb els dispositius utilitzats en el projecte s'utilitzaran diferents drivers per tal de facilitar i comprimir la programació amb LabVIEW:

- Per a la targeta d'adquisició de dades s'utilitzaran els drivers DAQmx.
- Per als dispositius connectats mitjançant GPIB s'utilitzaran els drivers NI-VISA.

En aquest apartat s'explicarà de forma molt breu els diferents drivers per tal d'entendre el seu funcionament.

B.1.3.1. DAQ en LabVIEW

LabVIEW proporciona tres tipus de drivers per l'adquisició de dades:

- Les Traditional NI-DAQ són les de més baix nivell, utilitzades en versions antigues de LabVIEW, actualment es conserven per raons de compatibilitat. National Instruments solament aconsella utilitzar-los amb alguns dispositius antics i en situacions molt concretes. No presenten la capacitat de multiprocessat que tenen les noves versions.
- Els VIs Express es van introduir en LabVIEW 7 i fan la programació molt més senzilla i compacta que amb els VIs normals al agrupar a varis VIs en un únic node molt configurable.
- Els NI-DAQmx també s'ha introduït amb la versió 7 de LabVIEW per Windows i són les que s'utilitzaran per controlar la targeta d'adquisició. Presenten molts avantatges respecte a les tradicionals, com per exemple, major integració amb MAX, un nivell de abstracció major, multiprocés (multithreaded), més robust, més estabilitat, etc. La senzillesa l'aconsegueixen bàsicament a través de les VIs (agrupació de varis VIs diferents amb un únic fitxer i icona), amb el que s'aconsegueix agrupar les tasques que fan varis VIs tradicionals en un de sol.

La següent figura és una comparació sobre un programa que llegeix una entrada analògica de les tres formes mencionades anteriorment.

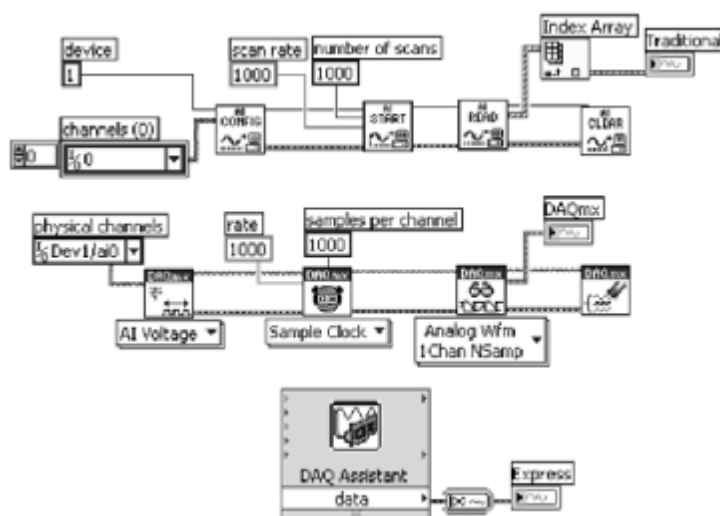


Figura B.2 Comparativa de codi [16]

B.1.3.1.1. DAQmx

DAQmx proporciona una llibreria o API (Application Programming Interface) que facilita la comunicació amb targetes d'adquisició de dades.

No es recomana utilitzar Traditional DAQ i DAQmx en el mateix programa, ja que, si s'utilitza un Traditional, els DAQmx posteriors generen errors fins que el dispositiu sigui resetejat. I si es crea una tasca sobre el dispositiu amb DAQmx, s'ha de borrar la tasca abans d'utilitzar els Traditional.

El mètode per treballar amb DAQmx es pot resumir en la següent figura: crear tasques, configurar temporitzador i dispar, llegir o escriure i borrar la tasca.

Alguns conceptes que utilitzen aquests VIs són canals virtuals i tasques. Un canal virtual és una col·lecció de línies o ports físics que s'agrupen per realitzar la mateixa tasca i informació associada, com la configuració o les escales. Una tasca és una mica més general, representa la mesura o generació de senyal que es vol realitzar, agrupa els canals, timing, triggering i altres propietats.

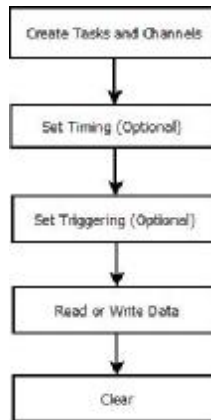


Figura B.3 Flux de programa per treballar amb DAQmx [16]

La paleta es troba en NI Measurements > DAQmx Data Acquisition

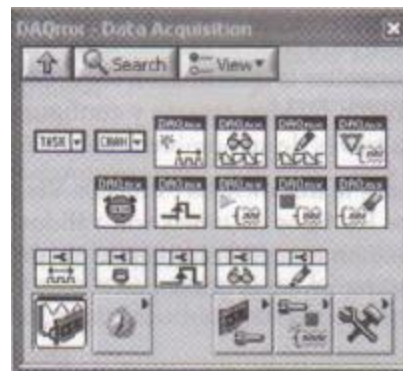


Figura B.4 Funcions per treballar amb targetes DAQ [16]

Aquests VIs també tenen una agrupació lògica en la paleta:

- Dalt a l'esquerra estan les constants per crear canals i tasques.
- Les VIs de la primera i segona fila són les principals.
- La tercera fila són els nodes de propietats.
- La última fila està l'assistent i 'subpaletes' d'utilitats que contenen funcions com autocalibració, reset, connexió interna de unes línies amb altres, etc.

A continuació, s'explicarà les VIs més importants i/o generals.

B.1.3.1.2. DAQmx Create Virtual Channel:

Com la majoria dels DAQmx, aquest VI és polimòrfic i la instància a utilitzar es seleccionada mitjançant una llista desplegable que apareix baix a la icona. En la imatge anterior es

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

mostra els terminals per si es vol configurar una entrada analògica de tensió. La seva funció es crear un canal virtual i afegir-lo a una tasca, si no s'especifica una tasca també la crea automàticament. Al crear una constant o un control per a seleccionar els canals (terminal physical channels) ha d'aparèixer una llista desplegable en la que es mostra els que estan disponibles, també es poden escriure amb el format <<Dev1/ai0>>, on 1 és el número de dispositius i ai0 la línia i per utilitzar més d'una línia es separen per comes. A més a més, es poden filtrar el tipus de línees que mostra la llista amb l'opció I/O Name Filtering del menú desplegable. També es poden acceptar un string amb el mateix contingut.

Sobre el VI particular que s'ha mostrat anteriorment s'ha de comentar que el input terminal configuration serveix per indicar el tipus de mesura: diferencial RSE i NSRE. A l'ajuda es poden consultar la informació sobre cada una de les instàncies que componen aquest VI. També es pot crear i configurar tasques mitjançant un assistent i accedir a elles a través del seu nom (ver MAX i DAQ Assistant Express).

A la llista per seleccionar la instància hi ha molt VIs per elegir. Per una determinada targeta no tots els tipus de tasques són vàlids. Per últim, comentar que una tasca creada mitjançant aquest VI també es poden guardar per poder modificar-se desde MAX o utilitzar en altres ocasions.

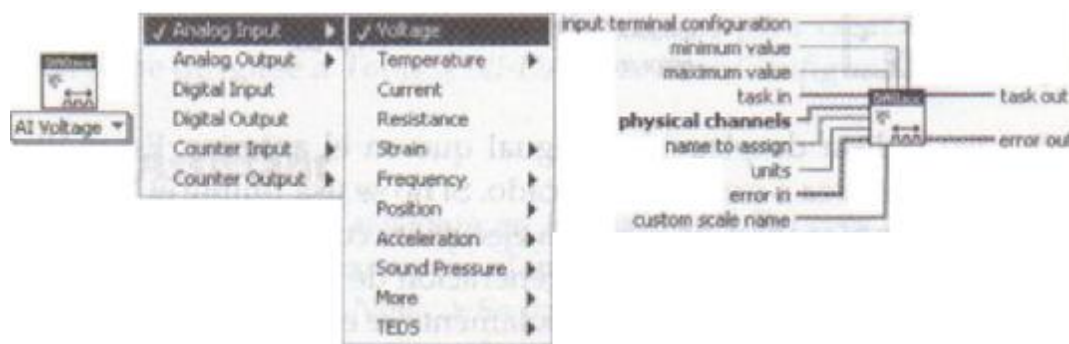


Figura B.5 DAQmx Create Virtual Channel [16]

B.1.3.1.3. DAQmx Read

Llegeix mostres pel canal o una tasca especificada, aquesta pot haver sigut generada amb el VI anterior o amb un MAX. Al seleccionar una instància en concret es determina el format de la lectura: una simple dada, arrays, etc. Totes les instàncies inclouen un paràmetre de timeout i, en les que es poden aplicar, un altre paràmetre per indicar el número de samples per canal.



Figura B.6 DAQmx Read [16]

B.1.3.1.4. DAQmx Write

En aquest cas la llista desplegable és igual que en l'anterior. Aquests VIs escriuen mostres per la tasca o canal especificat. Si no s'utilitza timing el VI espera a generar tots els samples abans de seguir l'execució, amb timing s'escriuen en el buffer i ja no es necessària la espera. La generació de la senyal pot començar immediatament o retardar-se, en tot cas, solament s'escriuen les dades, no es treuen al exterior.

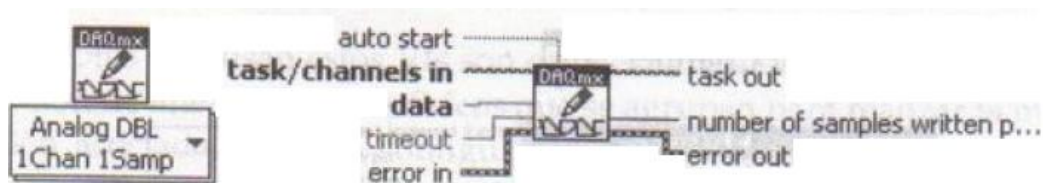


Figura B.7 DAQmx Write [16]

B.1.3.1.5. DAQmx Trigger

Configura la forma en que comença una tasca, és a dir, la seva condició de dispar. Un trigger pot tenir diverses fonts, tant analògiques com digitals, internes a la targeta o externes. Un trigger de tipus start indica la condició de inici de tasca i el tipus reference indica la condició de parada.

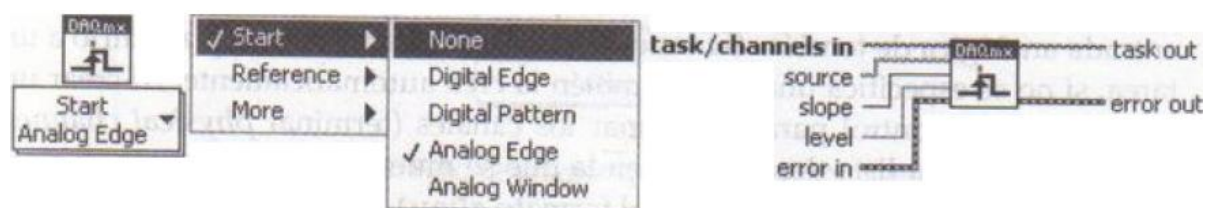


Figura B.8 DAQmx Trigger [16]

B.1.3.1.6. DAQmx Timing

Configura els paràmetres relatius al temps en una tasca i crea un buffer quan es necessari. Aquests paràmetres són, entra altres, els samples per segon i un terminal per utilitzar amb rellotge si la execució ha de ser continua o finita.

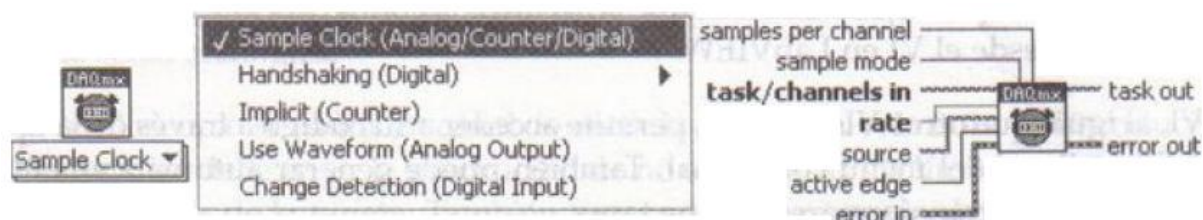


Figura B.9 DAQmx Timing [16]

B.1.3.1.7. DAQmx Clear Task

La seva funció és parar i esborrar la tasca. És útil quan es crea una tasca dins d'un bucle.

A més a més, d'aquests VIs i altres, DAQmx també tenen un gran número de propietats amb les que es poden tenir un gran nivell de control sobre la targeta.

NI-DAQmx Base és un altre API per adquisició de dades que no han de confondre amb NI-DAQmx, tot i que, externament són VIs molt semblants a les de NI-DAQmx, el mètode de programació és el mateix i amb el nom es pot arribar a la confusió. Un petit detall és que les icones de DAQmx Base tenen en la seva part superior les lletres <<mxBase>>, mentre que les "normals" tenen <<DAQmx>>. Internament es basen en APIs totalment diferents, és a dir, per crear tasques en lloc del MAX s'han de dirigir a Tools>NI-DAQmx Base Configuration Utility.



Figura B.10 DAQmx Clear Task [16]

B.1.3.2. VISA

VISA (Virtual Instrument Software Architecture) és un API o llibreria desenvolupada per varis fabricants d'equips que proporciona un estàndard software per les operacions de lectura i escriptura en el camp de la instrumentació.

NI-VISA és la implementació de National Instruments d'aquest estàndard i pot establir comunicacions a través de GPIB, sèrie, PXI, VXI o Ethernet. En les direccions dels dispositius es farà referència al tipus de comunicació i al dispositiu.

En aquest projecte, s'utilitzarà aquesta implementació (NI-VISA) per a les configuracions de tots els nostres dispositius que estan connectats via GPIB.

NI-VISA ve acompanyat d'una sèrie d'utilitats com:

- VISA Driver Development Wizard (DDW): Assistent que permet crear un driver per un dispositiu PXI, PCI,USB o firewire que al instal·lar-lo a Windows, habilita a NI-VISA per accedir a aquest dispositiu.
- VISA Interactive Control (VISAIC): és un assistent per provar d'una forma ràpida i senzilla la connectivitat i funcionalitat dels dispositius mitjançant les funcions de NI-VISA.
- VISA Server: és un servidor que permet controlar de forma remota un equip a través de VISA. El servidor ha de ser executat en la màquina on està connectat l'equip.

NI-Spy: es tracta d'un programa per ajudar a la depuració en baix nivell. Permet capturar les crides que es realitzen a les funcions del API i les respostes que produeixin.

Mitjançant NI-VISA també es pot accedir a dispositius d'altres fabricants creant i instal·lant un driver a través de VISA Driver Development Wizard. Per testejar el funcionament de la comunicació es pot aplicar MAX o VISA Interactive Control.

Com VISA suporta varis interfaces de comunicació sol ser el mètode escollit per crear drivers de control d'equips. El model d'un programa en LabVIEW per a un driver d'un instrument es representa en la següent figura.

Hi ha un assistent en File> New...> Project > Instrument Driver Project o en Tools > Instrumentation>Create Instrument Driver Project que crea el esquelet d'un projecte en el que es pot programar el driver d'un instrument utilitzant NI-VISA i seguint el esquema anterior.

Les funcions del API NI-VISA utilitzades en LabVIEW estan en el menú Instrument I/O > Visa.

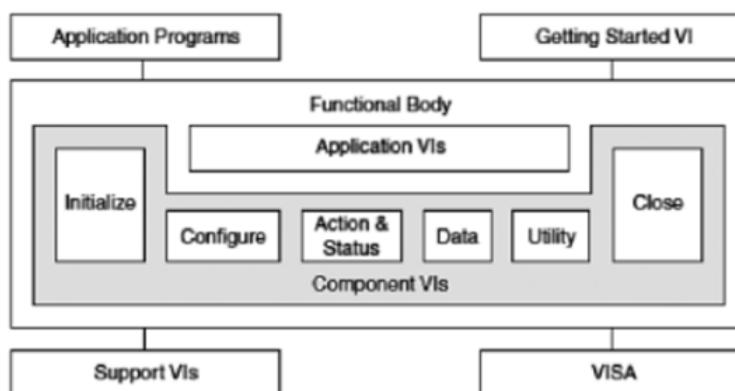


Figura B.11 Esquema dels drivers dels instruments [16]



Figura B.12 Menú VISA [16]

Existeixen moltes VIs dedicades a VISA per això no se estudiaran amb detall, simplement dir que els més utilitzats i típics són VISA Open, VISA Write, VISA Read i VISA Close.

