

**Continental** 



# ARAMIS II FEA TESTING

REPORTE FINAL DE RESIDENCIA PROFESIONAL

Elaborado por: Oscar de Jesús Espinosa Trujillo  
Alumno del Instituto Tecnológico de Tuxtla Gutiérrez.  
Área: Ingeniería Electrónica. No. De Control: 06270043  
Asesor Interno: M.C. Rafael Sánchez Maldonado.  
Compañía: Continental Guadalajara Services, S.A. De C.V.  
Asesor Externo: Ing. Edgar Madrigal Pérez

**JUNIO/2010**

## Resumen

Los Airbags son dispositivos de seguridad pasiva utilizados para evitar lesiones, fracturas e incluso la muerte cuando se suscita un accidente automovilístico. Por esta razón el diseño de estos dispositivos debe ser muy riguroso de tal forma que permita garantizar al usuario su correcto funcionamiento, si el sistema de airbags de un automóvil falla puede ocasionar problemas graves. En Continental Guadalajara Services S.A. de C.V., se diseña y producen sistemas electrónicos de control de bolsas de aire y se esta consciente del grado de seguridad el cual se debe contar; por eso, en todos los diseños electrónicos de seguridad pasiva se desarrollan proyectos que generen y realicen pruebas de funcionalidad para garantizar a los clientes la eficacia de todos los productos.

ARAMIS 2 FEA Testing es uno de esos proyectos el cual ha sido generado para probar la funcionalidad del diseño de un sistema de control de airbags denominado ARAMIS 2. Es un diseño que fue requerido por la empresa Mercedes Benz y que fue diseñado bajo sus especificaciones. Los errores encontrados en estas pruebas funcionales fueron analizados y clasificados para posteriormente realizar los ajustes necesarios en dicho sistema. A lo largo de este trabajo se explicará de manera detallada las actividades realizadas, los problemas que se presentaron y la manera en que se resolvieron.

## Índice

1. A cerca de la empresa	5
1.1 Datos Generales	5
1.2 Visión de Continental	5
1.3 Continental en el mundo	5
1.4 Divisiones y Business Units	6
1.5 Business Unit Chassis and Safety	6
2. Introducción	8
2.1 Justificación del proyecto	8
2.2 Objetivos generales y específicos	9
2.3 Alcances y limitaciones	9
3. Marco Teórico	10
3.1 Como funcionan los airbags	10
3.2 Leyes del movimiento	11
3.3 Partes de un sistema de airbags	11
3.4 Como usar un airbag	12
3.5 Descripción del proyecto ARAMIS II	13
3.5.1 Connector	25
3.5.2 Headrest	26
3.5.3 Analogue Crashout	26
3.5.4 CAN	27
3.5.5 Battery Feed	28
3.5.6 Ignition Feed	29
3.5.7 EVZ Power Supply	29
3.5.8 PVZ Power Supply	30
3.5.9 Vsat, Vperi, Vcore	31
3.5.10 VCC5 power Supply	31
3.5.11 MUC	32
3.5.12 SUC	33
3.5.13 Digital Crashout	33
3.5.14 EEPROM	34
3.5.15 XY Sensor	35
3.5.16 Firing ASIC	35
3.5.17 PPS Cap. Test	36
3.5.18 Resistance Test	36
3.5.19 Leakage Test	37
3.5.20 PUC	37
3.5.21 PVZ Discharging	38
3.5.22 Line comunication PUC-MUC	39

4. FEA Testing	40
4.1 Ejecución de las fallas de cortos y abiertos	40
4.2 Descripción de herramientas de diagnóstico	40
4.3 Procedimiento para realizar las pruebas	48
4.4 Elaboración del reporte del resultado de las pruebas	48
4.5 Detección, análisis y corrección de fallas de unidades ARAMIS II	54
4.6 Programación de microcontroladores de unidad ARAMIS II	55
4.6.1 Diagrama de Flujo de programación	55
4.6.2 Hardware Utilizado	56
4.6.3 Programación del APACE	57
4.6.4 Programación del PUC	57
4.6.5 Programación del SUC	60
4.6.6 Programación del MUC	61
4.6.6.1 Programación de MUC-ROM	61
4.6.6.2 Programación de MUC-FBL	63
4.6.6.3 Programación de MUC-Application	64
5. Conclusiones	65
6. Bibliografía	66

## 1. A CERCA DE LA EMPRESA

### 1.1 Datos Generales

Razón Social: Continental Guadalajara Services S.A. de C.V.

Domicilio: Camino a la tijera No. 3 C.P. 45640, Tlajomulco de Zúñiga Jalisco, México.

Giro: Electrónico-Automotriz

### 1.2 Visión de Continental

- Hacer tu movilidad más segura, cómoda y sustentable
  - Gracias a la competencia, productos y servicios, nosotros, junto con nuestros clientes, proveemos seguridad, confort y un manejo divertido. Nuestros productos, tecnologías y procesos significan para nosotros una contribución para hacer la movilidad más sustentable.
- El desarrollo es nuestra pasión
  - Marcamos la pauta con nuestras innovaciones, alto desarrollo de productos, calidad, compromiso y excelente servicio para nuestros clientes. Esto nos motiva en todas nuestras áreas de negocios. Combinando la tecnología, ecología y aspectos económicos personales para desarrollar y hacer la diferencia.

### 1.3 Continental en el mundo

Existen cerca de 190 plantas de producción y desarrollo en 25 países. La matriz se encuentra en Hanover, Alemania. (Véase Fig. 4.1)



*Fig. 1 Lugares donde Continental se encuentra (Áreas en gris oscuro)*

## 1.4 Divisiones y Business Units

Existen 3 principales divisiones en Continental, Safety, Environment y Information.

- Safety
  - Esta división tiene como objetivo innovar los sistemas de seguridad de un automóvil y sistemas que proveen confortabilidad. Esta división esta conformada por 4 business units: Chassis & Safety, Interior, Tires, Contitech.
- Environment
  - Desarrolla sistemas sustentables con el fin de hacer automóviles que no dañen al medio ambiente. Está conformado por las siguientes Business Units: Powertrain, Interior, Tires, Contitech.
- Information
  - Realiza innovaciones que permitan facilitar el manejo de la información dentro del vehículo mediante sistemas muy sofisticados haciendo de la movilidad inteligente. Esta conformada por la business unit llamada Interior.

## 1.5 Business Unit Chassis & Safety

Esta división esta compuesta por los siguientes departamentos:

- Electronic Brake Systems
  - Esta división se encarga de desarrollar sistemas de control hidráulico electrónico, como frenos ABS, etc.
- Hydraulic Brake Systems
  - En esta división se trabaja con sistemas de frenado hidráulico. Por ejemplo innovan sistemas de calibración, discos, frenos de tambor, reguladores de frenos de presión, etc.
- Sensorics
  - Desarrollan distintos tipos de sensores relacionados con la electrónica automotriz. Por ejemplo se trabaja con sensores de velocidad, ángulo, torque, posición, velocidad de transmisión, etc.
- Passive Safety & ADAS
  - En esta división se trabaja en el desarrollo de sistemas de bolsas de aire principalmente y también con sistemas de asistencia de manejo mediante radares y cámaras para asistir al conductor a la hora de cambiarse de carril, estacionarse, ir en reversa, etc.
- Chassis Components
  - Como su nombre lo dice, en esta división se realizan trabajos concernientes al chasis del automóvil. Se trabaja por ejemplo con Chasis Electrónicos, sistemas de suspensión, etc.

El proyecto ARAMIS II FEA Testing fue desarrollado dentro del departamento llamado Passive Safety & ADAS conocido por sus siglas como PSAD. Bajo la supervisión de el ingeniero de Hardware **Edgar Madrigal Pérez** quien se desempeño como asesor externo de este trabajo.

## 2. INTRODUCCION

Los sistemas de control de airbags son sistemas electrónicos que realizan el procesamiento de señales provenientes de varios sensores con el fin de determinar la condición de accidente en un automóvil. Cuando esto sucede y mediante señales eléctricas se activan actuadores los cuales permiten que una reacción química infle las bolsas de aire con nitrógeno en un tiempo no mayor a los 30 o 40 milisegundos.

Continental cuenta con un proyecto de este tipo llamado ARAMIS II el cual es una unidad de control de airbags que se ha diseñado para Mercedes Benz. ARAMIS II FEA Testing, es el proyecto que deriva de ARAMIS II y que tiene como finalidad realizar pruebas de hardware para asegurar su correcto funcionamiento. Estas pruebas se realizan a un módulo montado en una tarjeta de prueba llamada FEA (Failure Effect Analysis), la cual contiene el mismo circuito del ARAMIS II pero en un tamaño mayor ya que incluye puntos de prueba, sockets, jumpers, etc., que permiten realizar mediciones, abrir conexiones, realizar cortos, entre otras cosas.

De forma general el proyecto consiste en poner a prueba la unidad realizándole fallas adrede en el hardware con el fin de examinar el impacto que estas provocan en el funcionamiento general del sistema. Los errores encontrados se clasifican por su severidad, ocurrencia y detectabilidad para que posteriormente sean analizados por los diseñadores del ARAMIS II y de esta manera determinar si es necesario hacer cambios de diseño con el fin de evitar los errores más severos, con mucha probabilidad de ocurrencia y con baja posibilidad de ser detectados a tiempo.

Se cuenta con un equipo computarizado el cual es capaz de registrar las fallas que suceden en la unidad y mediante un software son visualizadas en una computadora, este es el método usado para detectar las fallas.

### 2.1 Justificación del Proyecto

La **principal razón por la que el proyecto es justificable** es la de poder ofrecer al cliente la garantía de la eficacia del producto. Si estas pruebas no se realizan, Continental no puede probar la certeza del buen funcionamiento del producto y resulta sumamente importante que sus productos sean 100% seguros porque de esto depende su reputación, su posición en el mercado internacional y sobre todo la seguridad de los usuarios.

Otra buena razón se debe a los requerimientos del cliente, ya que Mercedes Benz solicita a Continental un producto diseñado con estrictas normas de calidad y requerimientos demasiados exigentes, esto obliga a realizar procesos de diseño y fabricación muy especiales.



## 2.2 Objetivos Generales y Específicos

El objetivo general de este proyecto es cumplir con la realización de todas las pruebas posibles para este diseño, apegándose a todas las normas de calidad y cumpliendo todos los puntos especificados en los requerimientos del cliente de tal manera que el producto sea un producto 100% confiable y seguro.

Los Objetivos específicos son:

- Realizar actividades que resulten en el transcurso del trabajo y que permitan cumplir de manera consistente con el objetivo general de este proyecto.
- Realizar la documentación del trabajo para reportar los resultados obtenidos
- Solucionar los problemas que se presenten y actividades no planeadas, entre otros.

## 2.3 Alcances y limitaciones

Este proyecto esta basado en un diseño que es propiedad de Continental, y se pretende que debido al buen trabajo que se realice y el éxito general del producto, la planta Continental en Guadalajara pueda obtener más licitaciones para la fabricación de este y otro tipo de productos.

Sin embargo, el éxito no solo depende de la buena ejecución del ARAMIS II FEA Testing, ya que hay muchos otros proyectos basados en este mismo producto realizándole pruebas físicas, eléctricas, de software, etc. También el grupo de diseño juega un papel importante ya que este es el encargado de elaborar el primer diseño y realizar las correcciones necesarias que pudieran surgir de las observaciones y resultados de los proyectos que someten a pruebas el sistema ARAMIS II. Como se puede notar en lo dicho anteriormente, ARAMIS II FEA Testing no tiene como objetivo modificar o tratar de corregir los errores encontrados en el sistema, sino solamente realizar las observaciones necesarias mientras que el grupo de diseño se encarga de tomar las acciones preventivas para corregir los problemas encontrados.

### 3. MARCO TEÓRICO

Los airbags son un tipo de dispositivos de seguridad automotriz como los cinturones de seguridad los cuales están contruidos con cojines inflables colocados en puntos estratégicos dentro del automóvil como el volante, tablero de instrumentos, puertas, techo, asientos, etc., y son manipulados mediante un sistema de control el cual cuenta con sensores de choque y una computadora que permiten inflar rápidamente las bolsas para ofrecer protección a los ocupantes a la hora del impacto.

#### 3.1 Como funcionan los Airbags

Por años, el confiable Cinturón de Seguridad había sido el único método pasivo de seguridad para los coches. Había debates a cerca de la seguridad de estos, especialmente en relación a los niños, pero con el tiempo, muchos países implementaron leyes en cuanto a los cinturones de seguridad. Las estadísticas nos han mostrado que el uso de cinturones de seguridad ha salvado miles de vidas que pudieron haber muerto en choques.

Al igual que los cinturones de seguridad, el concepto de airbags, fue diseñado para brindar este mismo tipo de protección, conocido como seguridad pasiva. Las primeras patentes se remontan a unos dispositivos diseñados para aviones las cuales fueron usados en la segunda guerra mundial. En los años 80, apareció el primer diseño comercial de airbags para automóviles.

Desde e1998 todos los carros nuevos en los Estados Unidos incluyeron airbags para el chofer y copiloto como una requisición. A la fecha las estadísticas muestran que los airbags han reducido el riesgo de muerte en un choque frontal en un 30 por ciento. Actualmente existen coches que cuentan con 24 airbags en diferentes puntos estratégicos. Al igual que los cinturones de seguridad ahora los airbags han venido a formar parte en las leyes de muchos países, donde se han adoptado como una medida de seguridad importante y que exigen que las industrias que los fabrican tengan un proceso más exigente de fabricación y pruebas.

En este trabajo, se pretende mostrar un poco de la tecnología que Continental Corporation, emplea para fabricar airbags, así también aprender sobre la forma en que funcionan y los problemas que estás tecnologías han experimentado.

### 3.2 Leyes del Movimiento

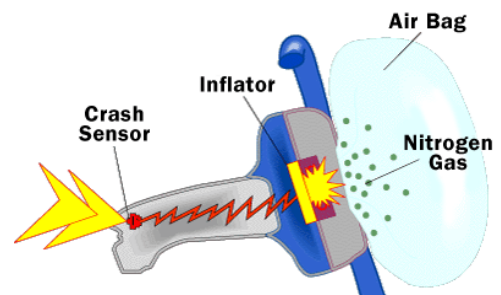
Antes de especificar detalles, vamos a echar un vistazo a algunas leyes físicas que rigen el movimiento. Sabemos que cuando los objetos se encuentran en movimiento estos tienen un impulso (el producto de la masa y la velocidad del objeto). A menos que una fuerza externa actúe sobre el objeto este continuará moviéndose en cierta velocidad y dirección. Si consideramos a un automóvil como un objeto incluyendo a sus ocupantes y si ambos se encuentran desplazándose a cierta velocidad, a menos que exista una fuerza externa que se oponga a que sigan avanzando estos seguirán su camino a cierta velocidad. Sin embargo si el automóvil se detiene repentinamente los ocupantes del automóvil seguirán viajando a la velocidad del automóvil por unos instantes más después de que el coche se haya detenido.

La fuerza del impulso que los objetos tienen a la hora del choque sigue presente por un período de tiempo después del choque. Cuando un automóvil se impacta, la fuerza requerida para detener el objeto es muy grande por que el impulso del carro cambia instantáneamente mientras la de los pasajeros no (esto no por mucho tiempo). Un sistema de amortiguamiento ayuda a detener al pasajero evitando posibles daños físicos. Esto es precisamente lo que buscan los airbags, bajar la velocidad del pasajero sin provocar daños sin embargo existen muchas limitaciones para lograr este objetivo. El espacio entre el chofer y el volante o el tablero es muy reducido por lo tanto el tiempo para actuar es muy pequeño, sin embargo esa pequeña cantidad de espacio y tiempo es valiosa.

### 3.3 Partes de un sistema de airbags

El objetivo de los airbags es bajar la velocidad de los ocupantes de la forma más amortiguada posible en fracciones de segundo. Existen 3 partes principales que permiten lograr esto:

- **La bolsa de aire:** Fabricada de nylon, la cual es colocada dentro del volante o tablero principalmente, aunque existen bolsas para las ventanas, techo, etc.
- **El control electrónico:** este es el dispositivo que activa el inflado de las bolsas. Esta activación sucede cuando se detecta un impacto.
- **El sistema de inflado:** es un sistema que usa ácido sódico ( $\text{NaN}_3$ ) con nitrato de potasio ( $\text{KNO}_3$ ) para producir gas nitrógeno. Una explosión de nitrógeno permite inflar las bolsas en cuestión de milisegundos. (Véase Fig. 2)



*Fig. 2 Partes que componen un sistema de airbags*

Los primeros intentos para hacer un sistema funcional para inflar las bolsas resultaron demasiado caros y con muchos obstáculos técnicos relacionados con el almacenamiento y liberación del gas comprimido. Los investigadores se preguntaban:

- Si había un espacio suficiente en el coche para el tanque del gas.
- Si el gas debía ser almacenado a alta presión.
- Como expandir rápidamente las bolsas en distintas temperaturas y sin emitir un estallido ensordecedor.

Ellos necesitaban encontrar una reacción química que cumpliera con las características anteriores y fue hasta los años 70's que esto se consiguió.

El sistema de inflado que actualmente se usa provoca una explosión extremadamente rápida que permite obtener un gran volumen de gas para inflar las bolsas. La velocidad promedio a la que una bolsa puede llegar a inflarse es de 322 Km/h. Un segundo después de esto el gas se empieza a liberar a través de unos pequeños orificios de la bolsa para permitir el movimiento de los pasajeros. Todo esto sucede en aproximadamente un cuarto de segundo, tiempo suficiente para evitar graves consecuencias.

### **3.4 Como usar un airbag**

Cuando estamos usando un vehículo con sistema de airbags es completamente necesario usar también el cinturón de seguridad ya que la eficacia de dicho sistema depende de la distancia que se guarde entre el volante o tablero y el cuerpo del pasajero, esta distancia debe ser de aproximadamente de 25 cm. Los investigadores han determinado que la zona de riesgo cuando un airbag se empieza a inflar son los primeros 8 cm., del volante al conductor ya que al haber una explosión demasiado rápida la velocidad con la que se dispara la bolsa podría provocar daños.

Para los niños las medidas son un poco diferentes, ya que un airbag puede causar severos daños o incluso matar a un niño que está sentado demasiado cerca del tablero. Los expertos agregan que para los niños es necesario tomar las siguientes precauciones:

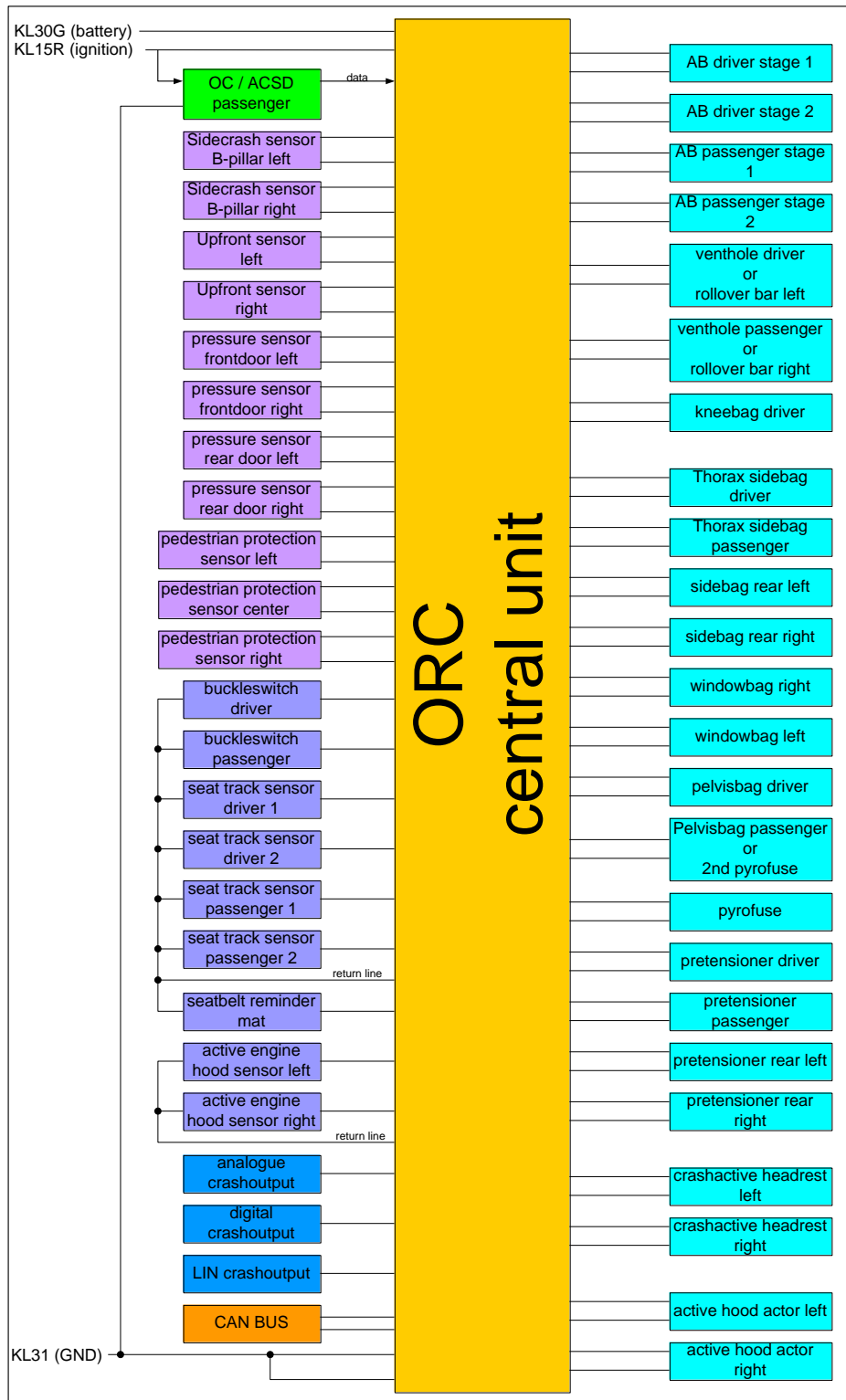
- Para los niños pequeños, deberán ser colocados en un asiento especial en la parte trasera del coche (Nunca adelante).
- Los niños menores de 12 años deberán usar el cinturón de seguridad de los asientos traseros, si el coche no cuenta con estos cinturones es conveniente instalarlos.
- En casos especiales habrá necesidad de desactivar los airbags colocados en un lugar reservado para una persona determinada, más adelante veremos cómo desactivar los airbags.