Les direccions s'han d'indicar en un control o constant de tipus VISA resource name. Aquest control o constant pot canviar de classe pressionant amb el botó secundari del ratolí sobre una constant o control i anant al menú Select VISA Class. Cada classe tindrà unes propietats i mètodes diferents.

B.1.3.3. Active X Functions

Active X functions és una llibreria on les seves funcions o blocs, anomenades d'ActiveX, serveixen per interconnectar-se amb altres aplicacions com Microsoft Excel, CANoe, entre d'altres programes.

En aquest projecte s'utilitzarà les Active X Functions del programa CANoe 7.1 per realitzar la interfície CANoe/LabVIEW i Microsoft Excel 12.0 per realitzar el report.

B.1.3.3.1. Automation refnum

S'utilitza per obrir una referència a un objecte de les funcions de ActiveX Server.



Figura B.13 Automation Refnum [20]

B.1.3.3.2. Event callback refnum

S'utilitza per anul·lar el registre o tornar a registrar la funció callback de l'esdeveniment. També es pot utilitzar aquest refnum per passar el registre de devolució de crida d'un esdeveniment a una subVI.



Figura B.14 Event Callback Refnum [20]

B.1.3.3.3. Automation open function

Serveix per obrir un automation refnum i retorna un refnum d'automatització, que apunta a un objecte específic d'ActiveX.

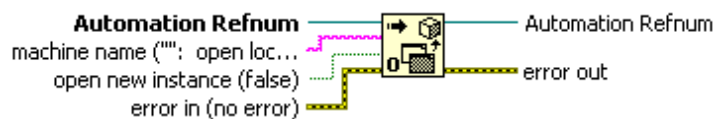


Figura B.15 Automation Open Function [20]

B.1.3.3.4. Variant to data function

Converteix les dades de la variant a un tipus de dades de LabVIEW per LabVIEW, per poder mostrar o processar les dades. També es pot utilitzar aquesta funció per convertir les dades de la variant a les dades d'ActiveX.



Figura B.16 Variant to data function [20]

B.1.3.3.5. Close reference function

Tanca una referència associat amb un procés obert VI o un objecte ActiveX.

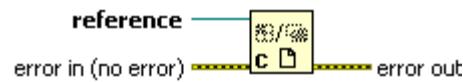


Figura B.17 Close Reference Function [20]

B.1.3.3.6. Invoke node

Invoca un mètode o una acció en una referència. LabVIEW inclou Invoke nodes preconfigurats per accedir als mètodes XML, .NET i els mètodes de ActiveX.

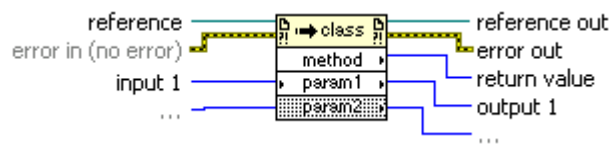


Figura B.18 Invoke node [20]

Per crear un Invoke node per una interfície Labview-CANoe o Labview-Excel s'ha de seguir els següents passos:

- Crear Invoke Node: Botó dret (en el Block Diagram)-> Programming->Application Control-> Invoke Node.

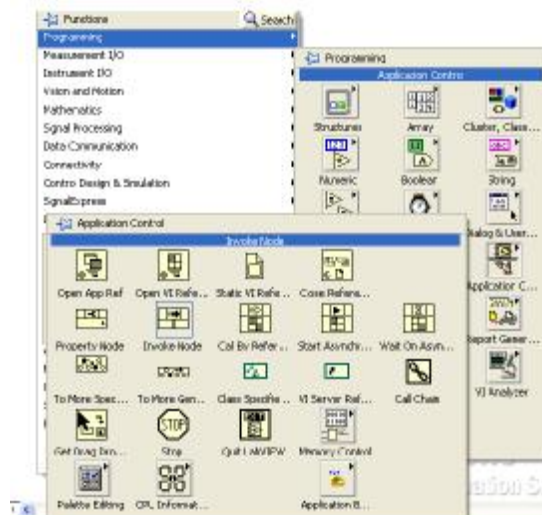


Figura B.19 Crear Invoke Node [20]

- Seleccionar la classe: Botó dret damunt del invoke node-> select class-ActiveX->Browse.

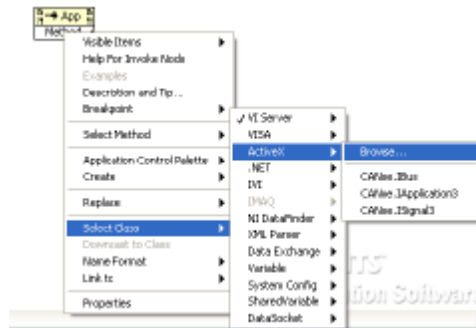


Figura B.20 Seleccionar classe [20]

- Buscar llibreria de CANoe-> Una de les llibreries que s'utilitzarà és la del CANoe 7.1 Type Library Version 7.1.

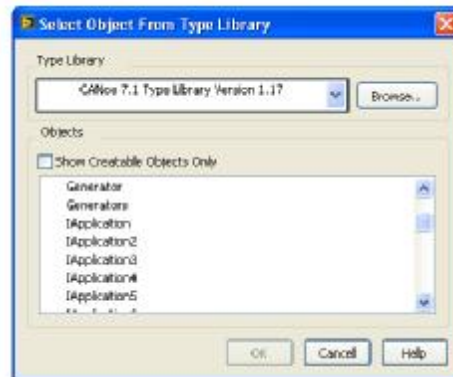


Figura B.21 Buscar llibreria [20]

- Seleccionar objecte. Depenén de les necessitats es seleccionarà el tipus d'objecte.
- Configurar el objecte. Depenén de les necessitats es configurarà el objecte prement el botó esquerre damunt d'aquest i prement l'opció desitjada.



Figura B.22 Exemple de configurar el objecte [20]

B.1.3.3.7. Property node

Llegeix i/o escriu les propietats d'una referència.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

El node de propietat s'adapta automàticament a la classe de l'objecte que fa referència. LabVIEW inclou nodes de propietat preconfigurats per accedir a les propietats.

Per crear un property node per una interfície Labview-CANoe o Labview-Excel s'ha de seguir els mateixos passos que en un Invoke Node:

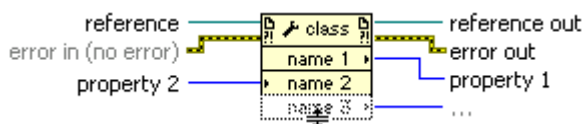


Figura B.23 Property node [20]

B.1.3.3.8. REGISTER EVENT CALLBACK

Realitza la crida d'un esdeveniment o subVI. Aquesta funció s'utilitza per gestionar i registrar.

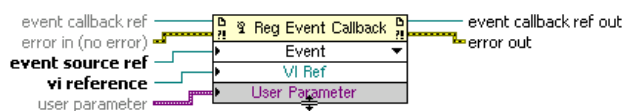


Figura B.24 Register Event Callback [20]

B.1.3.3.9. Unregister event callback

Tanca la crida (tots els registres) d'un esdeveniment o subVI.

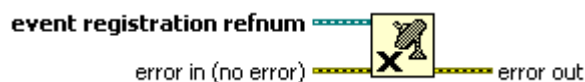


Figura B.25 Unregister Event Callback [20]

B.1.3.4. Llibreria VISA

És una llibreria on les seves funcions o blocs serveixen per configurar tots els dispositius connectats mitjançant GPIB.

Aquesta llibreria està situada a Functions->Instrument I/O -> VISA.

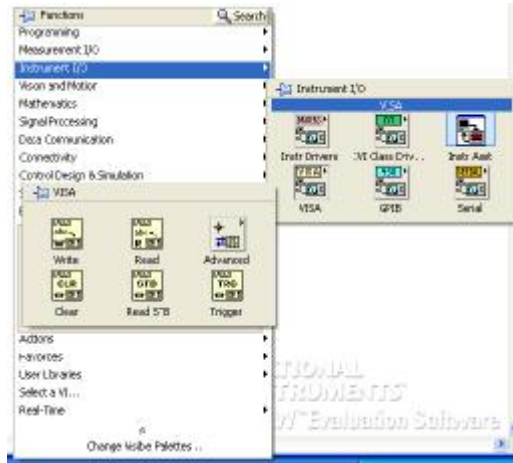


Figura B.26 Property node [20]

Els diferents blocs que s'utilitzaran en el projecte són els següents:

B.1.3.4.1. Write VISA

Serveix per enviar comandes i ordres als dispositius.



Figura B.27 Write Visa [20]

B.1.3.4.2. Read VISA

Serveix per llegir les lectures i les mesures dels dispositius.

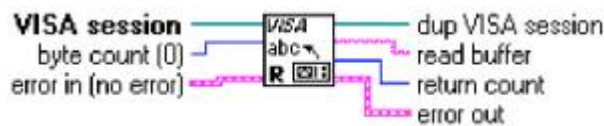


Figura B.28 Read Visa [20]

B.1.3.4.3. VISA find resource

Serveix per buscar les adreces dels dispositius que estan connectats mitjançant GPIB.



Figura B.29 Visa Find Resource [20]

B.1.3.4.4. Clear VISA

Borra la entrada i les sortides dels buffers del dispositiu seleccionat.



Figura B.30 Clear Visa [20]

B.2. CANOE (CAN Open Environment)

En la indústria automobilística, la comunicació mitjançant el bus CAN és àmpliament utilitzada i gràcies a la eina CANoe es pot desenvolupar, analitzar i simular diferents entorns per a la comunicació del bus CAN. Per altra banda, amb la eina CANoe es pot tenir la oportunitat de realitzar diferents simulacions sense tenir els mòduls físics.

En el projecte, el software d'aquest programa el subministrarà el departament de software de Lear Corporation i en l'apartat d'estructura del software s'explicarà com s'ha realitzat la interfície LabVIEW-CANoe per tal de poder manipular aquest programa a través de LabVIEW.

En aquest apartat, s'explicarà de forma teòrica, com funciona i com està format aquest programa.

B.2.1. ARQUITECTURA DEL CANOE

Per al correcte funcionament, el programa CANoe està format per 4 executables

- Editor database: Amb aquest programa es pot crear un arxiu (.dbc) que conté tota la informació simbòlica, és a dir, és on s'inclou tot el que fa referència a els nodes de la xarxa, tots els missatges que s'utilitzen, les senyals i les variables de entorn.
- El CAPL Browser (.can) s'utilitza per crear i compilar els programes CAPL a través de diferents funcions per a cada node. Aquest executable conté els missatges, les variables d'entorn que ja han estat definides al database o també es poden definir però automàticament es defineixen al nou database.
- El editor Panel (.cnp) s'utilitza per crear panells gràfics que són carregats posteriorment al CANoe i representen la interfase entre el usuari i la els nodes dissenyats anteriorment.

- L'aplicació CANoe .exe crea un arxiu .cfg i és el que simula tot el programa CAN i s'encarrega de associar tots els executables anteriors.

En la següent figura es pot observar tota l'arquitectura d'un programa CANoe.

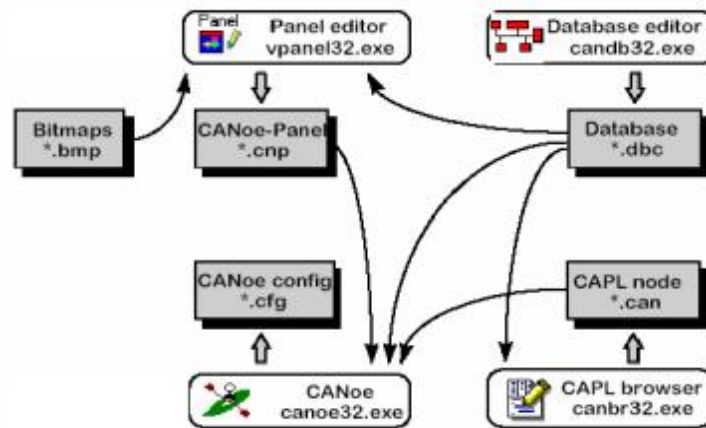


Figura B.31 Arquitectura del programa CANoe [22]

B.2.2. EDITOR DE BASE DE DATOS CANdb++

Totes les dades rellevants d'un sistema de comunicació processats en una xarxa de bus CAN, normalment són administrats per una base de dades central de comunicació.

CANdb++ és un programa creat per la empresa Vector Informatik per editar i gestionar bases de dades de comunicació. Aquest programa actualment està disponible en dos versions diferents: estàndard (CANdb++) i la versió més amplia (CANdb++ Admin).

Tot i que, en aquest projecte no farà falta dissenyar una base de dades, ja que, BMW ja haurà subministrat la seva que fa referència al model SboxSME_BK2.0, s'explicarà els diferents apartats que té un arxiu .dbc, per tal de familiaritzar-nos amb el programa.

B.2.2.1. Finestra principal

En la figura següent es pot observar la finestra i la forma que té un arxiu .dbc. Com es pot observar en la finestra, a l'esquerra estan classificats tots els tipus d'objectes que existeixen en un sistema de comunicació i a la dreta hi ha la descripció dels objectes "Messages" que són el paràmetre activat.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Node Groups: Múltiples controladors de bus poden combinar-se dins d'un grup de nodes. Aquests permeten estructurar variants de xarxes que s'aplicaran depenent del nivell de funcions del vehicle. Aquest objecte només es pot trobar a la versió més amplia de CANdb++.
- Network Nodes (NN): Aquests dispositius són la interfase de comunicació entre la unitat de control i el bus CAN. En el cas de que un node sigui definit, automàticament es crea una unitat de control amb el mateix nom i es assignat a una de les xarxes existents.
- Messages: Aquest són els que circulen entre network nodes. Estan formats pel identificador, nombre de dades, tipus de transmissió, control d'error i bytes de dades.
- Signals: Els bytes de cada missatge poden ser dividits en una o més senyals, depenent de la informació a enviar.

En la següent figura es pot observar l'arquitectura general d'un vehicle.

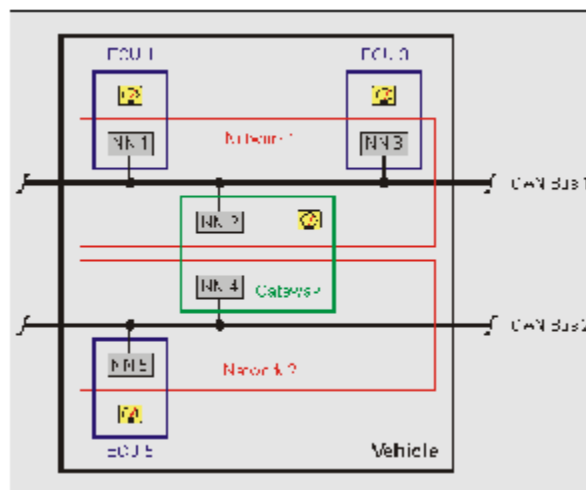


Figura B.33 Arquitectura simplificada d'un vehicle [22]

B.2.3. CAPL BROWSER

El CAPL és un llenguatge de procediment on la execució dels blocs del programa estan controlats per "bevents" i on cada "avant", comporta la execució d'un procediment.

Com és pot observar en el següent gràfic els tipus de bevents que pot ser detectats en el CAPL són:

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Quan és prem el boto de start.
- Quan és prem el boto stop.
- Quan és prem qualsevol tecla.
- Quan es rep un CAN message.
- Quan s'activa un timer.
- Utilitzar una entrada del graphic panel (aplicable només al CANoe).

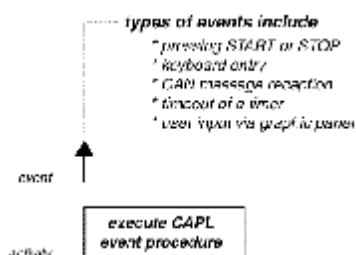


Figura B.34 Tipus de events utilitzats en el CAPL [22]

Es pot dividir el programa CAPL en dos parts:

- Quan es declarà les variables.
- Quan es declarà les funcions i procediments de resposta a events.

B.2.3.1. Declaració de variables

Els tipus de dades a utilitzar com a variables en CAPL inclouen enters (dword, long, word, int, byte i char) i números amb coma flotant (float i double) com el llenguatge C. També hi ha variables pròpies per a utilitzar missatge de CAN declarant-les com a message o timer.

Totes les variables excepte les de tipus timer es poden inicialitzar amb un valor determinat en el moment de declarar-les.

```
Variables{

timer Timer1 //declaració de un timer amb base en segons

mstimer Timer2 //declaració de un timer amb base en milisegons

}
```

Les variables de tipus timer han de ser inicialitzades mitjançant la funció:

```
setTimer(nom del timer, segons o milisegons en el que tardarà a generar un event)
```

Pel que a la declaració del missatge s'ha de declarar el identificador del missatge o el nom del missatge definit en la base de dades. Com per exemple:

```
message Precharge msg1; //el missatge msg1 es vinculat al nombre simbòlic
Precharge
```

B.2.3.2. Procediment d'events

Al produir-se un esdeveniment, el bloc del programa s'executa i el procediment involucrat en aquest event, genera una resposta de tipus missatge. Dos procediments d'events amb el mateix nom no es pot fer en CAPL. Com ja s'ha mencionat anteriorment, es pot reaccionar per a diferents tipus de events, en la següent taula s'indicarà els procediments en CAPL i a quin esdeveniment es refereixen.

Esdeveniments en format CAPL	Definició
On timer{}	Final de una temporització
On errorFrame{}	Recepció de un Error Frame
On message{}	Recepció d'un missatge de CAN
On key {}	Pressió d'una tecla
On prestart{}	Inicialització de la mesura
On start{}	Al inici de la mesura
On stopMeasurement {}	Al final de la mesura
On errorActive{}	Controlador de CAN en Error Actiu
On errorPassive{}	Controlador de CAN en Error Passiu
On warningLimit{}	Controlador de CAN al límit

On envVar{}	Canvi del valor de una variable de entorn
On busOff{}	Controlador de CAN en Bus Off

Taula B.1 Tipus d'esdeveniments en el CAPL [22]

Per altra banda també existeixen les funcions `getValue()` i `putVaule()` que s'utilitzen un cop dins de la execució del programa i serveixen per obtenir el valor de la variable activa i per modificar.

B.2.4. EDITOR PANEL

Aquest executable no es farà servir per al nostre projecte, ja que, es controlarà desde un panell LabVIEW, però per altra banda s'explicarà de forma molt breu que es fa en aquest executable.

Amb l'editor panel es pot crear o modificar tots els panel gràfics .cnp. Existeixen diferents tipus de variables que es poden crear, com per exemple, botons, switch, leds, etc. Amb el panells creats, el usuari pot variar els valors de les variables d'entorn interactiva ment durant la execució del sistema. Aquestes variables han d'estar definides prèviament en la base de dades.

Quan s'executa el editor de panells des de el CANoe, les variables d'entorn de la base de dades activa són associades al editor de panells.

Els elements que s'han d'incerta en el panell són anomenats controls. Existeixen tres tipus de controls: elements de visualització, que mostren el estat de les variables d'entorn, altres que les modifiquen i finalment, altres que poden ser utilitzades per a mostrar i modificar el seu valor.

En la següent figura es pot observar un exemple de un panell.

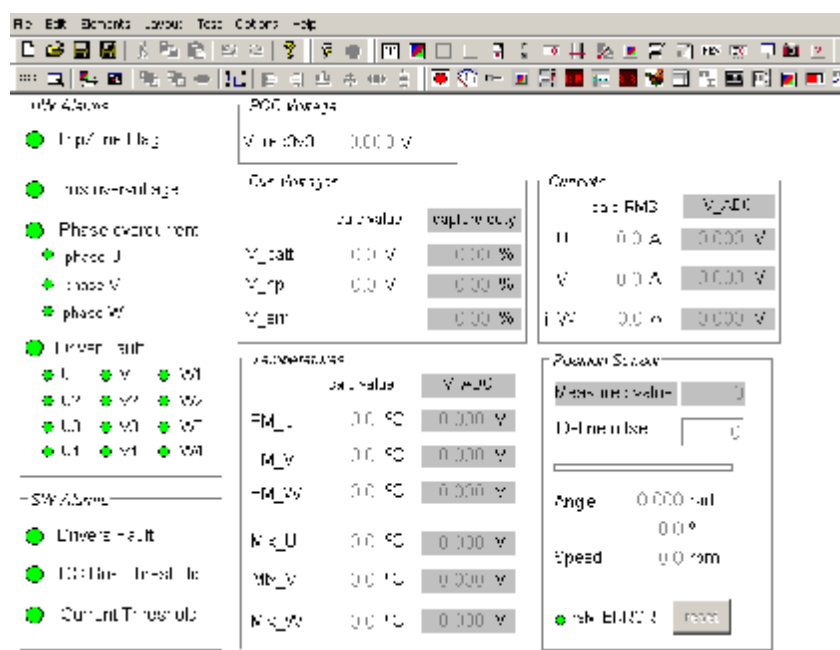


Figura B.35 Esquema d'un panell [22]

B.2.5. PROGRAMA CANoe (Arxiu .cfg)

Com s'ha comentat en altres apartats, l'aplicació CANoe es utilitza per a mesurar i simular sistemes de CAN. Per crear aquests sistemes el programa utilitza les dades creades en els programes anteriors i les associa entre si.

En la següent figura es mostra com és un arxiu .cfg, també es pot observar en la barra d'eines de la part superior els botons start i stop.

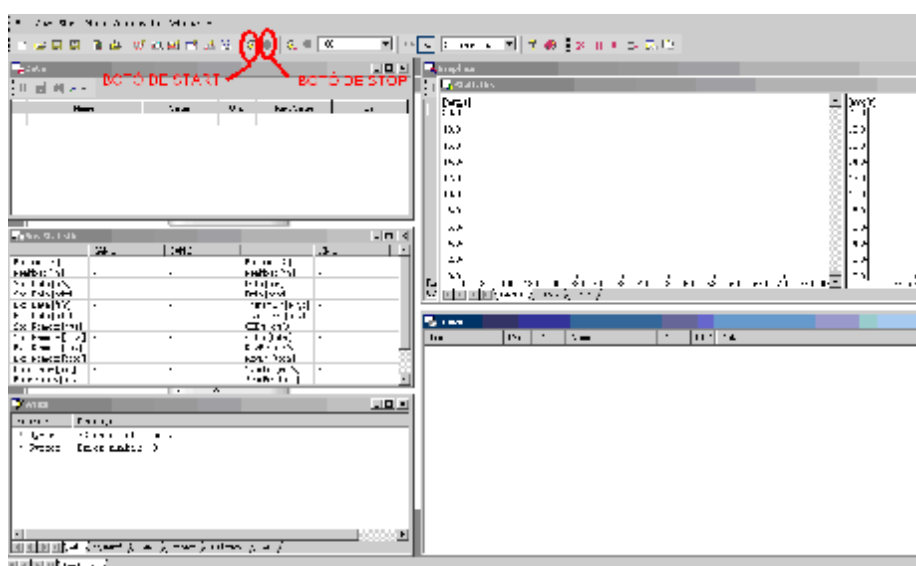


Figura B.36 Finestra d'un arxiu .cfg [22]

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

Aquest programa està format per tres finestres principals per a que els usuaris pugin configurar i visualitzar el sistema desitjat.

B.2.5.1. Configuració de la simulació (Simulation Setup)

Configuració de la simulació (Simulation Setup). Amb aquesta finestra es configura la xarxa de CAN que es desitja simular, afegint els nodes i els databases que tenen que participar. Aquests nodes poden ser de 2 tipus: reals o simulats.

B.2.5.1.1. Node de xarxa

Un node de xarxa és un node que porta associat un programa en llenguatge CAPL, un llenguatge propi de CANoe, obtenint així un bloc funcional. Les característiques d'aquests blocs són definits pel usuari mitjançant un programa en CAPL.

Aquests nodes, conjuntament amb els reals, són els que defineixen la funcionalitat del sistema. Aquesta funcionalitat inclou el comportament de un node respecte les variables d'entrada i sortida, així com els missatges a rebre i a enviar.

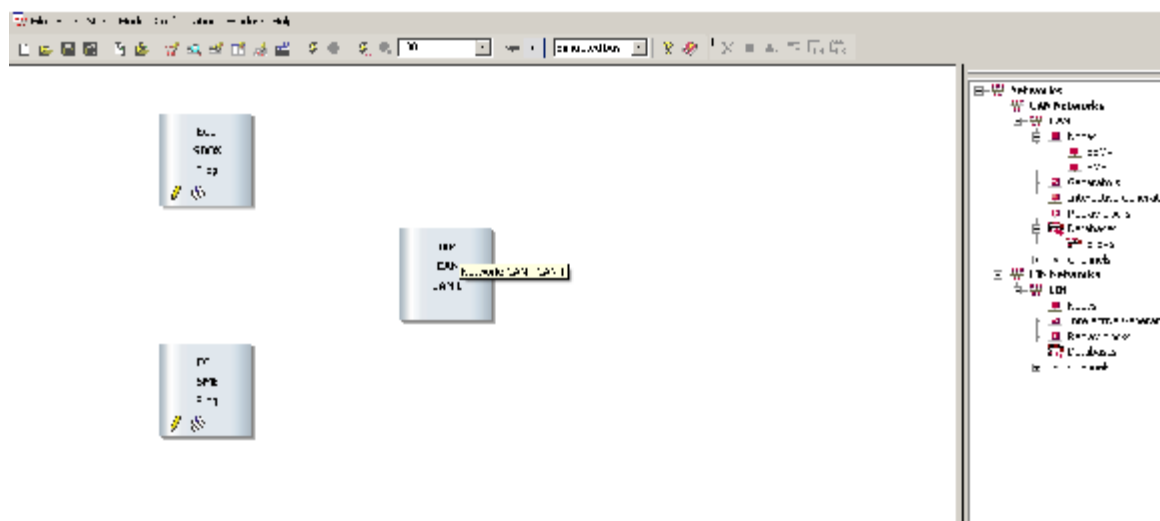


Figura B.37 Assignació d'un node de la base de dades

B.2.5.2. Configuració de la mesura (Measurement Setup)

La monitorització i la evolució de les dades que circulen per la xarxa es configuren a partir d'aquesta finestra, ja que des d'aquí, es poden accedir a les següents finestres: Estadística (Statistics), Estadística del Bus (Bus Statistics), Monitorització de trames (Trace), Dades (Data) i gràfics (Graphics), a més de disposar de una opció per emmagatzemar (Logging) les dades en un fitxer tipus ASCII o binari.

B.2.5.2.1. Monitorització de trames

Les trames que circulen per la xarxa de CAN es monitoritzen en la següent finestra en temps real. Per aconseguir una visualització d'un conjunt de missatges determinat, es poden utilitzar els filtres comentats anteriorment.

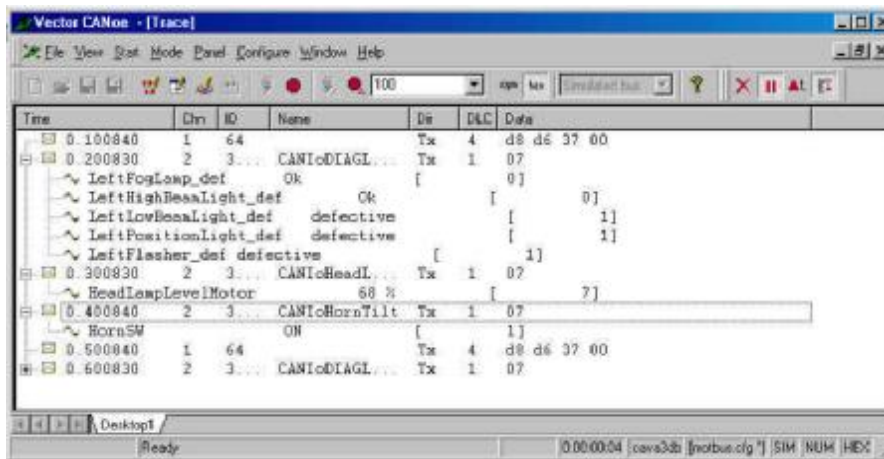


Figura B.38 Finestra de monitorització de trames

La finestra de la figura anterior es pot visualitzar de diferents mode, segons el paràmetre de temps escollit:

- Mode cronològic: els missatges són mostrats ordenadament segons el temps d'arribada.
- Mode cronològic amb actualització cíclica: els missatges són mostrats ordenadament com en el cas anterior, però l'actualització de la pantalla es realitza per un conjunt de missatges, estalviant temps d'operació.
- Mode de posició fixa: A cada missatge amb el mateix ID se l'hi assigna una posició en la pantalla. Quan arriba un nou missatge aquest reemplaça el anterior.
- Mode de posició fixa amb l'actualització cíclica: el tipus de funcionament s'assembla al anterior però l'actualització de la pantalla es realitza per un conjunt de missatges estalviant temps d'operació.

B.2.5.2.2. Dades

La finestra de dades serveix per mostrar els valors de les senyals d'una manera més senzilla. Quan la senyal visualitzada a una base de dades, la unitat especificada en la base de dades es mostrada junt amb la senyal.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

Els valors obtinguts en el últim missatge rebut, estan visibles fins que no sigui reemplaçat per valors de un nou missatge. En el cas de que el nou missatge contingui valors igual al missatge anterior, una barra anirà variant per mostrar aquest fet.

Com en altres casos, per permetre el procés de grans quantitats de dades s'hauria d'activar l'opció d'actualització cíclica, tot i que, per obtenir resultats raonables, el període ha de ser menor de 500ms.

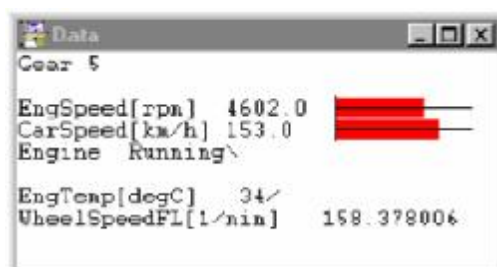


Figura B.39 Finestra de dades

B.2.5.2.3. Gràfics

Aquesta finestra s'utilitza per mostrar el valor de la senyal desitjada respecte el temps. Com en el cas de la finestra de Dades, si una base de dades es afegida a la nostra aplicació, les senyals es poden tractar com a valors físics.

A més de mostrar les senyals, aquesta finestra també ofereix l'opció de visualitzar el temps de resposta d'una variable d'entorn.

En la següent figura es mostra un exemple en el que els paràmetres (escala, mínim, màxim, divisió, tipus de línia, temps de mostreig, etc) són configurats prèviament. Per altra banda, les dades obtingudes per les diferents senyals poden ser exportats a un fitxer.

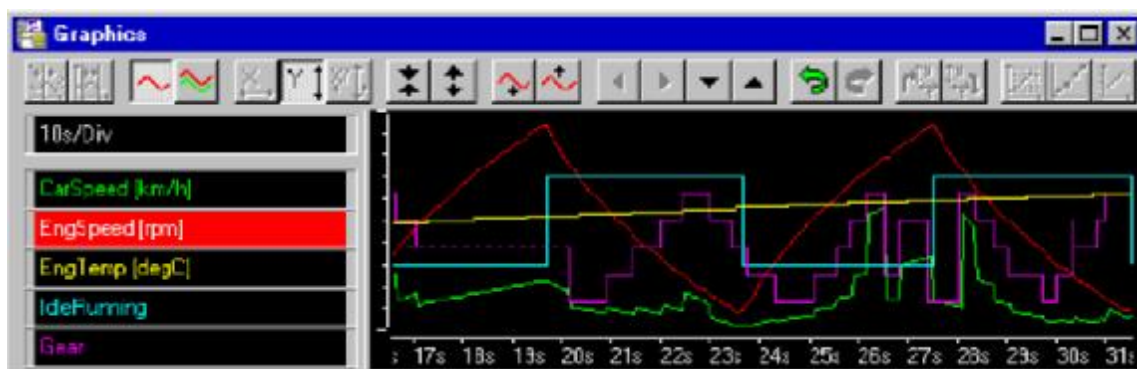


Figura B.40 Finestra de gràfics

B.2.5.2.4. Estadística

Aquesta finestra té com a funcions principals processar i mostrar el número de missatges que circulen pel bus. Una de les funcions principals es mostra el promig de temps entre els enviaments dels missatges (segs/missatge) o els missatges enviats per segon (missatges/seg.) durant la mesura. La estadística es realitza per cada identificador de missatge.