La National Highway Traffic Safety Administration (NHTSA) de los Estados Unidos en 1997 reformó una ley la cual permite a las manufactureras desarrollar airbags de alta y baja potencia es decir la ley permite en algunos casos que los airbags detonen a un 20 a 35 por ciento, esto con el fin de brindar protección a niños y personas con capacidades diferentes. Además en 1998 se comenzaron a agregar switches para activar o desactivar ciertas bolsas que no desean ser usadas, esto con una previa autorización de la NHTSA para propietarios de vehículos quienes consideren se localicen en uno de los siguientes grupos de riesgo:

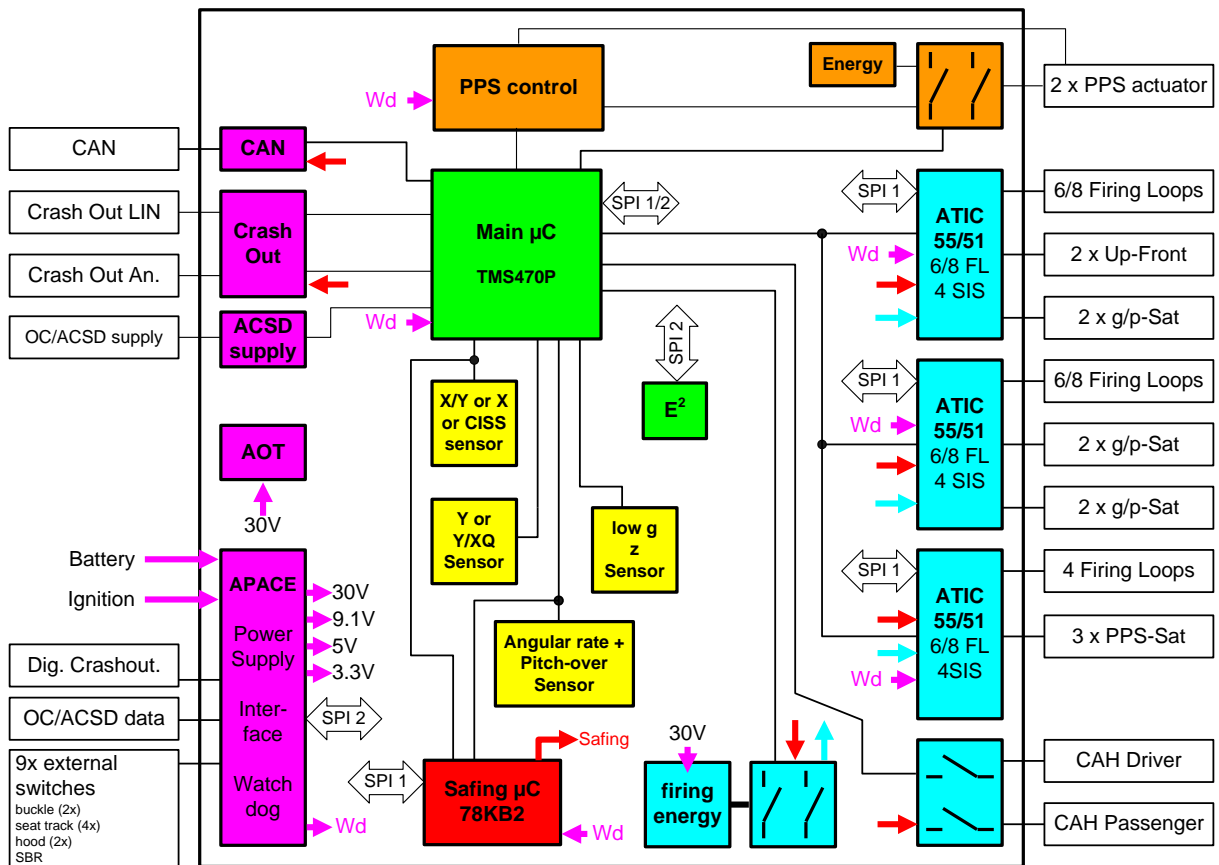
- Individuos con condiciones médicas las cuales no le permitan recibir impactos como los de una detonación de un airbag
- Para personas que por sus condiciones médicas o físicas no puedan manejar a una distancia mínima del volante de 25 cm.
- A personas que necesiten mantener una mínima distancia a su niño, debido al constante monitoreo médico al que está sujeto el niño.

### **3.6 Descripción del proyecto ARAMIS II**

En esta sección se ofrece una descripción general del proyecto ARAMIS II sobre el cual se estuvo trabajando. Se presentan los diagramas eléctricos a si como una pequeña descripción sobre los sub bloques que podemos encontrar en dichos diagramas. Cabe mencionar que los detalles que se mencionarán no están relacionados con la lógica de funcionamiento en particular si no con la tarea que realiza el circuito de forma general y de las señales que utiliza y la relación que tienen cada señal con los demás circuitos.



**ESQUEMA GENERAL DEL AIRBAG ARAMIS II**



*Diagrama a bloques la unidad de control del airbag ARAMIS 2*

En la Figuras anteriores se muestra el diagrama a bloques del airbag ARAMIS 2. Se pueden observar todas las partes que lo componen.

A continuación se hace una descripción breve de cada uno de los bloques que se muestran:

**OC / ACSD Passenger (Occupant Classification / Automatic Child seat detection Passenger):** Esta es una característica del airbag ARAMIS 2 la cual permite determinar el tamaño aproximado del pasajero. Esto la hace a través de un tapete interno en los asientos del coche que detecta el peso.

**Side Crash Sensor B pillar left:** Sensor colocado en el pilar izquierdo del coche para detectar el choque lateral.

**Side crash sensor B pillar right:** sensor colocado en el pilar derecho del coche para detectar el choque lateral.

**Upfront sensor left:** sensor colocado en la parte superior izquierda en el frente del coche para detectar choques.

**Upfront sensor right:** sensor colocado en la parte superior derecha en el frente del coche para detectar choques.

**Pressure sensor frontdoor left:** sensor de presión colocado en la puerta delantera izquierda.

**Pressure sensor frontdoor right:** sensor de presión colocado en la puerta delantera derecha.

**Pressure sensor reardoor left:** sensor de presión colocado en la puerta trasera izquierda.

**Pressure sensor reardoor right:** sensor de presión colocado en la puerta trasera derecha.

**Pedestrian protection sensor left:** sensor de choque con peatón lado izquierdo.

**Pedestrian protection sensor center:** sensor de choque con peatón lado central.

**Pedestrian protection sensor right:** sensor de choque con peatón lado derecho.

**Buckleswitch driver:** sensor del cinturón de seguridad para el chofer.

**Buckleswitch passenger:** sensor del cinturón de seguridad para el pasajero.

**Seat track sensor driver 1:** Sensor de ocupación de asiento 1

**Seat track sensor driver 2:** Sensor de ocupación de asiento 2

**Seat track sensor driver 3:** Sensor de ocupación de asiento 3

**Seat track sensor driver 4:** Sensor de ocupación de asiento 4

**Analogue crashoutput:** Señal análoga indicadora de choque

**Digital crashoutput:** Señal digital indicadora de choque

**LIN crash output:** Señal de choque emitida por CAN

**CAN bus:** Bus de comunicación CAN.

**Kneebag Driver:** Bolsa para las rodillas del conductor.



**Thorax sidebag driver:** Bolsa para el tórax del conductor.

**Thorax sidebag passenger:** Bolsa para el tórax del pasajero.

**Sidebag rear left:** Bolsa trasera lateral izquierda.

**Sidebag rear right:** Bolsa trasera lateral derecha.

**Windowbag right:** Bolsa de Ventana derecha.

**Windowbag left:** Bolsa de Ventana izquierda.

**Pelvisbag driver:** Bolsa para la pelvis del conductor.

**Pelvisbag passenger or 2<sup>nd</sup> pyrofuse:** Bolsa para la pelvis del conductor.

**Pretensioner driver:** Pretensionador del conductor.

**Pretensioner passenger:** Pretensionador para el pasajero.

**Pretensioner rear left:** Pretensionador izquierdo parte trasera.

**Pretensioner rear right:** Pretensionador derecho parte trasera.

**Crashactive headrest left:** Actuador para el cuello lado izquierdo

**Crashactive headrest right:** Actuador para el cuello derecho

Las características descritas arriba pueden o no estar presentes en ciertas variantes, esto depende de las especificaciones del cliente y a demás porque incrementan el precio del producto.

Más adelante se podrá ver más a detalle la interconexión entre los bloques y los componentes principales que conforman el sistema. Por ejemplo el sistema cuenta con 6 microcontroladores que realizan tareas específicas, entre estos están:

- **Microcontrolador principal:** este ejecuta el algoritmo de control, el cual está basado en las señales de los sensores y emite señales de control para los otros microcontroladores y algunos actuadores.

- **Microcontrolador secundario:** La tarea de este microcontrolador es realizar algunos procesos a la par del microcontrolador principal, para asegurar que la decisión que tomó el micro principal sea la correcta y no haya sido provocada por una señal de ruido. Se le conoce como microcontrolador de seguridad.
- **Microcontrolador APACE:** Este microcontrolador realiza la tarea de distribuir los voltajes necesarios a cada bloque de circuito ya que cada uno maneja voltajes diferentes.
- **ATIC 55/51:** Existen 3 de estos microcontroladores en el diseño, se usan para controlar la detonación de las bolsas de aire.

Se observa además una memoria EPROM en donde se almacenan datos como: No. de serie, configuración del sistema, banco de fallas, etc. Está también el sensor angular (Pitch-over) que permiten detectar si el coche volcó hacia el frente o el sensor XY (Rollover) que detecta si el automóvil volcó hacia el lado derecho o hacia el lado izquierdo.

La comunicación se realiza mediante un protocolo llamado SPI (Serial Peripheral Interface) pero es de manera interna, las señales que envía al exterior las envía a través de un puerto de CAN.

Ahora vamos a ver más a detalle el sistema ARAMIS II a través de su diagrama eléctrico. Vamos a analizar algunos de talles sobre componentes, señales, etc., hablar sobre la naturaleza de las señales (analógicas, digitales, amplitud, frecuencia, etc.) tomaría demasiado tiempo, ya que se haría necesario hacer un análisis mucho más profundo de toda la circuitería. Sin embargo, en la medida de que una explicación necesite ser más profunda se hará lo posible para dejar el tema claro. En las Figuras 3 a la 8 se observa los diagramas eléctricos del airbag ARAMIS II.