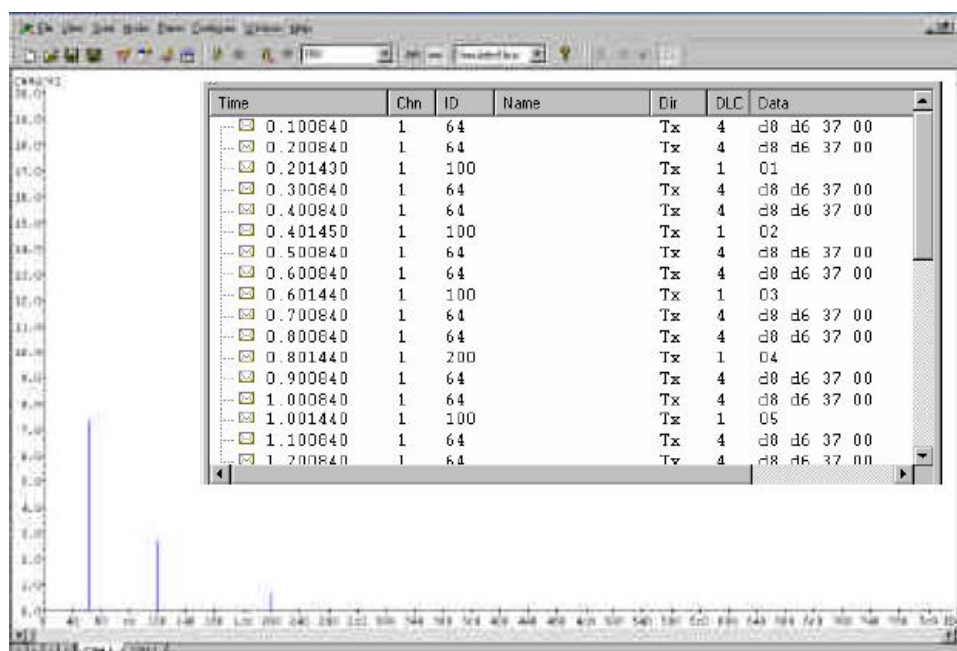


Figura B.41 Finestra d'estadística i monitorització de trames

En el menú de configuració es pot personalitzar totes les característiques de la mesura, tant a nivell visual com de processament de dades. El temps mesurat, el qual defineix el període de refresc de dades, és més precís amb un valor baix, això suposa un requeriment molt elevat dels computadors. Un temps mesurat acceptable per aplicacions estàndard és 1000ms aprox.

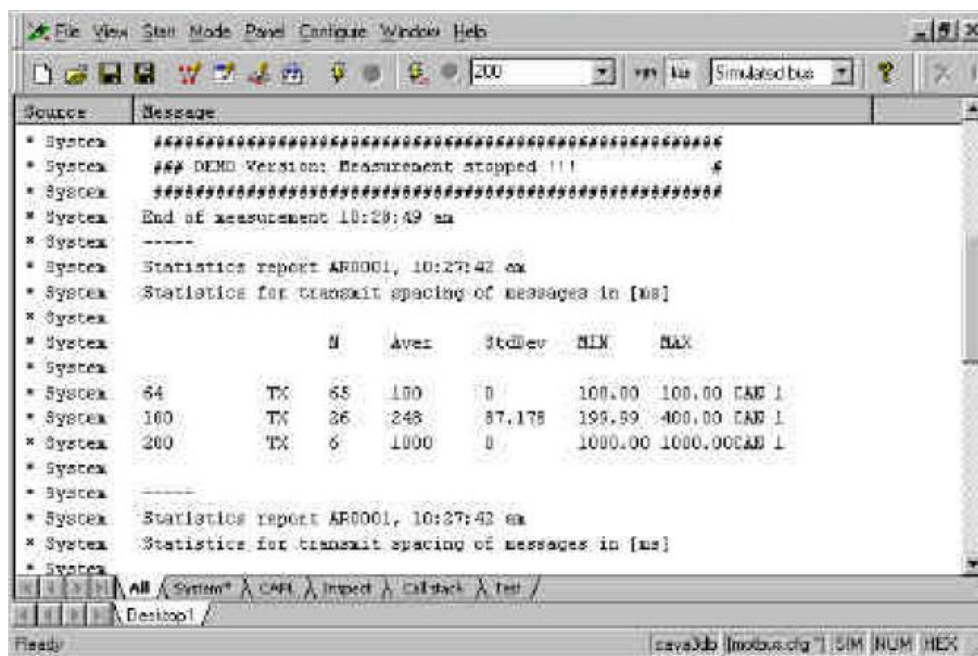


Figura B.42 Informe estadístic mostrat en la finestra d'escriptura

Per altra banda, també té la opció de realitzar un informe una vegada acabada la simulació. Aquest informe es pot guardar en un fitxer tipus text, tipus Excel o en una finestra d'escriptura. En aquest informe es mostren els diferents identificadors de missatges que s'han produït durant la mesura i per a cada un d'ells es mostra el número total de missatges, el interval mig de transmissió, la desviació estàndard i el mínim i el màxim temps transcorregut entre les transmissions.

B.2.5.2.5. Estadística del bus

EL programa CANoe adquireix dades estadístiques dels controladors de CAN implementats. Les dades adquirides depenen del hardware utilitzat. Els resultats es mostren per pantalla o es graben en un fitxer. Aquests resultats inclouen el percentatge d'utilització del bus la màxima carga produïda, el número total de dades transmeses i rebudes, les trames d'error, les trames de sobrecarrega i el estat del controlador (active, error pasive or bus off). La fiabilitat dels resultats poden variar segons l' interval de temps entre cada mostra.

	CAN 1	CAN 2
Bus load [%]	0.62	0.33
Peak load [%]	0.97	0.4
Data [frames/s]	27	20
Data [total]	1784	704
Extended [frames/s]	0	0
Extended [total]	0	0
Error [frames/s]	0	0
Error [total]	0	0
Chip state	Active	Active

Figura B.43 Finestra d'estadística del bus

B.2.5.2.6. Emmagatzemament de dades

El programa CANoe ofereix la opció de guardar el tràfic de dades del bus en un fitxer tipus ASCII o binari (genera fitxers de dimensions més reduïdes) per a que puguin ser avaluats posteriorment. En la següent figura es pot observar l'emmagatzemament de dades en un fitxer ASCII.

```

CANOE ASC - Bloc de notes
Archivo Edición Formato Ver Ayuda
Date Sat Mar 5 12:56:29 am 2005
Case hex timestamps absolute
Internal events logged
Begin TriggerBlock Sat Mar 5 12:56:29 am 2005
0.0000 Start of measurement
0.1008 1 64 Tx d 4 08 06 37 00
0.2008 1 64 Tx d 4 08 06 37 00
0.2014 1 100 Tx d 1 01
0.3008 1 64 Tx d 4 08 06 37 00
0.4008 1 64 Tx d 4 08 06 37 00
0.4014 1 100 Tx d 1 02
0.5008 1 64 Tx d 4 08 06 37 00
0.6008 1 64 Tx d 4 08 06 37 00
0.6014 1 100 Tx d 1 03
0.7008 1 64 Tx d 4 08 06 37 00
0.8008 1 64 Tx d 4 08 06 37 00
0.8014 1 200 Tx d 1 04
0.9008 1 64 Tx d 4 08 06 37 00
1.0008 1 64 Tx d 4 08 06 37 00
1.0014 1 100 Tx d 1 05
1.1008 1 64 Tx d 4 08 06 37 00
End TriggerBlock

```

Figura B.44 Emmagatzemament de dades en un fitxer ASCII

Existeixen tres modes d'emmagatzemament, segons el punt d'inici i final requerits i el temps d'emmagatzematge:

- Mode Complet: Totes les dades que circulen pel bus són emmagatzemats en un buffer i seguidament guardats en un fitxer al finalitzar la simulació.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Mode Individual: En aquest cas solament les dades que han transcorregut en els períodes de temps especificats al inici de la mesura són emmagatzemats en un fitxer.
- Mode altern: aquest mode es diferencia del anterior en la forma d'especificar els períodes de selecció de dades.

B.2.5.2.7. Filtres

El volum de dades poden ser selectivament reduït mitjançant la utilització de filtres. Per la introducció d'un filtre s'ha de situar el ratolí damunt d'alguna de les opcions comentades en els punts anteriors i polsar el botó dret del ratolí.

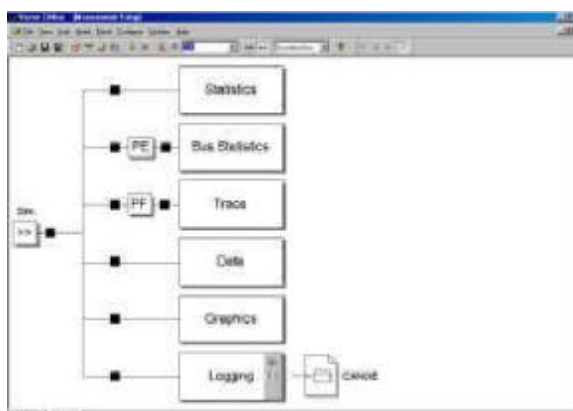


Figura B.45 Aplicació de filtres [22]

Una vegada escollits el filtre desitjat es pot accedir a configurar-lo. En la figura següent es pot observar la finestra de diàleg dels filtres tipus PF (Pass Filter).

B.2.5.3. Finestra d'escriptura (Write Window)

La informació que es genera sobre el procés i el estat de xarxa de CAN es visualitza en aquesta finestra de manera automàtica, com per exemple quan comença la mesura o possibles incidències del sistema provocades per programes creats pel usuari.

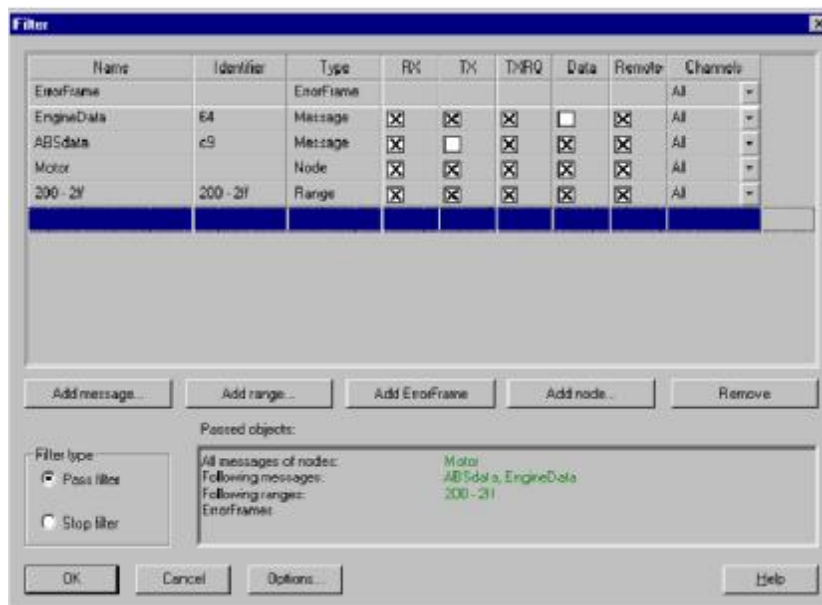


Figura B.46 Configuració d'un filtre tipus PF [22]

C. DOCUMENT DE FORMACIÓ PEL TEST BENCH DE LA SBOX

En el transcurs del projecte s'ha realitzat un document de formació del Test Bench i del conjunt de programes que el formen per als treballadors de Lear Corporation per facilitar, d'aquesta manera, la correcta manipulació per a la gent que manipularà en un futur aquest banc de proves.

C.1. INTRODUCCIÓ

La SBOX és un dispositiu que està instal·lat en els cotxes elèctrics endollables (PHEV) de la marca BMW.

La SBOX mitjançant tres relés i connectada a una bateria d'alta tensió, controla el subministrament a diferents dispositius del cotxe, com el aire condicionat, el convertidor DC/AC que controla el motor, el reductor que controla la bateria i el convertidor DC/AC que serveix per connectar-la a la xarxa elèctrica.

La SBOX tindrà una PCB de control que realitzarà les següents funcions:

- Mesura de diferents tensions i la corrent.
- Mesura del voltatge i del diagnòstic dels relés.
- Control dels relés.
- Comunicació via CAN a SME (centraleta de la SBOX).

El esquema elèctric de la SBOX el formen tres relés. Les mesures que serveixen per verificar la SBOX són aquestes quatre tensions i la intensitat.

- La U_b és la tensió d'entrada de la SBOX.
- La U_{zk} és la tensió de sortida de la SBOX.
- La U_q que és la suma entre la U_{zk} i U_{k2} .

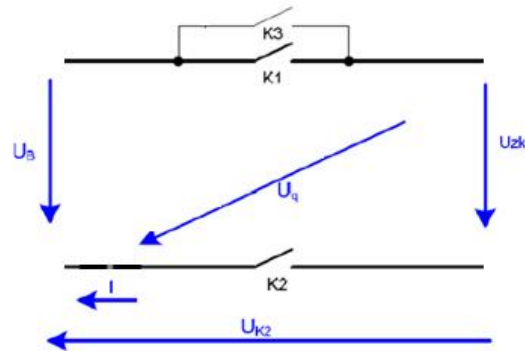


Figura C.1 Esquema elèctric SBOX

L'objectiu principal és realitzar un disseny amb diferents tests automàtics per tal de calibrar les tensions i la corrent que hi ha en la SBOX i també per comprovar que un cop calibrades, les tensions i la corrent compleixen amb els marges d'error especificats per BMW.

Per automatitzar la verificació de la SBOX es dividirà el disseny en 2 blocs. La estructura hardware on es tindrà un Test Bench amb diferents dispositius i la estructura software on s'ha dissenyat diferents programes LabVIEW per controlar la SBOX i els dispositius.

C.2. ESTRUCTURA HARDWARE

C.2.1. DIAGRAMA DE BLOCS DEL TEST BENCH

Tot el disseny està controlat mitjançant el programa LabVIEW a través d'un PC, amb l'objectiu de verificar de forma automàtica la SBOX. Amb el PC es controlarà via GPIB les fonts de tensió i d'intensitat, les fonts de mesura, la font de tensió auxiliar (DC POWER SUPPLY) i la carga que servirà per realitzar totes les proves de calibratge i mesura. Un cop acabat aquestes proves es realitzarà un report amb les mesures realitzades i es comprovarà si compleixen amb les especificacions.