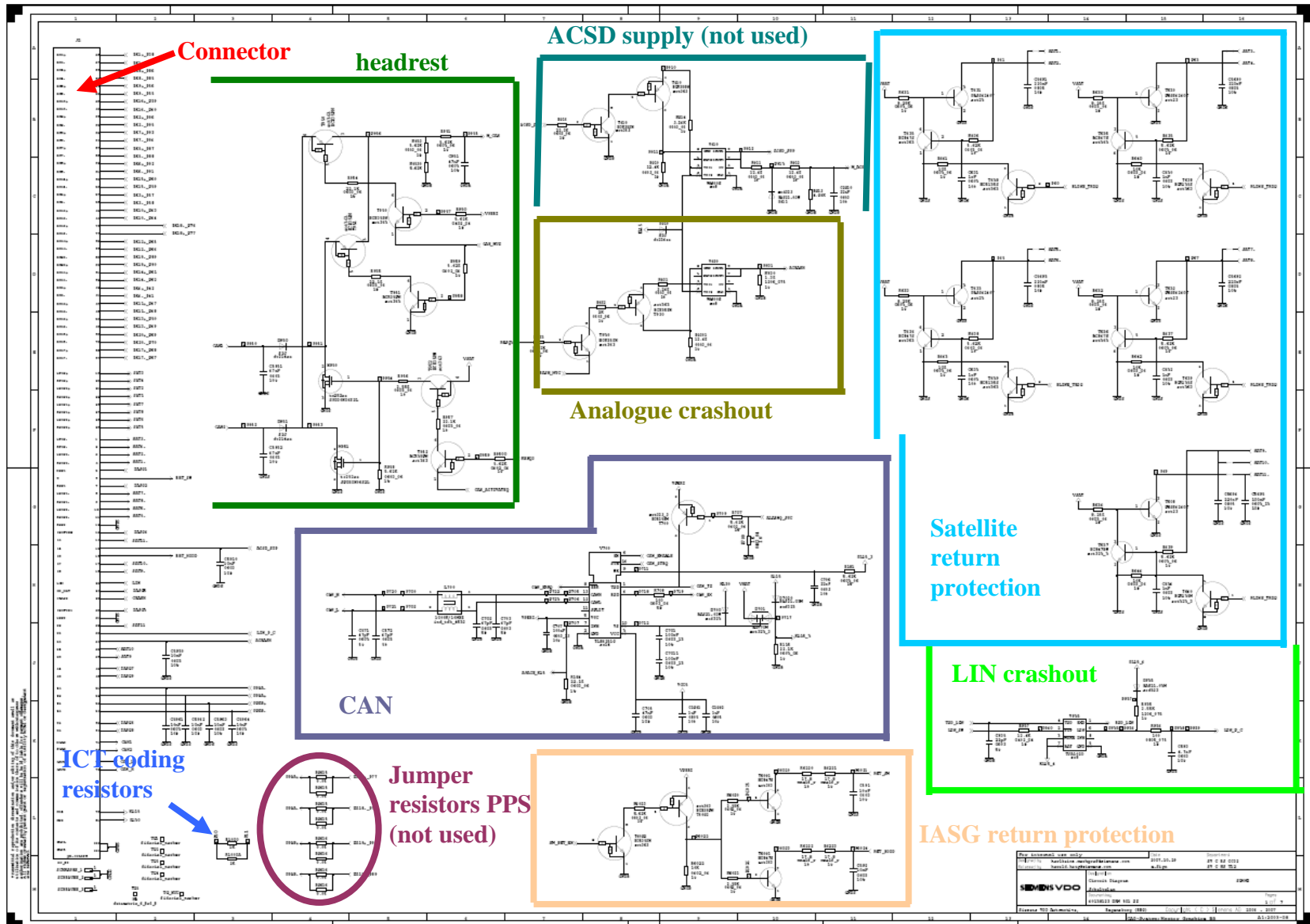


Fig. 3 Sub-bloques del airbag ARAMIS II, 1/6

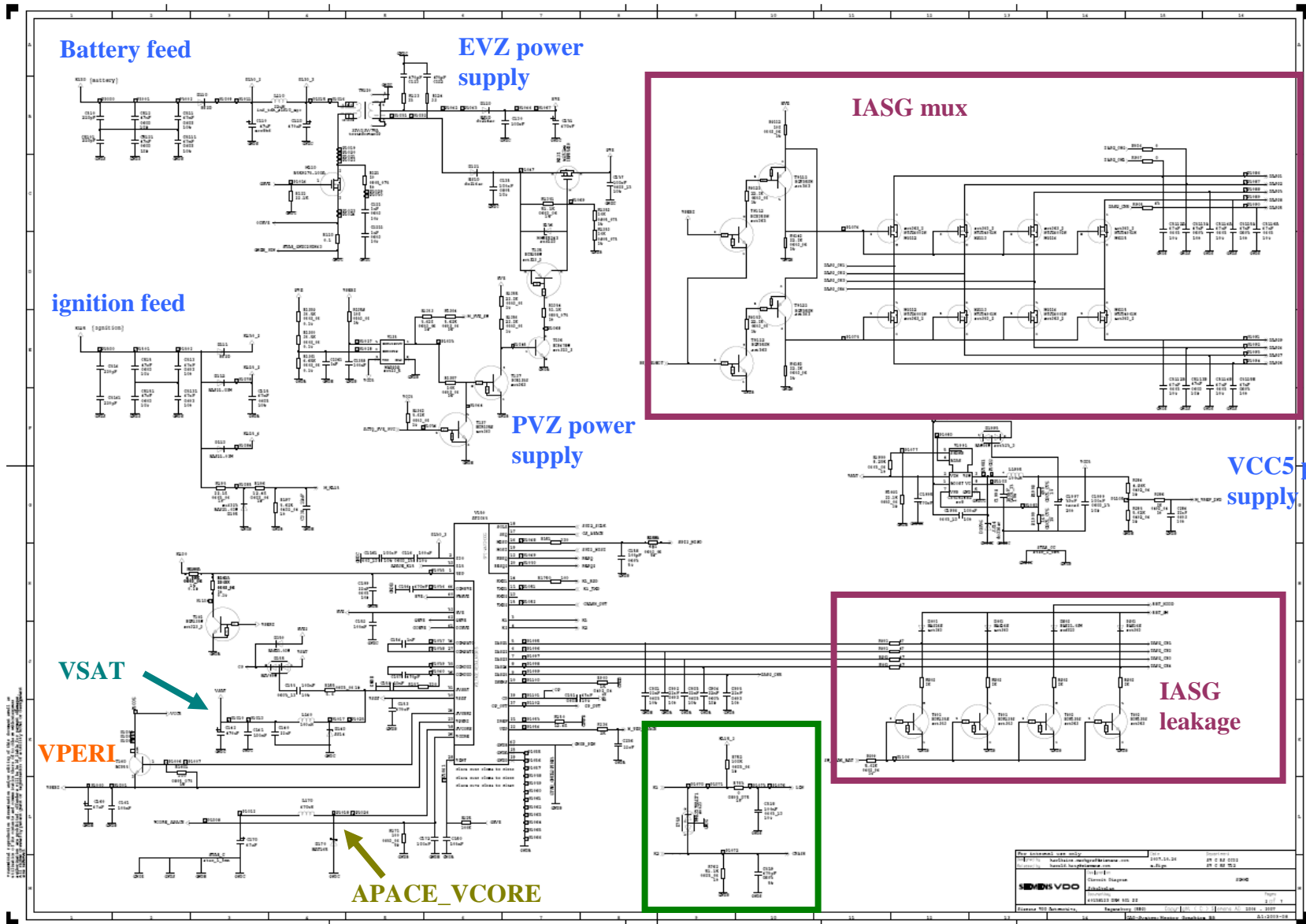


Fig. 4 Sub-bloques del airbag ARAMIS II, 2/6

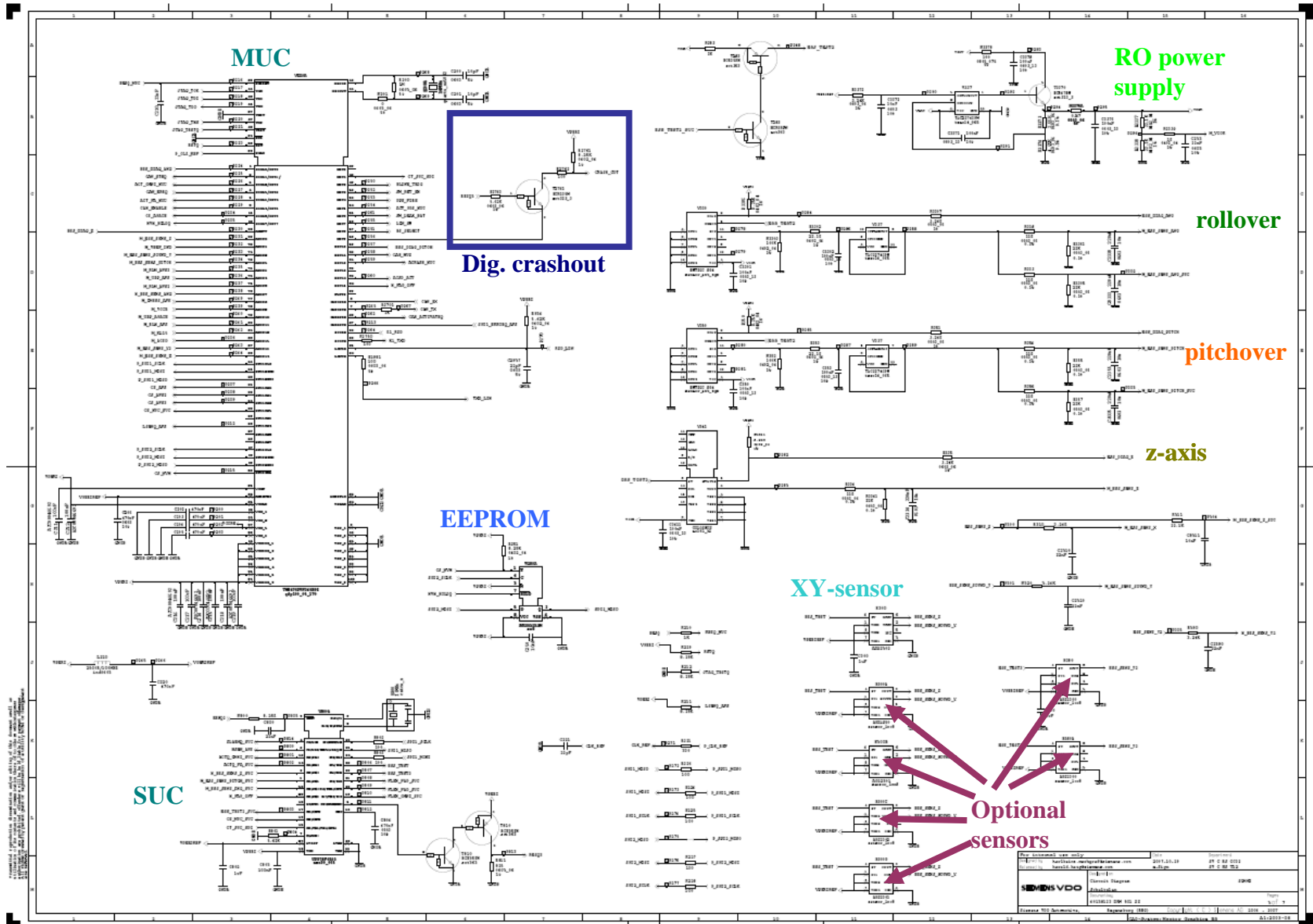


Fig. 5 Sub-bloques del airbag ARAMIS II, Pag. 3/6

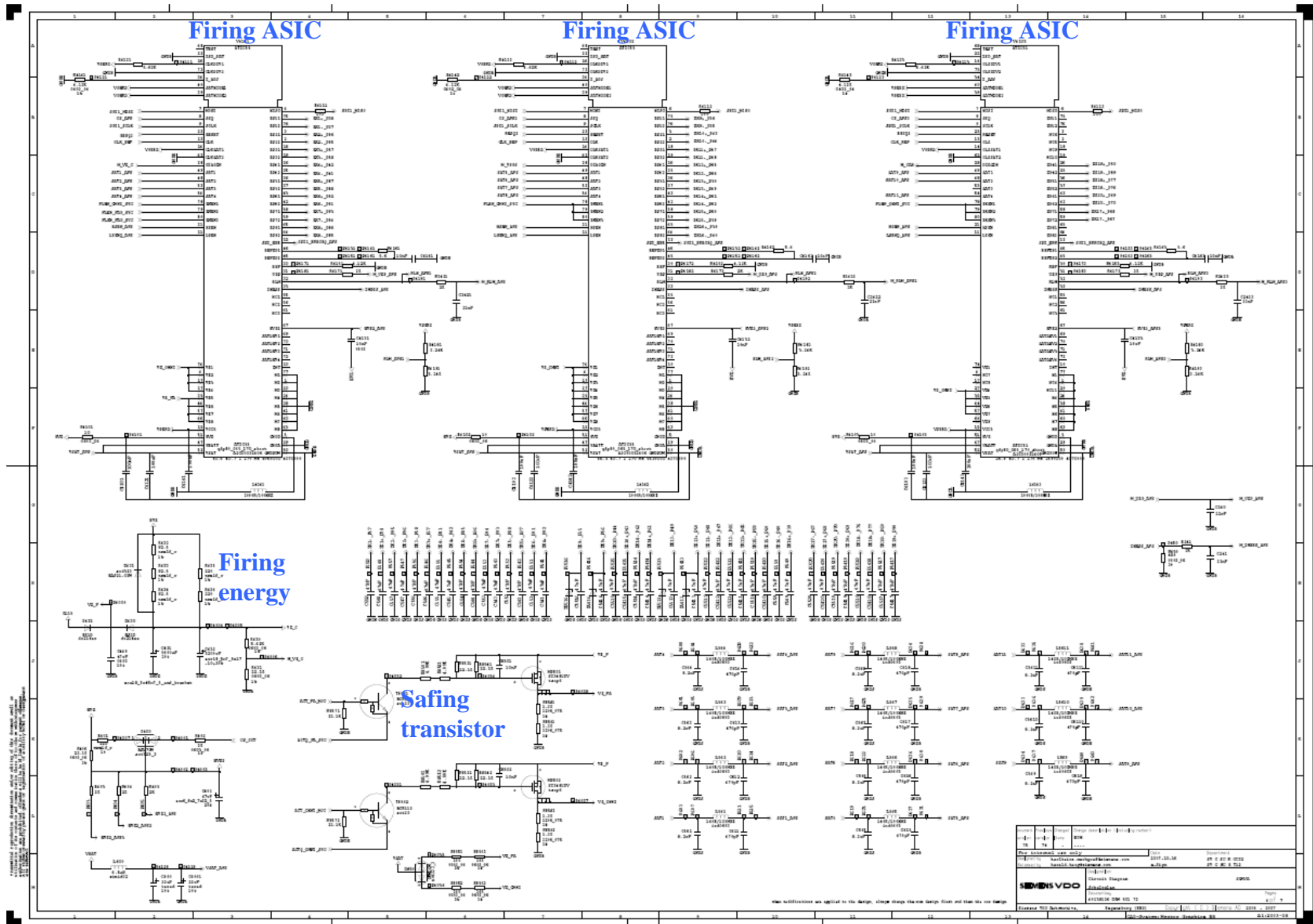


Fig. 6 Sub-bloques del airbag ARAMIS II, Pag. 4/6

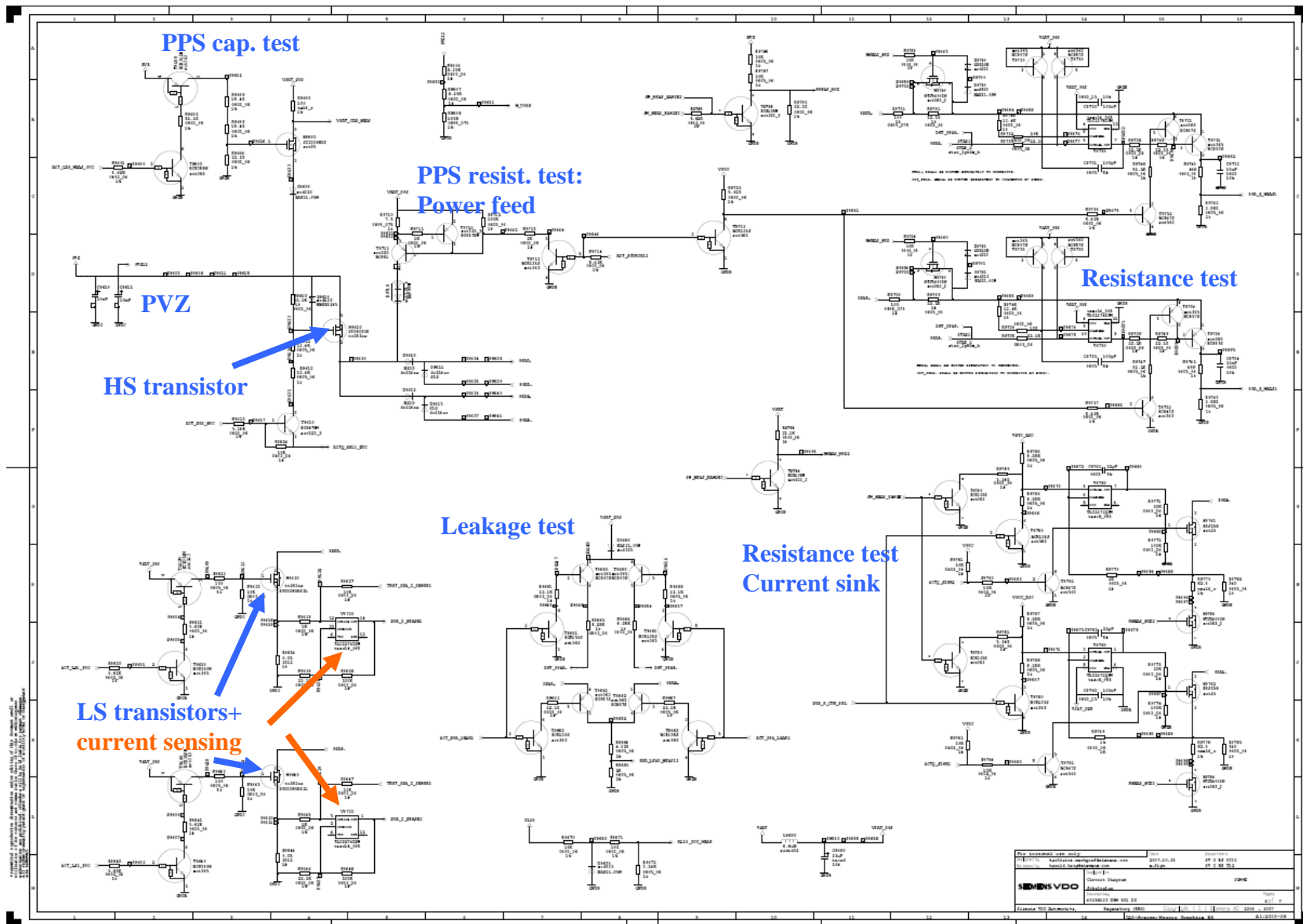


Fig. 7 Sub-bloques del airbag ARAMIS II, Pag. 5/6