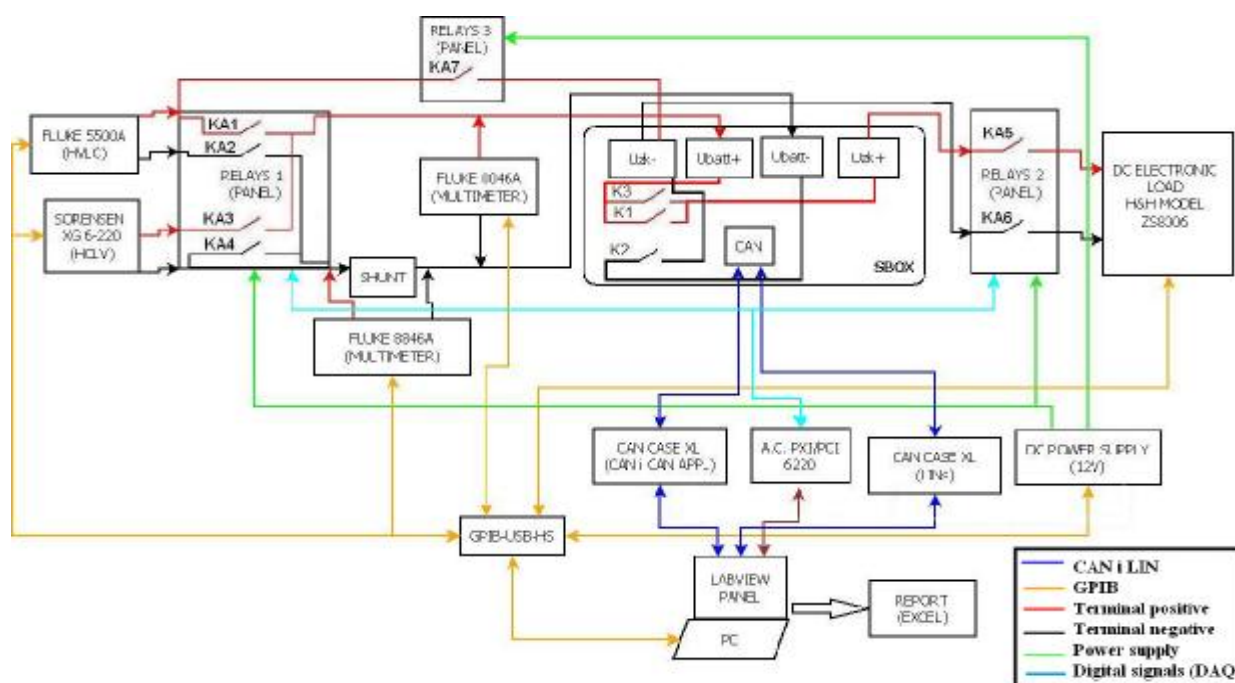


Figura C.2 Diagrama de blocs del Test Bench

En el transcurs del calibratge i de la mesures s'anirà connectant els relés (panell de relés 1, 2 i 3) de forma automàtica mitjançant una targeta d'adquisició. Depenent de la tensió i intensitat que es vulgui calibrar i depenent de la mesura que es vulgui fer en la SBOX (Uzk, Uk2, Uq i lbatt), es tancarà uns relés o uns altres. Aquestes condicions de connexió s'explicarà en el apartat de la targeta d'adquisició que és el dispositiu que controla les senyals digitals que permeten controlar els relés.

En les calibracions i mesures de tensió només es farà servir la font d'alta tensió i baixa corrent (HVLC) i per realitzar tots els calibratges i mesures de corrent, s'utilitzarà la font de baixa tensió i alta corrent (HCLV) i la carga.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

Les calibracions i les mesures de la SBOX es realitzaran a través dels busos de comunicació CAN i LIN, és a dir, mitjançant el programa CANoe i dos targetes CANCEXSL:

- 1.- La targeta CANCEXSL amb un canal de CAN, on es realitzarà totes les mesures de les tensions i de la corrent, i un canal de CAN APPL, on es realitzaran tots els procediments de calibratge en tensió.
- 2.- La targeta CANCEXSL amb els dos LINs per calibrar i mesurar la corrent.

Per altra banda, pel que fa a les mesures externes del circuit, es farà mitjançant dos multímetres on es podrà mesurar la entrada de tensió que hi ha a la SBOX i la tensió que cau en el shunt. Per últim, també es disposarà d'una DC POWER SUPPLY auxiliar que servirà per alimentar els relés de la PCB i els relés del panell.

C.2.2. ESQUEMA ELÈCTRIC DEL TEST BENCH

En la següent figura es pot observar el esquema elèctric de tot el circuit i en vermell les entrades i sortides que necessita el panell de relés per connectar-se amb els dispositius i la SBOX:

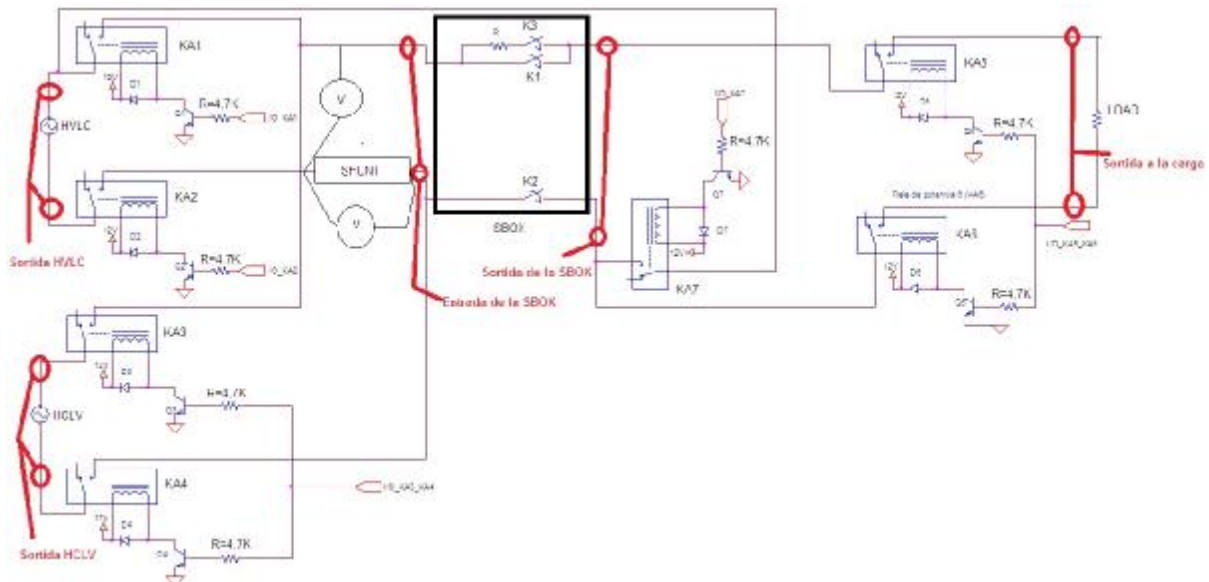


Figura C.3 Esquema elèctric del Test Bench

C.2.2.1. Panell de relés

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

El panell de relés és una capsa rectangular on només es pot veure una tapa rectangular situada a la part superior del Test Bench. Els dispositius que hi han a la tapa són els següents:

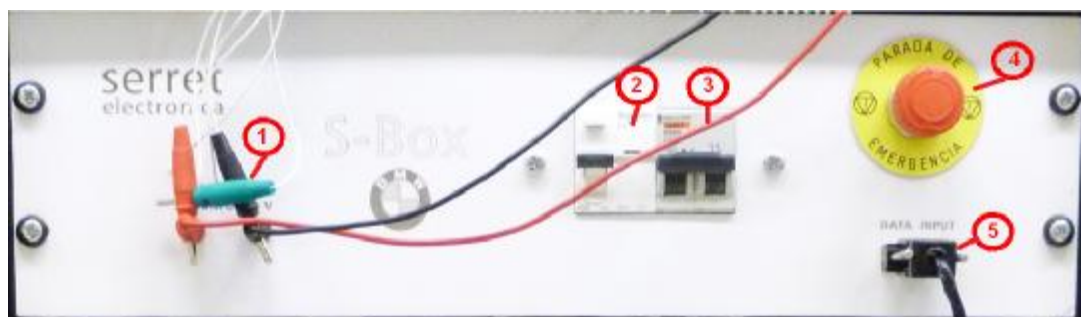


Figura C.4 Panell de relés. Vista frontal

- 1.- Els dos terminals de 12 V serviran per a l'activació dels drivers dels relés.
- 2.- El Interruptor Diferencial protegeix de possibles derives en cas que hi hagi en algun equip.
- 3.- El Interruptor Magnetotèrmic protegeix als equips en cas que hi hagi una sobretensió en la instal·lació elèctrica.
- 4.- El polsador d'emergència desconnecta de forma manual tots els relés del panell independentment de les entrades digitals.
- 5.- La entrada DB9 per on es connectaran totes les entrades digitals procedents de la targeta d'adquisició.

Les entrades i sortides pertinents per connectar els dispositius al panell de relés (evure) estaran a la part posterior del panell.

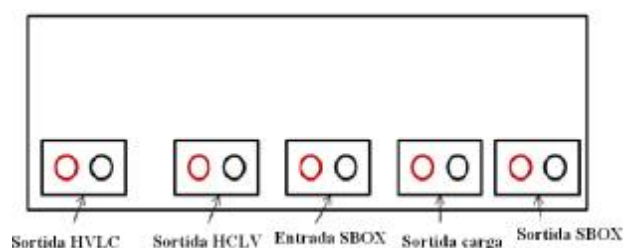


Figura C.5 Panell de relés. Vista posterior

En el interior del panell de relés hi haurà:

- 7 relés (KA1, KA2, KA3, KA4, KA5, KA6 i KA7) amb els seus drivers.

Pel que fa al tipus de relés, els KA1, KA2 i KA7 seran els mateixos relés que s'utilitzen a la SBOX (Panasonic AEV120A i 12V) i els relés KA3, KA4, KA5 i KA6 són uns altres relés que aguanten una corrent de més de 400A (Kilovac EV200). Cal dir, que cada relé estarà format per un driver (format per un transistor darlington, un díode, una resistència i una entrada digital) que servirà per controlar-lo.

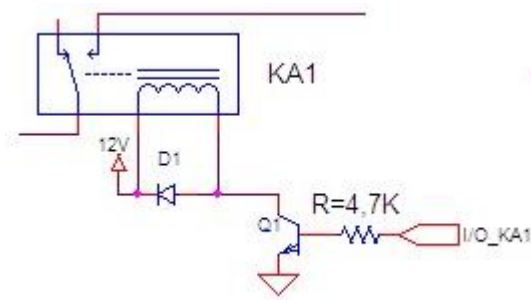


Figura C.6 Drivers dels 7 relés

- El Shunt ($60\mu\Omega$) que es farà servir per mesurar la corrent.
- 5 Senyals digitals: I/O_KA1 (activa KA1), I/O_KA2 (activa KA2), I/O_KA3_KA4 (activa KA3 i KA4) i I/O_KA5_KA6 (activa KA5 i KA6).

Per motius de seguretat, les senyals digitals que activen els relés hauran de complir unes característiques i unes condicions de connexió:

- Els relés KA1, KA2 i KA7 connectaran la font de tensió HVLC. Aniran connectats de tal manera que, a través del software, quan es tanca KA1 es tancarà automàticament KA2 i quan es tanqui KA7 també es tancarà KA2. Per altra banda, els relés KA7 i KA1 no podran estar mai connectats al mateix moment.
- Els relés KA3 i KA4 connectaran la font de corrent HCLV i aniran connectats junts, és a dir, amb una senyal digital (HCLV) de la targeta d'adquisició es controlaran els dos relés.
- Els relés KA5 i KA6 connectaran la carga i aniran connectats junts, és a dir, amb una senyal digital (LOAD) de la targeta d'adquisició es controla els dos relés.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

- Els relés KA1, KA2 i KA7 no poden estar al mateix instant de temps connectats que KA3 i KA4, ja que, la font de tensió i la font de corrent no poden estar mai connectats alhora.
- Els relés KA1, KA2 i KA7 no poden estar al mateix instant de temps connectats que KA5 i KA6, ja que, la font de tensió i la carga no poden estar mai connectats alhora.
- De les 5 entrades digitals, es controla 4 entrades digitals, ja que, KA2 depèn exclusivament de si KA1 o KA7 estan tancats.

C.2.3. INSTRUMENTS I DISPOSITIUS

En la següent figura es pot observar el Test Bench al complet. El Test Bench mesura 1,70m d'altura, 0,7m d'ample i 0,5m de profunditat. Es dividirà el banc de proves amb 9 parts.

- 1.- Tapa del Test Bench i SBOX.
- 2.- Panell de relés.
- 3.- Multímetre Fluke 8846 (National Instruments) que mesura la tensió d'entrada de la SBOX (Ubatt).
- 4.- Multímetre Fluke 8846 (National Instruments) que mesura la tensió que hi ha en el Shunt del panell de relés.
- 5.- Font de tensió HVLC Fluke 5500A (National Instruments).
- 6.- Font de corrent Sorensen XG 6-220 (Ametek) connectada en mode master.
- 7.- Font de corrent Sorensen XG 6-220 (Ametek) connectada en mode slave.
- 8.- Carga DC electronic Load H&H model ZS8006 (Höcherl & Hackl GmbH).
- 9.- Font de tensió auxiliar E3633A (Agilent), situada fora del Test Bench.

Cal dir que, les dos fonts de corrent aniran connectades en mode master-slave, és a dir, es controla mitjançant GPIB el master però les dos fonts estaran connectades en paral·lel i aplicaran la meitat de la corrent que s'introdueix per tal de poder arribar a corrents de 400A.

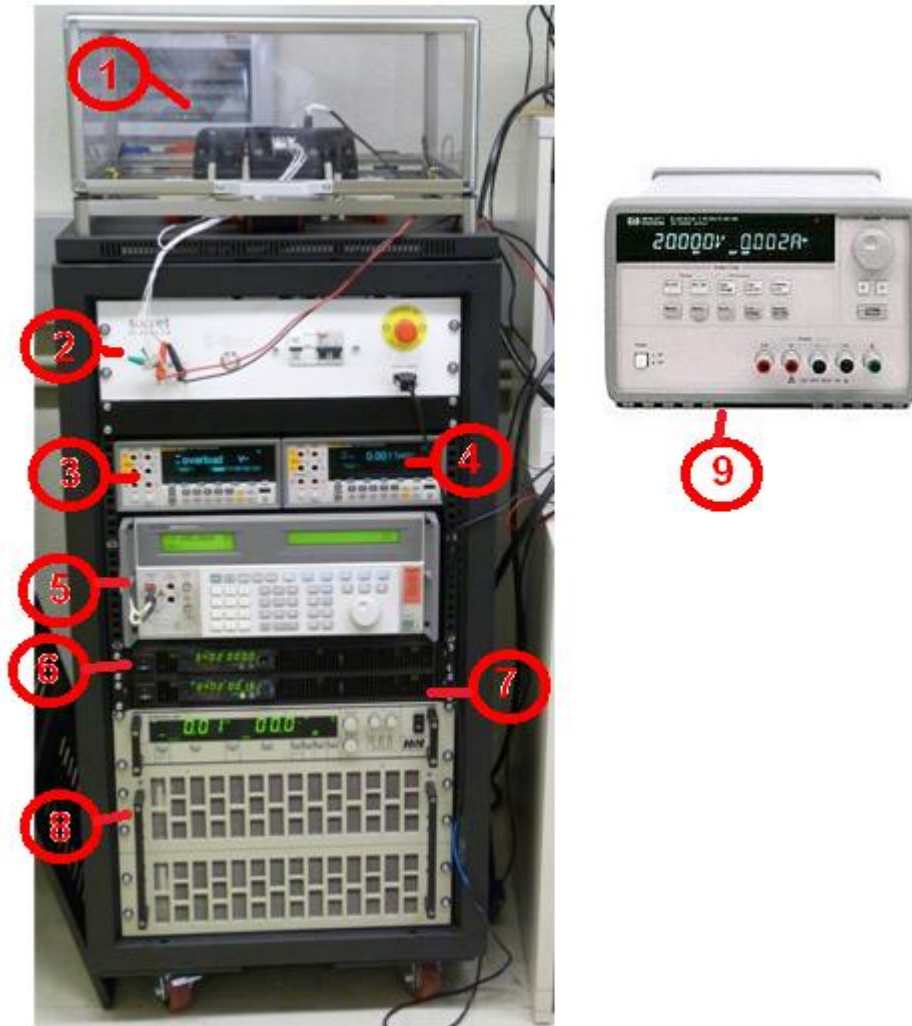


Figura C.7 Test Bench

Per altra banda, hi ha diferents dispositius i cablejat que no es veuen en l'anterior figura:

- Dos CANCASE XL (Vector), una amb dos canals de CAN i l'altra amb dos canals de LIN.
- Una targeta d'adquisició PCI/PXI 6220 de National Instruments
- Un cable GPIB-USB (National Instruments) per connectar l'ordenador amb els dispositius.
- Cables GPIB-GPIB per connectar els dispositius entre ells.



Figura C.8 Can Case XL, USB-GPIB i GPIB-GPIB

C.2.4. TAPA DE LA SBOX

La SBOX s'introdueix dins de la tapa del Test Bench, es fixa en una petita taula i es connecta amb els connectors següents per tal que es pugi connectar amb totes les parts del circuit que s'ha citat anteriorment:

- 1.- Connector Uz-.- Connector negatiu de sortida de la SBOX.
- 2.- Connector Ubatt+.- Connector positiu d'entrada de la SBOX.
- 3.- Connector de CAN. Connector CAN que uneix el dispositiu amb l'ordinador a través de les targetes CANCEXL
- 4.- Connector Sèrie serveix per comunicar-se amb el microprocessador de forma directa. Només es realitzarà aquesta comunicació per comprovar els valors verificats però no formarà part del test de validació. És un altra manera de saber quines són les mesures del microprocessador.
- 5.- Connector Ubatt-.- Connector negatiu d'entrada de la SBOX.
- 6.- Connector Uz+.- Connector positiu de sortida de la SBOX.
- 7.- Tapa de la SBOX.
- 8.- Sensor de la tapa. Detecta si està obert la tapa i obri tots els relés del panell.

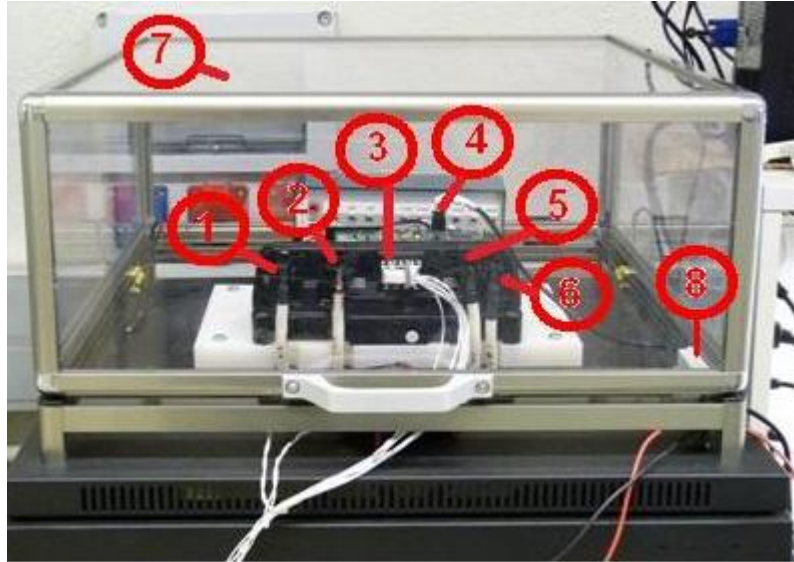


Figura C.9 Tapa del Test Bench i connexions de la SBOX

C.3. SISTEMA DE CALIBRATGE

En aquest apartat, s'explicarà quins mètodes de calibratge s'han seguit per tal de poder verificar la SBOX.

En la següent figura es pot observar com és l'estructura hardware de la placa de la SBOX. El objectiu és calibrar i mesurar les tensions i les corrents dels ADUCs.

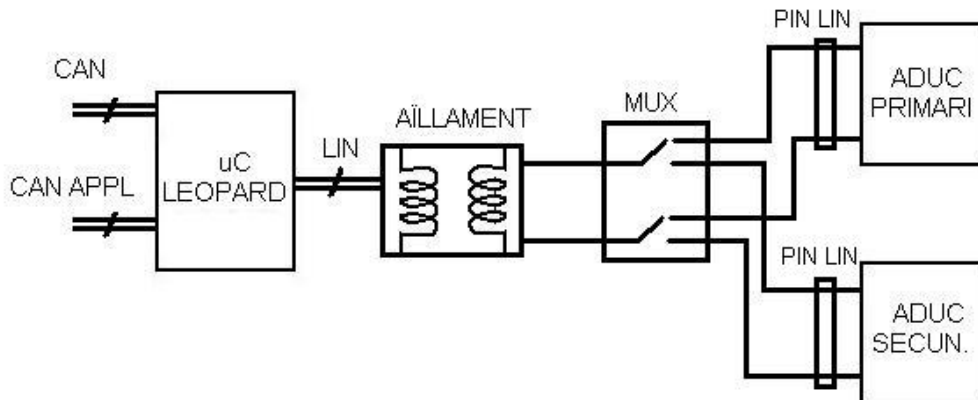


Figura C.10 Diagrama de blocs de la placa SBOX

El esquema està format per un microprocessador Leopard que es comunica amb els dos ADUCs a través d'un LIN intern. El Leopard i els ADUCs estan aïllats i mitjançant un multiplexor es selecciona un dels dos ADUCs. Es comunica amb el Leopard a través dels dos canals de CAN (CAN i CAN APPL) que serà una de les dos targetes CAN CASE, i

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

també es comunicarà, de forma directa amb els ADUCs, a través dels dos canals de LIN (PIN LIN) que serà l'altra targeta CAN CASE.

S'utilitzaran diferents sistemes per calibrar i mesurar tensió i corrent, ja que, s'utilitzaran diferents programes CANoe.

C.3.1. CALIBRATGE AMB TENSIÓ

Per calibrar amb tensió es calibra pel CAN APPL i es mesura pel CAN. El Leopard serà l'encarregat de aplicar als ADUCs els valors del guany i offset que els s'han d'aplicar, ja que, inicialment els valors estaran descalibrats.

C.3.1.1. Càlcul de guany i offsets

Per realitzar el procediment per calibrar les tensions, s'utilitzarà diferents sistemes per a càlculs de funcions lineals i els passos seran els següents:

- 1.- Aplicar les dos tensions (130V i 435V) que seran els dos punts per fer la recta de referència imaginària.
- 2.- Mesurar els valors reals en aquests dos punts que seran els dos punts per fer la recta de mesura imaginària.
- 3.- Calcular mitjançant formules matemàtiques (funcions lineals) els valors de guanys i els offsets que s'han d'aplicar perquè els valors reals s'aproximin als valors de referència (tot mitjançant LabVIEW).
- 4.- Aplicar el guany i el offset al Leopard.

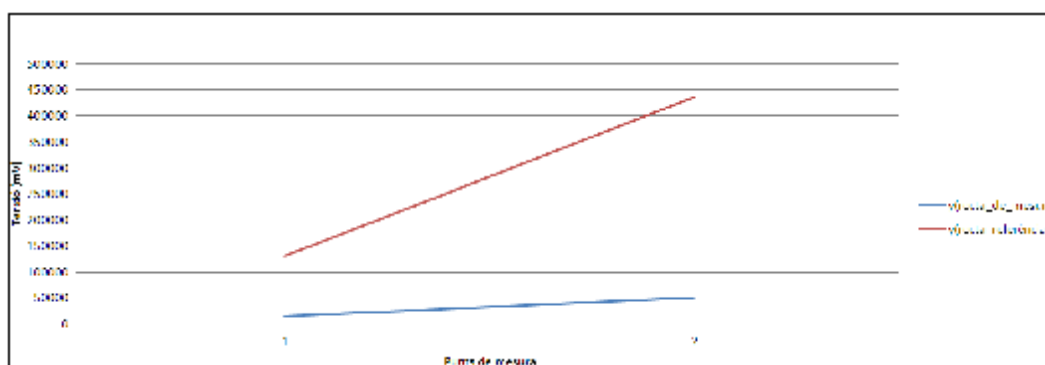


Figura C.11 Gràfic de calibració

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

En l'anterior figura es pot observar en color vermell la recta de referència i en color blau la recta de les mesures de les tensions que es llegeix pel CAN. Es pot observar com a la recta de color blau se l'hi ha d'aplicar un guany (augmentar la pendent) i un offset positiu (pujar la recta) perquè els valors siguin iguals que la recta de color vermell.

C.3.1.2. Lectura de valors

Al software del CANoe hi haurà dos sistemes per mesurar els valors reals depenent de la tensió (Ubatt, Uzk, Uq i Uk2) a calibrar:

- Pel canal 2 (CAN). Les lectures es faran de forma directa, a través del CAN (Ubatt i Uzk).
- Pel canal 1 (CAN APPL). Un cop s'obté els valors reals (Uq i Uk2) de cada tensió dins del marge d'error especificat, s'envia un ordre de lectura (request) i després es llegeix el valor de senyal que correspon a la tensió real .

Un cop calibrada la recta, es realitza un seguit de lectures de tensió als valors especificats a les especificacions i serà les que s'enviaran al report, on es comprovarà si compleixen amb el marge d'error, i per tant, la SBOX ha estat ben calibrada.

C.3.2. CALIBRATGE AMB CORRENT

Per calibrar amb corrent es connecta directament amb els ADUCs (sense que intervingui el Leopard) amb els pins de LIN. S'obrirà un altre programa CANoe per aquesta calibració.

El procediment de calibratge serà el següent:

- 1.- Aplicar corrent. Realitzar uns punts de mesura ja establerts que seran a 0A, 60A i 200A.
- 2.- Llegir corrent del LIN (per cada valor) fins que la lectura compleixi amb els errors, ja que, en aquest cas no cal aplicar guanys ni offsets perquè és el mateix ADUC que fa els càlculs per tal de donar la corrent que es vol aplicar.

Un cop calibrats els tres punts, es realitza un seguit de lectures de corrent als valors especificats a les especificacions i serà les que s'enviarà al report, on es comprovarà si compleixen amb el marge d'error, i per tant, la SBOX ha estat ben calibrada.

C.4. ESTRUCTURA SOFTWARE

El operari que manipuli el Test Bench només tindrà un panell LabVIEW on, abans de començar el programa, introduirà el numero de sèrie i el numero de prototip de la SBOX. Per altra banda, es podrà observar en forma de leds, en quina part del programa es troba la validació de la peça.

A partir del software que es rep del programa CANoe que és desde on es realitza totes les operacions (tancar relés i llegir valors de tensió i corrent), s'ha introduït un petit codi en aquest software que farà d'interfície LabVIEW-CANoe per tal de controlar el programa CANoe desde un altre programa LabVIEW.

S'ha realitzat una interfície LabVIEW-Excel per tal de controlar el programa Excel desde LabVIEW i d'aquesta manera poder introduir els valors mesurats en un report Excel.

C.4.1. PROGRAMA PRINCIPAL

La estructura principal del programa LabVIEW és la següent:

- Primer que tot, es configura tots els dispositius que funcionen mitjançant GPIB, és a dir, es comprova que estan correctament connectats i que funcionen correctament.
- Connectar el programa CANoe i comprovar si les trames de CAN i LIN són les correctes.
- Activar els relés de la SBOX i comprovar que funcionen.
- Selecció de la SBOX. Depenén de quina versió sigui (Baukasten i v.444) es realitza un tipus de calibratge, mesura i report diferent.
- Realitzar el programa de consum de la PCB de la SBOX sense cap relé tancat.
- Realitzar el calibratges i mesures en tensió i corrent de la versió especificada.
- Realitzar un REPORT amb un arxiu Excel.

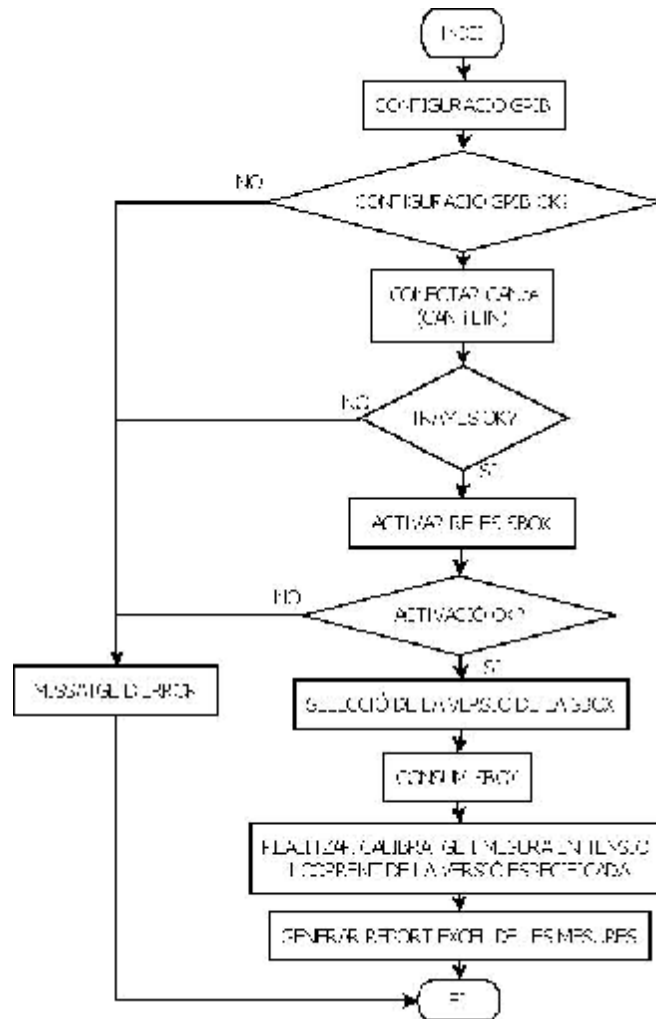


Figura C.12 Programa principal

C.4.1.1. Selecció de la versió de la SBOX

En el Test Bench es podrà verificar dos tipus de SBOX, la Baukasten i la v.444. En aquest programa, un cop introduïda la SBOX dins la tapa, s'introdueix el Part Number que indicarà el tipus de SBOX i la seva versió i el sample number que és el numero de prototip de la placa.

Com que en cada tipus de SBOX es farà diferents proves, es realitzarà dos subprogrames diferents depenén del tipus de SBOX que s'hagi de verificar. Un altra de les diferències entre la versió 444 i la Baukasten és que en la Baukasten es controla els relés de la SBOX mitjançant CAN i amb la versió 444 es fa amb la targeta d'adquisició.

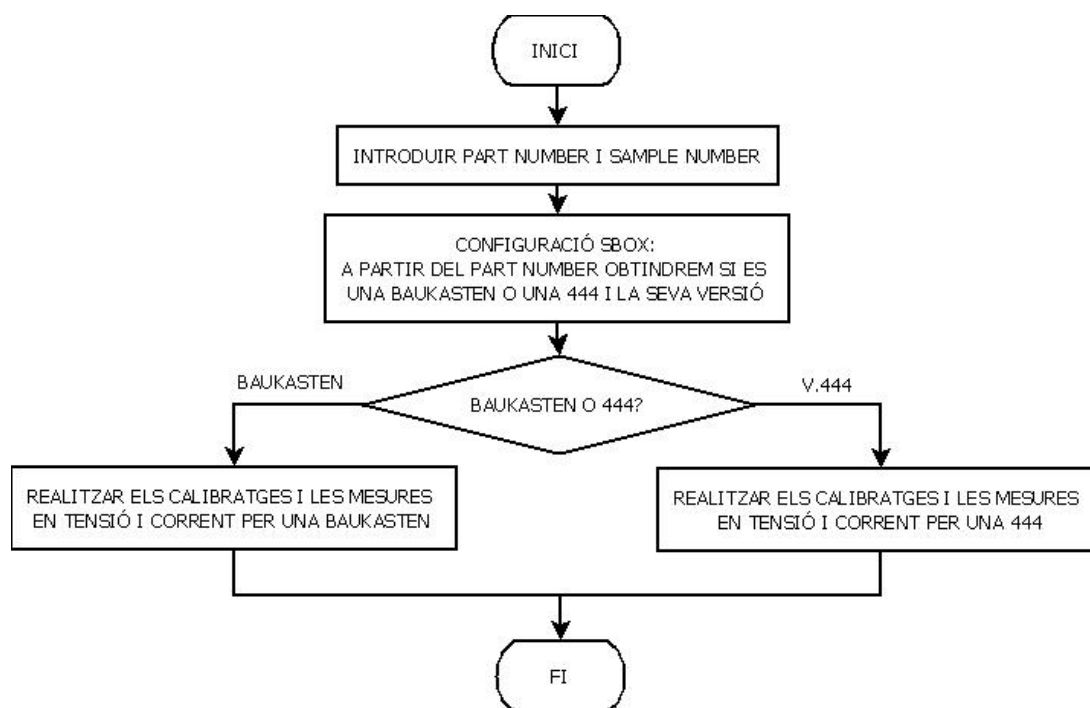


Figura C.13 Programa principal de calibratge i mesura

Un cop es sap quin tipus de SBOX és, es segueix els següents passos:

- 1.- Programa de consum de la SBOX. En aquest programa, s'introdueix diferents valors de tensió amb la font auxiliar de tensió i es mesura el consum de la placa de la SBOX. Aquestes proves es realitzaran amb els relés de la SBOX i els relés del panell obert, ja que, es vol calcular el consum de la placa.
- 2.- Calibrar, mesurar i reportar les tensions.
- 3.- Calibrar, mesurar i reportar la corrent.