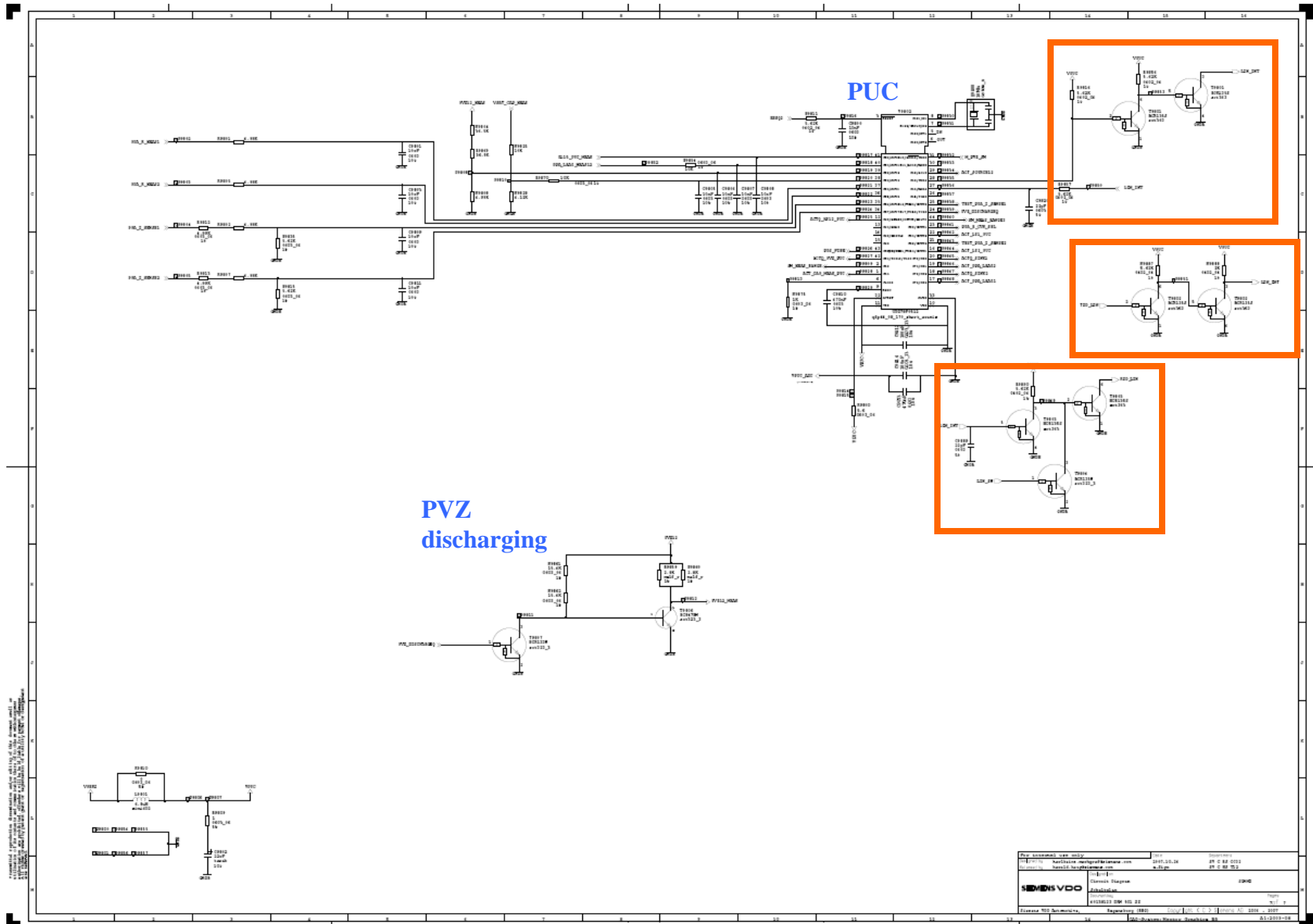


Fig. 8 Sub-bloques del airbag ARAMIS II, Pag. 6/6



### 3.5.1 Connector

El conector reúne en sus 96 pines las señales más importantes de la unidad. Se utiliza para distribuir estas señales con los distintos periféricos de la unidad como: airbags, batería, sensores, etc., a través de un arnés. La Fig. 9 muestra un listado de las señales que se encuentran el conector y el pin que corresponde a cada una.

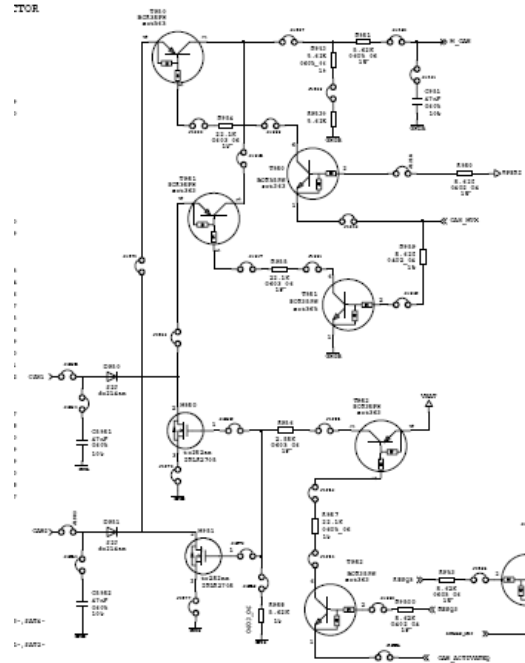
Signal Name	Pin Number
ECS LF-	C1-1
ECS RF-	C1-2
GSAT PILLAR RS2-	C1-3
GSAT PILLAR LS2-	C1-4
DR BELT SW+	C1-5
BSW, STPS and SBR RET	C1-6
PASS BELT SW+	C1-7
PSAT FRONT LS1-	C1-8
PSAT FRONT RS1-	C1-9
PSAT REAR RS3-	C1-10
PSAT REAR LS3-	C1-11
GND	C1-12
HOOD SENSOR RIGHT	C1-13
PPS SAT CENTER-	C1-14
ACSDWSS SUPPLY	C1-15
RETURN LINE HOOD	C1-16
PPS SAT LEFT-	C1-17
PPS SAT RIGHT-	C1-18
ECS LF+	C1-19
ECS RF+	C1-20
GSAT PILLAR RS2+	C1-21
GSAT PILLAR LS2+	C1-22
OCWSS LIN	C1-23
SBR	C1-24
Digital Crashout	C1-25
PSAT FRONT LS1+	C1-26
PSAT FRONT RS1+	C1-27
PSAT REAR RS3+	C1-28
PSAT REAR LS3+	C1-29
HOOD SENSOR LEFT	C1-30
GND	C1-31
PPS SAT CENTER+	C1-32
LIN crashout	C1-33
Analog Crashout	C1-34
PPS SAT LEFT+	C1-35
PPS SAT RIGHT+	C1-36
1R DRV PRET-	C1-37
1R DRV PRET+	C1-38
WINDOW BAG LEFT 1 +	C1-39
WINDOW BAG LEFT 1 -	C1-40
1R PASS PRET -	C1-41
1R PASS PRET+	C1-42
2R_SIDEBAG LEFT + / LEFT 2 COMFORT EXIT +	C1-43
2R_SIDEBAG LEFT - / LEFT 2 COMFORT EXIT -	C1-44
STPS DRV 2	C1-45
STPS PASS 2	C1-46
2R_LEFT PRET + / 1R DRV PRET +	C1-47
2R_LEFT PRET - / 1R DRV PRET -	C1-48
1R_PASS_SIDEBAG -	C1-49
1R_PASS_SIDEBAG +	C1-50
PPS ACT LEFT -	C1-51
PPS ACT LEFT +	C1-52
PPS ACT RIGHT +	C1-53
PPS ACT RIGHT -	C1-54
1R DRV_SIDEBAG -	C1-55
1R DRV_SIDEBAG +	C1-56
PYROFUSE +	C1-57
PYROFUSE -	C1-58
WINDOW BAG RIGHT 1 -	C1-59
WINDOW BAG RIGHT 1 +	C1-60
2R_SIDEBAG RIGHT + / RIGHT 2 COMFORT EXIT +	C1-61
2R_SIDEBAG RIGHT - / RIGHT 2 COMFORT EXIT -	C1-62
STPS DRV 1	C1-63
STPS PASS 1	C1-64
2R_RIGHT PRET + / 1R PASS PRET +	C1-65
2R_RIGHT PRET - / 1R PASS PRET -	C1-66
1R_PELVIS BAG PASS - / 2 <sup>nd</sup> PYROFUSE - / RIGHT 1 COMFORT EXIT -	C1-67
1R_PELVIS BAG PASS + / 2 <sup>nd</sup> PYROFUSE + / RIGHT 1 COMFORT EXIT +	C1-68
1R_PELVIS BAG DRV + / LEFT 1 COMFORT EXIT +	C1-69
1R_PELVIS BAG DRV - LEFT 1 COMFORT EXIT -	C1-70
CAH1	C1-71
CAH2	C1-72
OPEN	C2-73
CANL	C2-74
CANH	C2-75
VENTHOLE_PASS - / RIGHT RO BAR -	C2-76
VENTHOLE_PASS + / RIGHT RO BAR +	C2-77
OPEN	C2-78
KL15R, ignition (radio)	C2-79
OPEN	C2-80
KL30, permanent battery	C2-81
OPEN	C2-82
OPEN	C2-83
OPEN	C2-84
DRV_AB1 -	C2-85
DRV_AB1 +	C2-86
PASS_AB1 +	C2-87
PASS_AB1 -	C2-88
VENTHOLE_DRV - / LEFT RO BAR - / KNEE BAG PASS -	C2-89
VENTHOLE_DRV + / LEFT RO BAR + / KNEE BAG PASS +	C2-90
PASS_AB2 -	C2-91
PASS_AB2 +	C2-92
DRV_AB2 +	C2-93
DRV_AB2 -	C2-94
KNEE_BAG_DRV -	C2-95
KNEE_BAG_DRV +	C2-96

**Fig. 9. Conector y descripción de Pines**

### 3.5.2 Headrest

Este bloque permite al sistema activar un actuador colocado en la cabecera del conductor con fin de amortiguar el golpe en el cuello o evitar el movimiento brusco de la cabeza lo cual puede provocar lesiones. En la Fig. 10 se muestra el diagrama eléctrico que permite desempañar este trabajo. Las señales principales que actúan sobre este circuito son:

- **M\_CAH:** Esta señal está relacionada con un ATIC 65 quien se encarga de activar los actuadores Crash Active Headrest.
- **CAH\_MUX:** esta línea está conectada directamente al pin 60 del MUC. Por lo que podemos saber que se trata de una señal de control también.
- **CAH1 y CAH2:** Estas son las señales que están conectadas a los actuadores Crash Active Headrest izquierdo y derecho respectivamente. Están conectadas a estos a través del conector y el arnés en los pines 71 y 72.
- **VSAT y VPERI:** Son los voltajes necesarios para el circuito. Los provee el APACE y tienen los valores de: 9.1 para VSAT y 3.3 para VPERI.
- **CAH\_ACTIVATEQ:** Línea usada para enviar mensajes CAN relacionados con el CAH.
- **RESQ3:** Señal usada para establecer comunicación con el SUC.



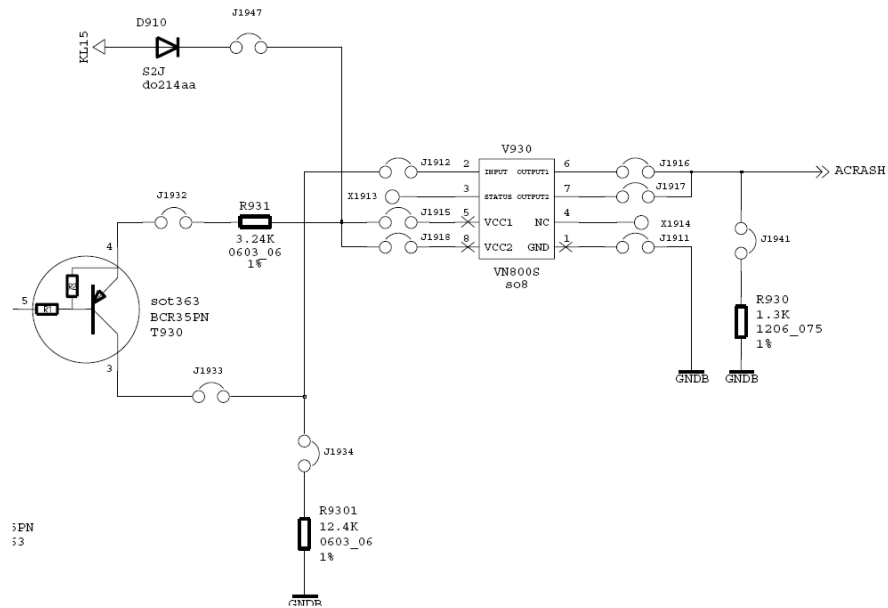
*Fig. 10. Diagrama eléctrico del Headrest*

### 3.5.3 Analogue Crashout

Cuando la unidad detecta un choque, se generan tres tipos de señales, una digital, una analógica y un mensaje de CAN. El analogue Crashout es el circuito que se encarga de generar la señal de tipo analógica (Fig. 11). Las señales que se usan en este caso son:

- **ACRASH\_MUC:** Es la señal que el circuito recibe del MUC para determinar el estado de choque.
- **RESQ3:** Esta es una señal correspondiente al SUC. Esto nos da a entender que el analoque Crashout no genera una señal de salida si el SUC no se le indica.

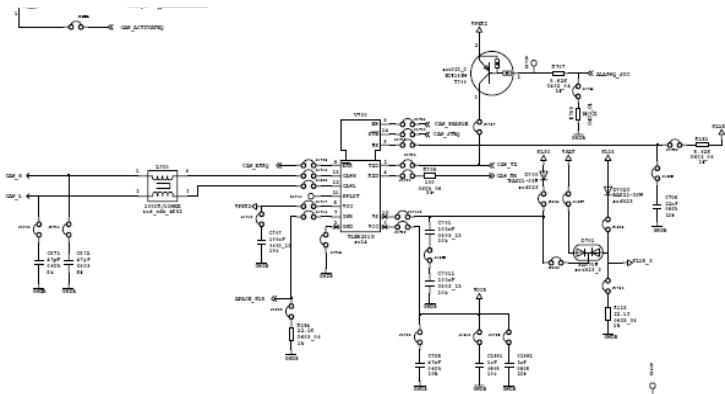
- **ACRASH:** Esta es la señal que se genera en el circuito. Está conectada en el pin 34 del conector.



*Fig. 11 Diagrama eléctrico del Analogue Crashout*

### 3.5.4 CAN

Se le conoce como CAN a un protocolo de comunicación muy utilizado en la tecnología automotriz, todos los dispositivos electrónicos de los automóviles lo usan para poder establecer comunicación con la computadora del automóvil. El proyecto ARAMIS II tiene integrado una circuitería que le permite enviar y recibir mensajes mediante el bus can del automóvil. En la Fig. 12 Se puede observar la configuración electrónica de dicha circuitería.



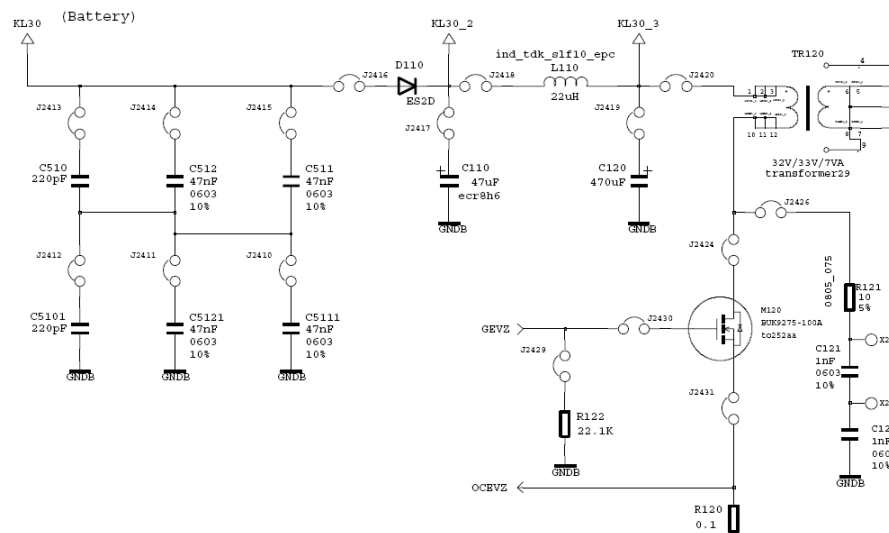
*Fig. 12 Diagrama eléctrico CAN*

Esta comunicación es posible ya que se usa un CI V700 el cual es un transmisor/receptor de mensajes de CAN, que tiene como característica principal su alta velocidad de transmisión de datos esto lo hace apto para usarse en aplicaciones automotrices.