C.4.1.2. Programa calibrar en tensió

En la configuració en tensió s'haurà de tancar i/o obrir els relés de tal manera que es pugui fer les calibracions i les mesures desitjades.

Els esquemes que s'utilitzaran seran els següents:

- L'esquema del circuit per calibrar i mesurar U_{batt} i U_q és el següent:

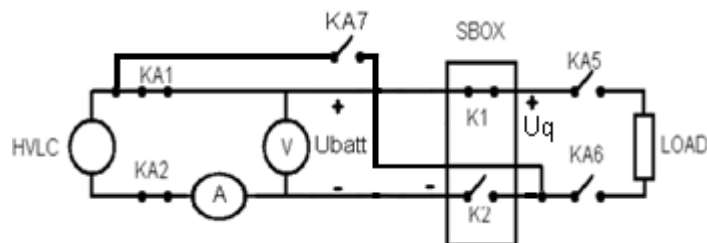


Figura C.14 Configuració estructura hardware per mesurar U_{batt} i U_q

- L'esquema per calibrar i mesurar U_{k2} és:

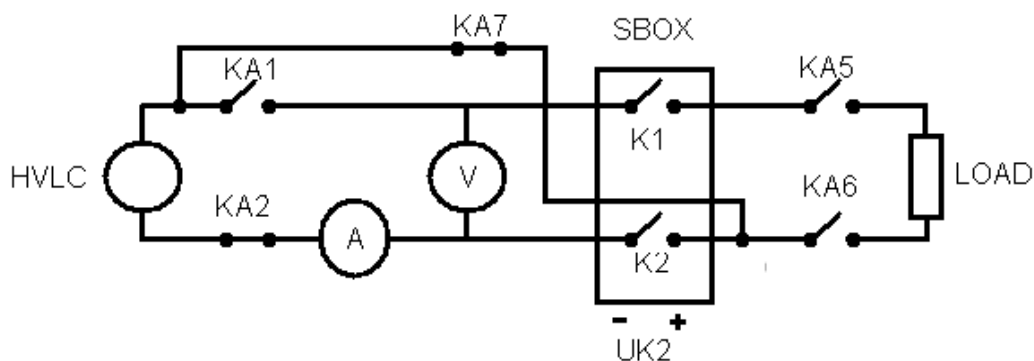


Figura C.15 Configuració estructura hardware per mesurar U_{k2}

- L'esquema per mesurar U_{zk} és:

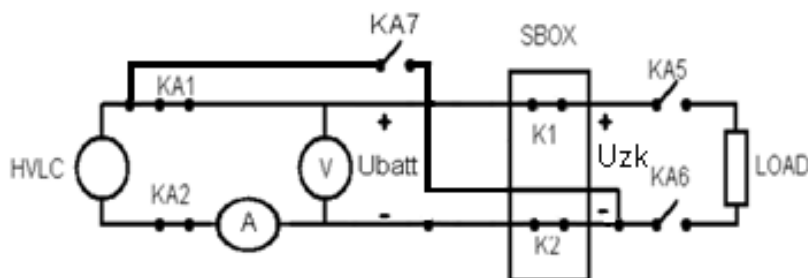


Figura C.16 Configuració estructura hardware per mesurar U_{zk}

C.4.1.3.2. Calibrar corrent en una v.444

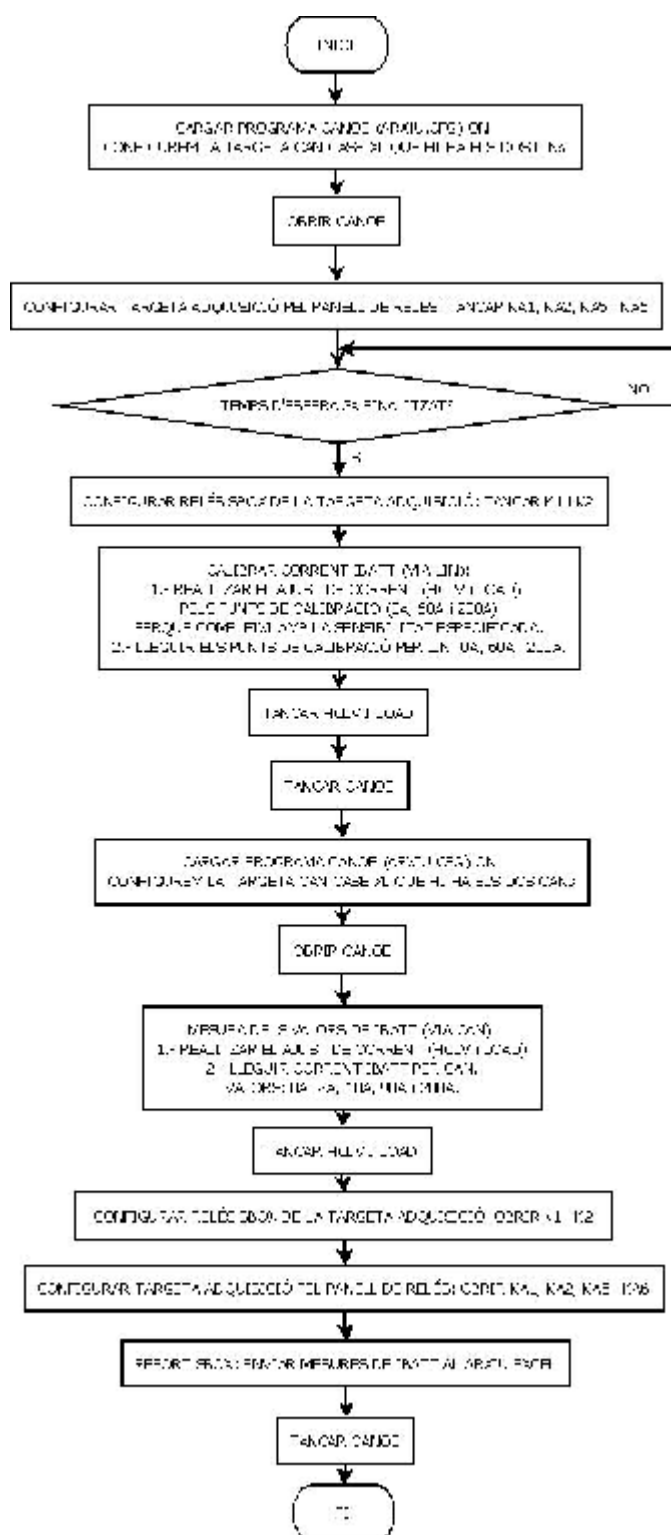


Figura C.21 Diagrama de flux del programa calibració de corrent de la v.444

C.4.1.4. Report

Per realitzar el Report s'obri el arxíu que es farà servir de motlle (un per cada tipus de SBOX), després es realitza uns bucles que estan escrivint en el report nou fins que totes les dades ja s'hagin introduït en el Excel, i un cop estiguin totes les dades introduïdes, es guarda l'arxíu on el nom serà SBOX més el part number de la peça. En el Excel cada cèl·lula que l'hi s'introdueix el valor estarà programada per indicar si compleix amb les especificacions.

Hi haurà dos reports, un pels tipus Baukasten i un altre per la v.444, ja que, les mesures que s'han de reportar són diferents.

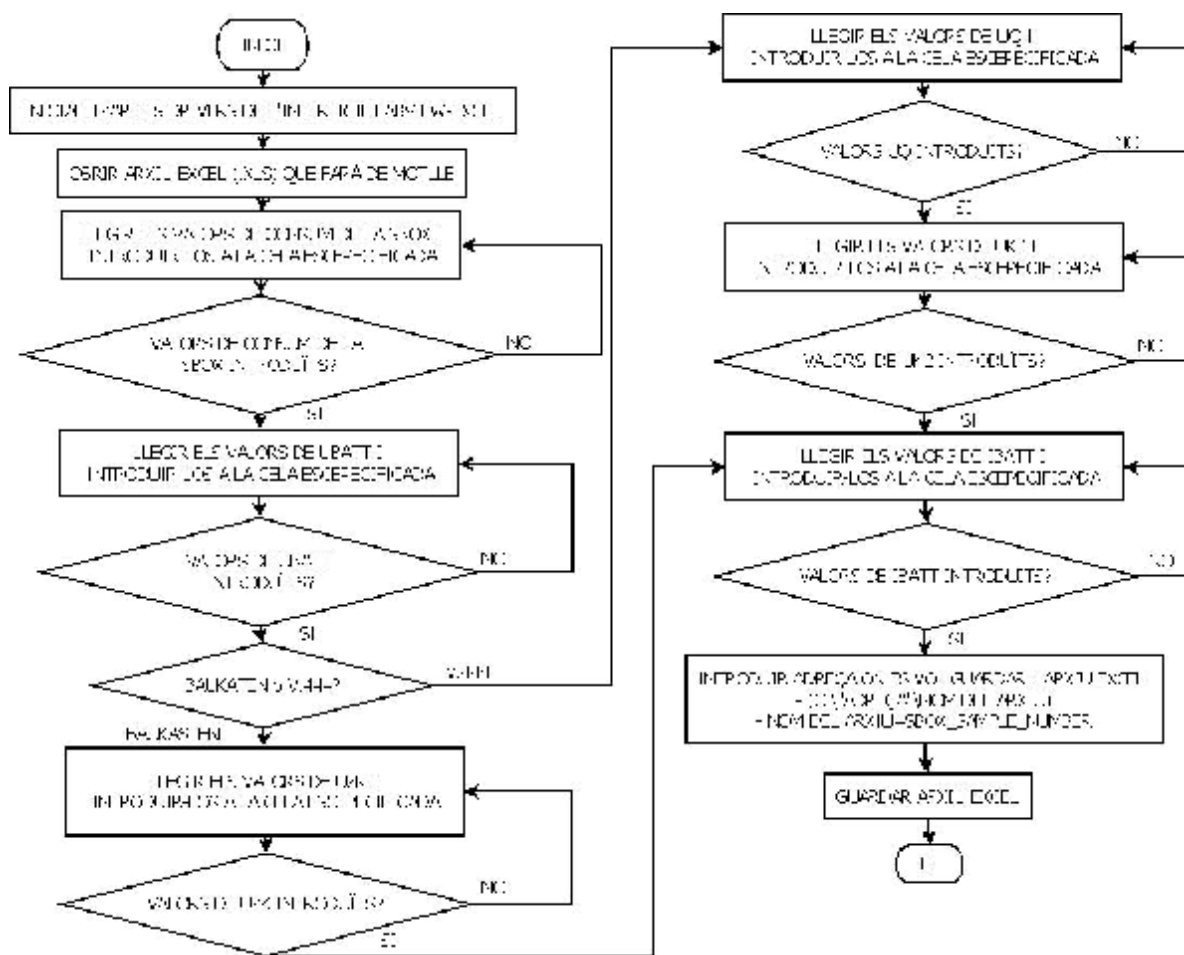


Figura C.22 Diagrama de flux del programa Report

C.4.1.5. SEGURETAT

En aquest Test Bench existeixen varis sistemes de protecció que protegeixen a les persones que estan treballant amb la SBOX i als dispositius:

- Botó d'emergència: Serveix per obrir de forma automàtica els relés del panell.
- Interruptor Diferencial: Protegeix al circuit (estructura del sistema) de possibles derives.
- Interruptor Magnetotèrmic: Protegeix els equips de possibles sobretensions.
- Tapa: La tapa aïlla la SBOX de l'exterior i impedeix que es pugui tocar qualsevol connexió de dins la placa quan aquesta està en funcionament. Aquesta tapa és molt important, ja que, en les proves es treballa amb altes tensions i corrents.
- El sensor de la tapa. Hi ha un sensor en la tapa que quan s'obri la tapa s'obrin de forma automàtica tots els relés del panell eliminant qualsevol possible descarrega.
- Tot el Test Bench anirà dins d'un armari tancat d'acer, aïllant d'aquesta manera, totes les connexions entre els dispositius de l'usuari.

D. PROJECTE VERDE

D.1. INTRODUCCIÓ

El Projecte VERDE és un projecte CENIT (Consortis Estratègics Nacionals en Investigació Tècnica) que forma part del Programa Ingenio 2010 que és un projecte del Govern espanyol del Ministeri de Ciència i Innovació i gestionat pel Centre de Desenvolupament Tecnològic Industrial per incrementar la inversió I+D, tant pública com a privada. Aquest projecte està dedicat a la investigació i generació del coneixement necessari per a la futura fabricació i comercialització de vehicles ecològics a Espanya, bàsicament híbrids endollables (PHEV) i elèctrics (EV).

El Projecte VERDE té per finalitat investigar i generar coneixement en els temes clau necessaris per a la fabricació i comercialització de vehicles ecològics a Espanya, el que permetria reduir la dependència energètica del petroli del nostre país, reduir les emissions de CO₂ en el sector del transport, afavorir la penetració de les energies renovables i garantir el futur del sector industrial i de l'I + D de l'automoció a Espanya. Està subvencionat pel Ministeri de Ciència i Innovació dins del Programa CENIT (Consortis Estratègics Nacionals d'Investigació Tècnica) a través del Centre per al Desenvolupament Tecnològic Industrial (CDTI).

El Projecte VERDE està liderat pel Centre Tècnic de SEAT i coordinat administrativa, tècnica i científicament per CTM Centre Tecnològic, on participen 16 empreses del sector d'automoció, infraestructura i energia, com ara, AIA, Cegasa, Circutor, Cobra, Endesa, Ficosa , Green Power, Iberdrola, Infranor, Lear, Mapro, Red Eléctrica Espanya, Siemens, Rovalma i Tècniques Reunides.

A més participen 13 centres entre Universitats i Organismes Públics de Recerca, com AICIA, ASCAMM, CIDETEC, CIRCE, CNM del CSIC, CTM Centre Tecnològic, IIC, IIT de la Universitat Pontifícia de Comillas, IREC, Leitat, Tecnalia, Universitat Politècnica de Catalunya (UPC) i Universitat Carlos III. Es desenvolupa en un total de 6 comunitats autònomes: Andalusia Aragó, Catalunya, Madrid, País Basc i C. Valenciana i el seu pressupost total és de 34M €. [23]

D.2. PRÀCTIQUES SOBRE EL PROJECTE VERDE

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

Al laboratori de Lear Corporation hi ha un motor elèctric on es treballa en aquest projecte i es controlat mitjançant el programa CANoe, és a dir, el PC i una placa on hi ha una DSP es comuniquen mitjançant el bus CAN (utilitzant la targeta CANcase) per controlar el motor.

Uns membres del equip Lear han realitzat un programa on mitjançant un panell (executable) pots realitzar diferents ordres al motor, com per exemple, aplicar-li la velocitat, llegir tensions, entre multitud d'altres funcions.

Les pràctiques amb aquest projecte tenen com a objectiu realitzar un estudi per saber com funciona el programa CANoe a través d'un programa real i saber com modificar i ampliar el programa, per tal de poder escriure i llegir mitjançant LabVIEW a través de CANoe, és a dir, l'objectiu és practicar amb la interfície CANoe/LabVIEW.

Cal especificar que el projecte no es centrarà amb el programa CANoe que han fet els companys (el programa del projecte VERDE està en el apartat del annex) sinó en la configuració de la lectura i escriptura de la interfície CANoe/LabVIEW.

El programa LabVIEW s'ha reconfigurat a partir d'una estructura prototip de un programa anterior de Lear.

Es dividirà aquest apartat amb els canvis i ampliacions que es facin en els dos programes (LabVIEW i CANoe) en la lectura i l'escriptura d'algunes variables.

D.2.1. ESCRIPTURA

En aquest cas s'haurà de modificar una petita part del programa CANoe del VEDRE i crear un programa que pugi escriure a través de LabVIEW una senyal de control que fa referència a l'activació d'un convertidor (CTRL_EN_DCDC). S'ha escollit aquesta senyal perquè es pot observar de manera molt senzilla que funciona, ja que, si es mira el consum de la placa, quan s'activa la senyal augmentarà i si es desactiva tornarà al seu valor de consumició normal.

D.2.1.1. CANoe

La simulació del programa tindrà la següent forma. Hi haurà la ECU que fa referència a la placa que es treballa amb el projecte Verde on hi ha una DSP i la ECU desde on es controla el Panell, és a dir, desde on es realitzen tots els tests. Aquests dos ECUs estaran comunicades mitjançant el bus CAN.