A continuación se describe cada una de las señales que se conectan al V700 para poder realizar su función:

- **CAN\_TX:** Esta señal se obtiene del pin 1 del V700. Y se trata de la salida de datos o mensajes ya codificados.
- **CAN\_RX:** Es el pin 4 del V700 y se usa para recibir los mensajes de CAN
- **VUC:** El V700 nos da la opción de configurar el nivel lógico alto a un voltaje que nosotros deseemos. En este caso configuramos el VUC a 3.1V. que es VPERI.
- **CAN\_ENABLE:** Esta es la señal que habilita la recepción/transmisión de mensajes de CAN. Es una señal emitida por el MUC.
- **APACE\_K15:** Esta señal es un voltaje

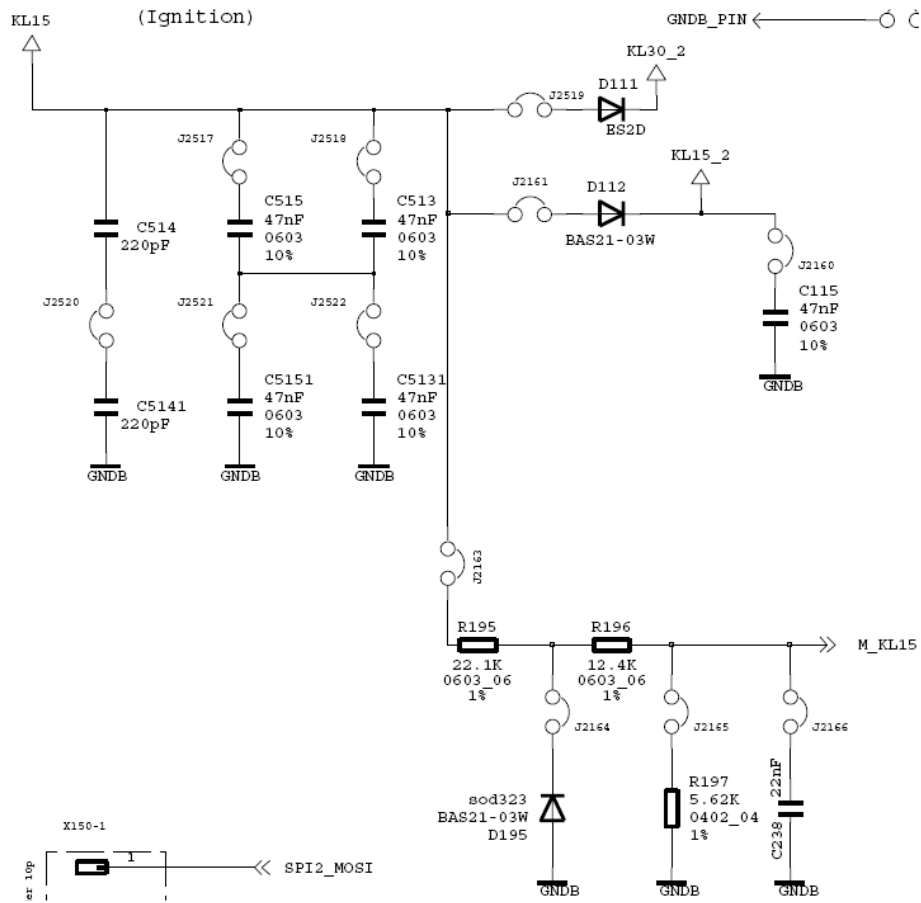
### 3.5.5 Battery Feed



*Fig. 13. Circuito de alimentación de batería*

En la Fig. 13 podemos observar el circuito que acondiciona la señal de alimentación de la batería para poder transformar este voltaje a valores que se necesitan en los distintos módulos de la unidad. Más adelante analizamos estos voltajes.

### 3.5.6 Ignition Feed

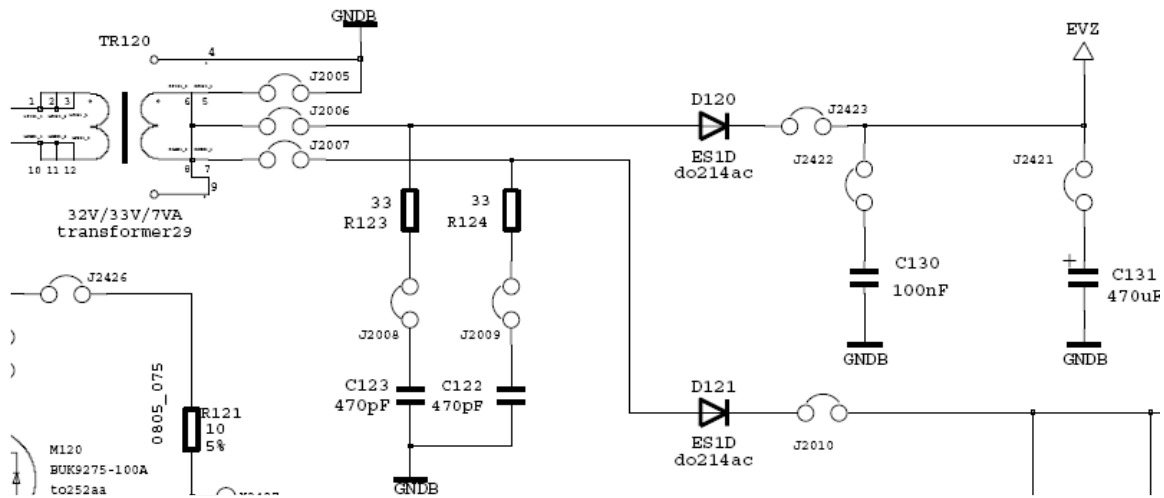


**Fig. 13** Circuito de alimentación del sistema de ignición

Mediante este circuito se monitorea el estado del sistema de ignición ya que la unidad necesita estas señales para ejecutar sus algoritmos. (Ver Fig. 13)

### 3.5.7 EVZ power supply

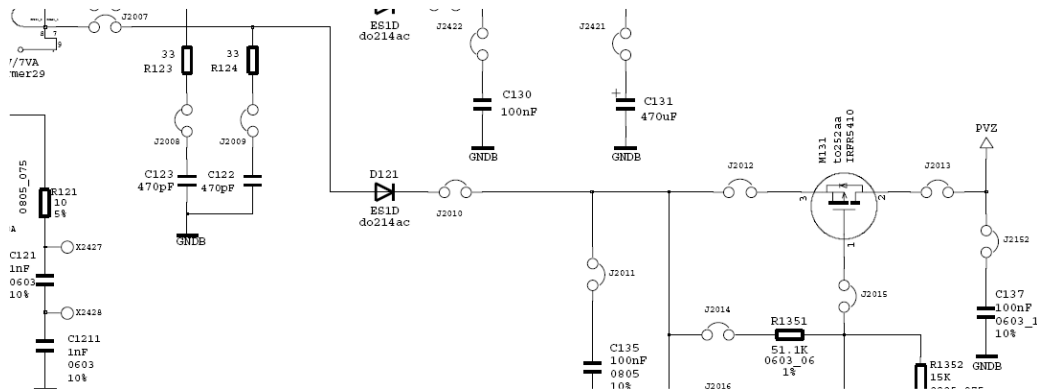
EVZ es un voltaje de valor nominal de 30V., es requerido en el sistema para distintos dispositivos y se obtienen directamente del alternador mediante un transformador, como se observa en la Fig. 14.



**Fig. 14 EVZ fuente de alimentación**

### 3.5.8 PVZ power supply

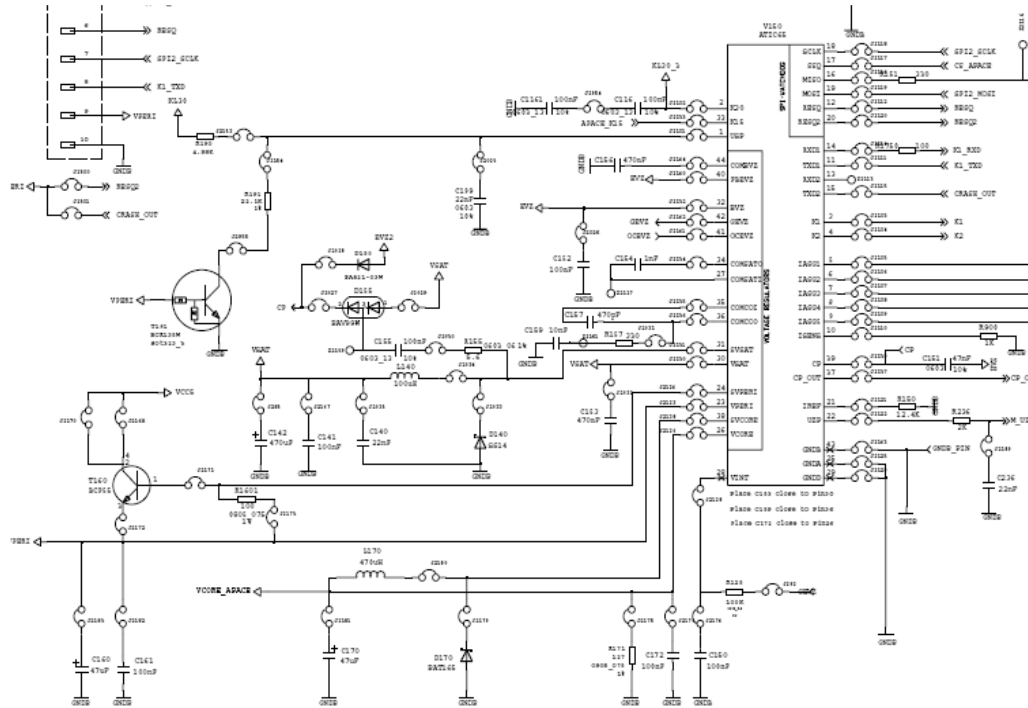
Esta es la línea de alimentación por la que se suministra la energía de reserva. Esta energía se suministra cuando la fuente de alimentación principal de la unidad es desconectada (batería) y permite que el sistema continúe activo por unos cuantos milisegundos más, por si es necesario activar las bolsas en ese momento. Se usan dos capacitores los cuales se encargan de almacenar la energía necesaria. (Véase Fig. 15)