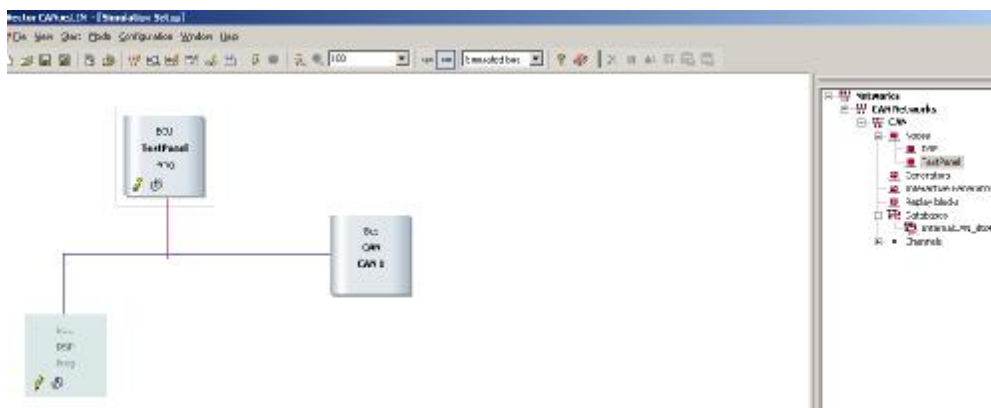


Figura D.1 Esquema de la simulació del programa

En el programa CANoe s'haurà de realitzar diferents modificacions en el CAPL i en el DataBase.

D.2.1.1.1. CAPL

L'única de les dos ECUs que tindrà un arxiu .can serà la ECU del Test Panel, ja que, la DSP ja està programada, i ella envia i rep els missatges de manera autònoma.

La forma i les parts importants del arxiu .can del programa VERDE seran les següents:

En el apartat de System (en start) s'inicialitzarà les diferents funcions i els dos timers (indicant en quin període en ms comença el timer).

Les funcions que s'han creat al programa són:

- DisableAll(): Inicialitzar variables. Es realitzarà de la següent manera:

```
putValue(env_ctrl_en_dcdc,0);
```

- SendConfig(): Envia configuració de variables de control a través de senyals. Es realitzarà la configuració de cada variable de la següent manera.

```
msg1.CTRL_EN_DCDC = getValue(env_ctrl_en_dcdc);
```

- SendMsg2(): Envia diferents tipus de senyals de control. Es realitzarà la configuració de cada variable igual que en l'apartat anterior.

Els timers serveixen per realitzar les diferents configuracions i condicions que són particulars del programa dels companys de Lear Corporation i l'objectiu serà on s'ha d'aplicar la nostra funció que s'ha creat.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

En els CAN Messages hi han totes les senyals que rebra el panell. Un exemple de com es fa la definició de cada missatge és:

```
on message INVERTER_REPORT_1 {

    putValue(env_disp_3v3, (float)this.DISPLAY_3V3_REG / 1000);

    putValue(env_disp_v_batt, (float)this.DISPLAY_V_BATT / 10);

    putValue(env_disp_v_np, (float)this.DISPLAY_V_NP / 10);

    putValue(env_disp_curr_Urms,(float)this.DISPLAY_CURR_U_RMS / 10);

}
```

Les environments són les variables de control que serveixen per controlar el panell. En aquest apartat estan totes les variables de control que fan referència al panell.

Els canvis que es realitzà en aquest arxiu seran només per la escriptura de variables a traves de LabVIEW, ja que, la lectura de variables es pot realitzar sense modificar cap dels arxius CANoe. Els canvis són els següents:

Es crearà una quarta funció anomenada Set_Signal(long,float) que servirà per l'escriptura de senyals a través de LabVIEW. Aquesta funció serà un case en el que ha partir de dos variables (long i float) es pot triar quina variable es vol modificar. En aquest cas només s'activarà la senyal que fa referència a l'activació del convertidor DC/DC.

```
void Set_Signal (long channel, double p1){

    switch(channel){

        case 1:

            putValue (env_ctrl_en_dc dc, p1); //Posar el valor de la variable p1 a
            la //variable de control env_ctrl_en_dc dc

            break; } }
```

En el apartat de System (en start) s'inicialitzarà la funció anterior. Sempre amb aquest ordre:

```
on start {
```



```

    DisableAll(); // inicialització

    Set_Signal(channel,p1); //escriptura

    SendConfig(); //enviament

    SendMsg2(); //enviament

}

```

Es definirà les dos variables de la funció en el apartat de variables:

```

long channel;

float p1;

```

En el cas de que es volgués realitzar un programa molt més complex no només s'hauria de cridar la funció d'escriptura ha l'inici sinó que també s'hauria de cridar-la dins dels dos timers (sempre abans de cridar les funcions SendConfig() i SendMsg2()), per tal de que complís amb les condicions de les variables però com que per el practica només es vol saber com s'escriu i es llegeix les variables a través de Labview, es podrà dir que per a l'objectiu d'aquestes pràctiques és suficient.

D.2.1.1.2. DataBase

En el arxiu només es modificarà les senyals que es volen escriure a través de LabVIEW, ja que, les senyals que es llegeixen no cal modificar-les.

Com s'ha dit anteriorment, en aquesta pràctica només es vol escriure a través de la senyal del convertidor DC/DC (CTRL_EN_DCDC) i, per tant, serà l'única modificació que es realitzarà.

La modificació de la senyal es realitzarà seguint els següents passos:

- Prémer amb el botó dret damunt de la senyal a modificar i entrar en la opció d'editar senyal.
- Canviar la taula de valors (en l'apartat de definició de la senyal) per la de VtSig_CTRL_EN_DCDC.

En l'apartat de descripció de valors (value description), assignar els diferents valors que pot tenir la senyal. En aquest cas només ha de tenir el valor d'activar (0x01, ON) i de desactivar (0x00, OFF).

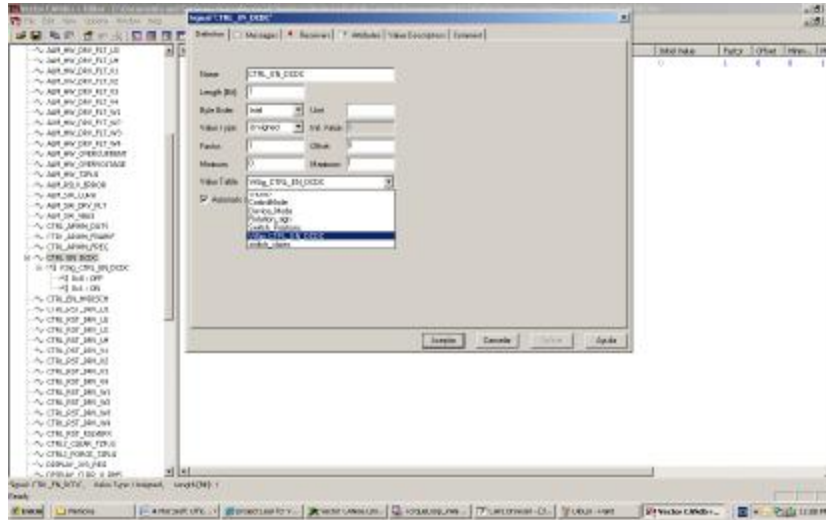


Figura D.2 Modificacions del DataBase

D.2.1.2. LabVIEW

El diagrama de flux per l'escriptura és el següent:

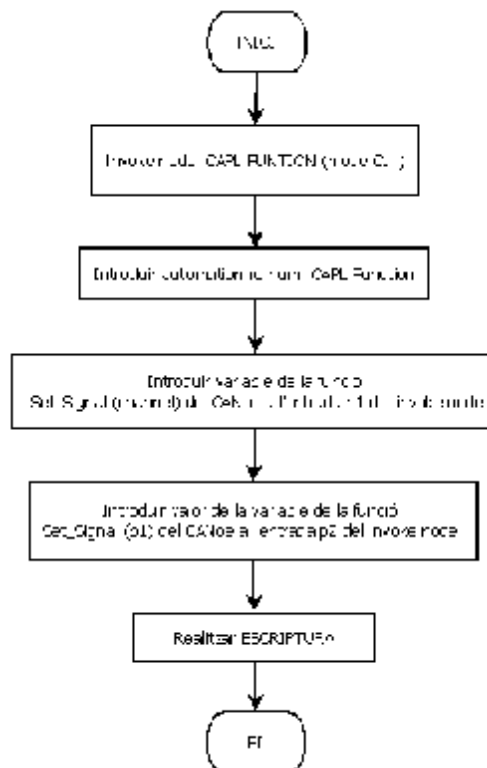


Figura D.3 Diagrama de flux d'escriptura

D.2.2. LECTURA

En aquest apartat es realitzarà la lectura de dos senyals aprofitant la missatgeria de CAN que ja està feta. Aquestes senyals seran:

- Senyal DISPLAY_3v3_REG del missatge INVERTER_REPORT_1 que fa referència a la tensió de la PCB de la placa.
- Senyal DISPLAY_V_BATT del missatge INVERTER_REPORT_1 que fa referència a la tensió de bateria .

D.2.2.1. CANoe

En la lectura no caldrà realitzar cap tipus de modificació al programa CANoe, ja que, al programa LabVIEW que s'ha dissenyat, llegeix les variables que s'envien la DSP de la placa VERDE i el Test Panel sense cap modificació de codi del CANoe.

D.2.2.1.1. CAPL

L'únic que s'ha de fer en el CAPL del panell, és fixar-se en com estan escrites les senyals i els missatges a través de CAN Messages i posar-les al LabVIEW de forma exacta.

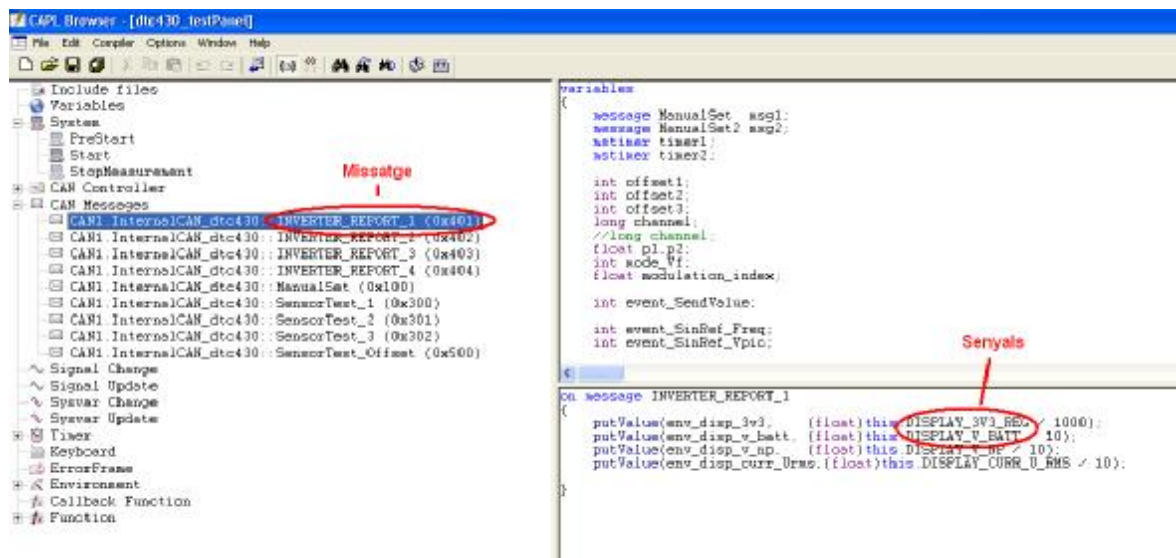


Figura D.4 Missatgeria en el CAPL

D.2.2.2. LabVIEW

El diagrama de flux de la lectura és el següent

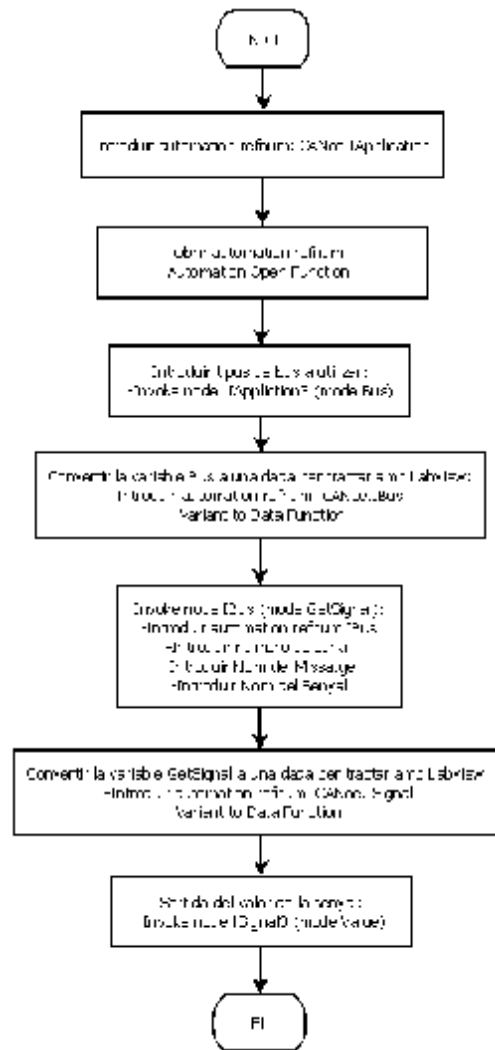


Figura D.5 Diagrama de flux de la lectura de senyals

D.2.3. PROGRAMA PRINCIPAL PROJECTE VERDE

Es realitza el programa principal amb una subVI i una VI de variables globals:

- El programa principal segueix 4 passos:

- 1.- Encendre CANoe.
- 2.- Crida de la subVI Set_Signal que serveix per escriure a la interfície LabVIEW/CANoe.

Desenvolupament d'un entorn de test automàtic per a components de vehicles híbrids

3.- Escriure i llegir les senyals del CANoe.

4.- Apagar el CANoe.

- Variables globals: Per tractar els diferents automation refnoms (Application, CAPL i CAPL Function), ja que, si s'introdueix com a variables locals, el programa no funcionarà.

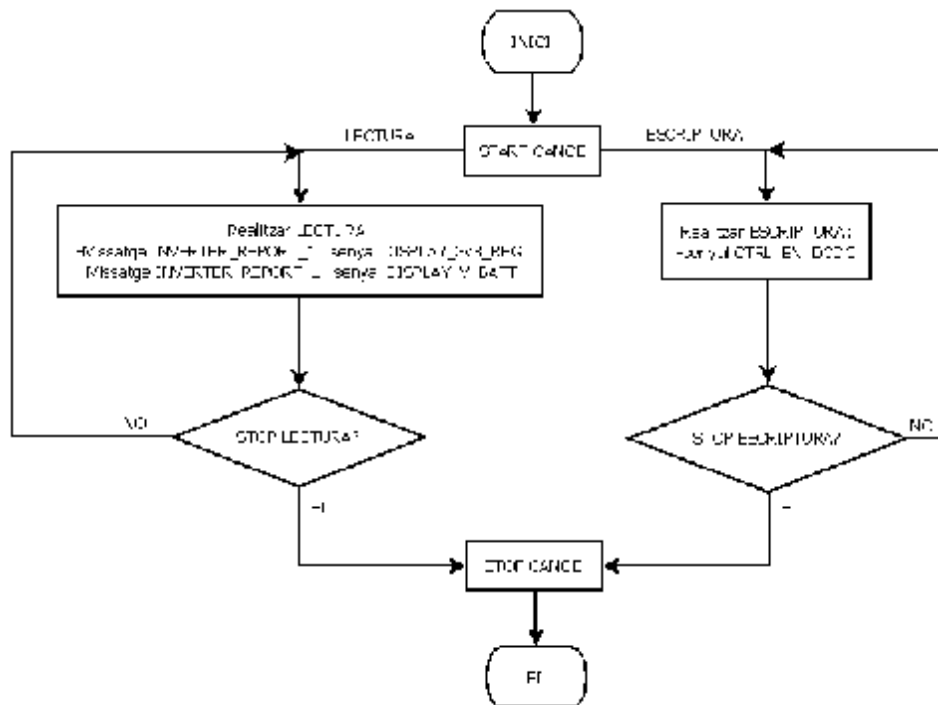


Figura D.6 Diagrama de flux del programa principal