**Fig. 15 PVZ fuente de alimentación**

### 3.5.9 Vsat, VPeri, VCore

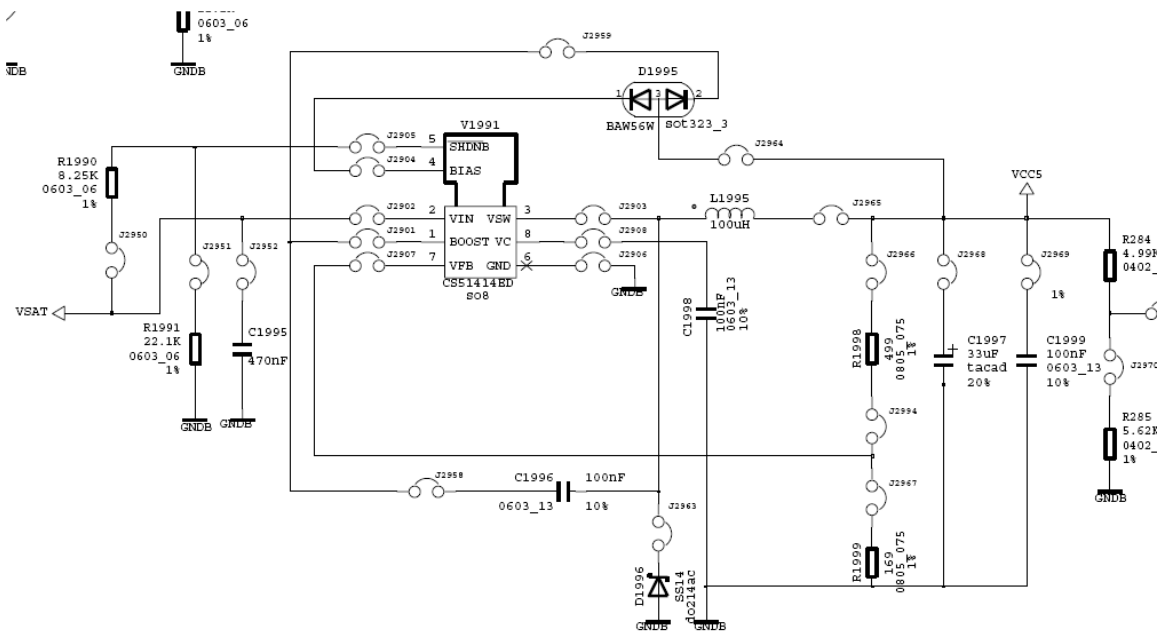
Estos voltajes son generados por un microcontrolador conocido como ATIC65 APACE (véase Fig. 16). En el caso de V<sub>Sat</sub> es un voltaje utilizado para alimentar los satélites y tiene un valor de 9.1 volts. V<sub>Peri</sub> es usado para alimentar a los circuitos digitales de la unidad tal como los microcontroladores, entrega un voltaje de 3.1 Volts. Y el V<sub>Core</sub> es un voltaje que es usado solo por el mismo APACE, su valor es de 1.8V.



*Fig. 16. Generación de voltajes V<sub>sat</sub>, V<sub>Peri</sub>, V<sub>Core</sub>*

### 3.5.10 VCC5 power supply

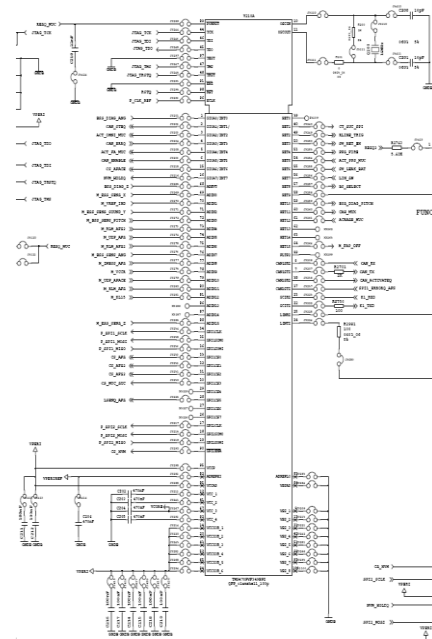
Este voltaje es generado por una fuente conmutada que entrega aprox. 5V. Es un nivel de voltaje usado para la alimentación de algunos circuitos integrados con funciones específicas. En la Fig. 17 se observa que VCC5 es generado por un el CI CS51414ED.



**Fig. 17. Generación de VCC5**

### 3.5.11 MUC (Main Microcontroler)

Este es el microcontrolador principal, quien se encarga de realizar las tareas más importantes de la unidad como: monitorear, ejecutar algoritmo principal, emitir señales de control a los demás microcontroladores, etc. En la Fig. 18 se observan las conexiones de este integrado.



**Fig. 18 Diagrama de conexionado del MUC**



### 3.5.12 SUC (Safig Microcontroler)

El SUC Es un microcontrolador que se utiliza para brindar seguridad sobre las decisiones que toma el MUC, es decir, existen señales que no solo son procesadas por el MUC si no también por el SUC para tener la certeza de que la decisión que se ha tomado respecto a esas señales sea correctas y no solo producto de lecturas erróneas o cualquier otro defecto que pudo ser considerado. Por ejemplo: las señal **ME\_ESS\_SENS\_ANG** está conectada al pin número 76 del MUC (Fig. 18), también al pin número 30 del SUC aunque en este caso la recibe el nombre de **ME\_ESS\_SENS\_ANG\_SUC** pero en realidad es la misma señal (Fig. 19). Esto se hace con el propósito de asegurar que la señal leída del sensor angular sea correcta y también la decisión tomada por un el MUC sea respaldada por el SUC.

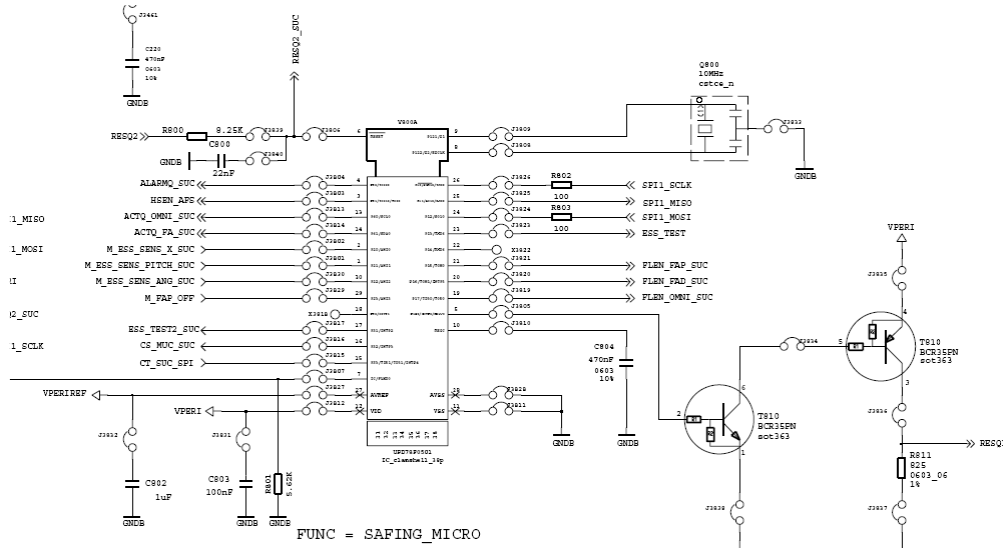


Fig. 19 Diagrama de conexionado del SUC

### 3.5.13 Digital Crashout

Al igual que el Analogue Crashout, existe una señal en formato digital para indicar que ha ocurrido un choque. Es una señal que se genera en el MUC y es transmitida a través de un transistor marcado como T2761. (Véase Fig. 20)

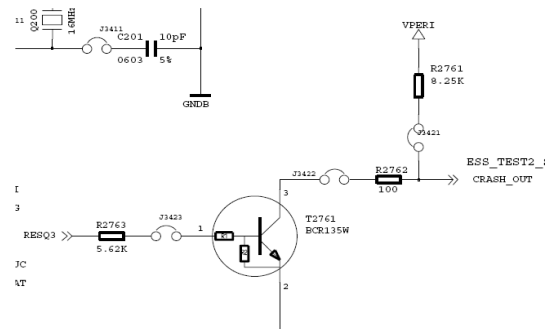


Fig. 20 Digital Crashout

### 3.5.14 EEPROM

En esta memoria el sistema almacena información sobre la configuración de hardware de la unidad, es decir, número de bolsas de aire colocadas en la unidad, valores de referencia de los sensores, contador de choques, etc. También se almacenan registros con información sobre los errores que se han generado en la unidad, o el número de choques, etc. Con esto se puede realizar un análisis de las actividades de la unidad.

Tiene una capacidad de 64Kbits, Se programa mediante comandos SPI, como podemos ver en la Fig. 21 la circuitería es muy sencilla solo son necesarios algunas señales para el buen funcionamiento de la memoria, como son:

- **Chip Select (S).** Cuando esta entrada tiene una señal en nivel alto, el dispositivo es deseleccionado y Serial Data Output (Q) se pone en alta impedancia, a menos que un ciclo de escritura este en progreso, de cualquier otra forma el dispositivo estará en Standby. Para seleccionar el dispositivo debemos proporcionarle una señal de nivel bajo a S, entonces comenzará a funcionar.
- **Serial Data Output (Q):** Esta señal es usada para transferir datos de forma serial hacia otros dispositivos. Los datos son transmitidos cada flanco de bajada de la señal de reloj.
- **Write Protect (W):** El propósito principal de esta señal es retener cierto bloque de memoria que es protegido por las instrucciones de escritura. Este pin debe ser estable durante todas las operaciones de escritura,
- **Serial Data Input (D):** Esta señal es usada para transferir datos de forma serial hacia el dispositivo. Esta recibe instrucciones, direcciones y datos que deben ser escritos. Los datos son asegurados con cada flanco de subida de la señal de reloj (C).
- **Serial Clock (C):** Esta señal provee temporización a la interfaz serial. Instrucciones, direcciones, o datos presentes en Serial Data Input (D) son fijados con los flancos de subida de esta señal mientras que los datos en Serial Data Output (Q) son retenidos con los flancos de bajada.
- **Hold (HOLD):** Esta señal es usada para pausar la comunicación con cualquier dispositivo sin necesidad de deseleccionar el dispositivo (Chip Select). Mientras la memoria esta en modo HOLD la Serial Data Output (Q) esta en alta impedancia. Para iniciar el modo HOLD, el dispositivo debe estar seleccionado, con el pin Chip Select en nivel bajo.

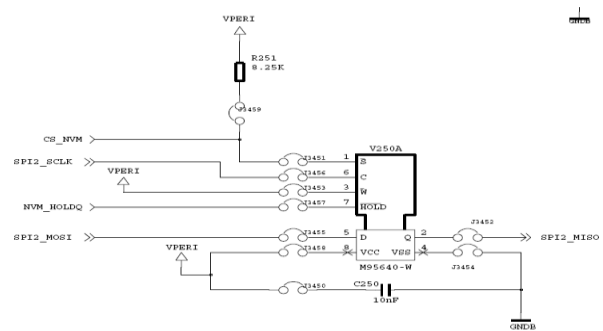
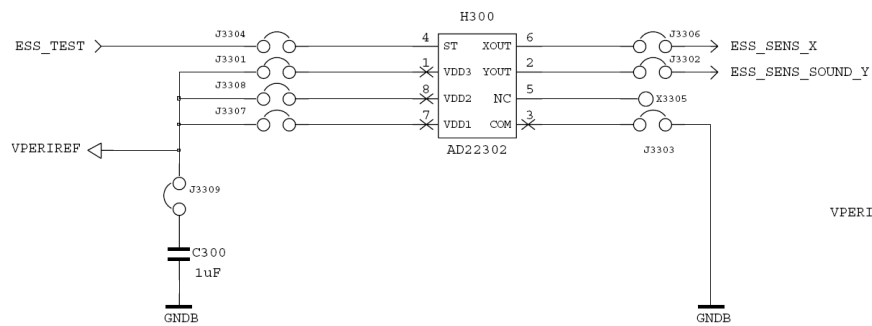


Fig. 21 Memoria EEPROM

### 3.5.15 XY Sensor

Este bloque de la unidad se encarga de detectar desaceleraciones tanto en los ejes X y Y. Esto es posible mediante un sensor llamado AD22302 el cual es un completo sistema de medición de aceleración de doble eje en un solo encapsulado. En la Fig. 22 se observan las conexiones de este bloque, es un bloque muy sencillo, ya que al CI solo es necesario dotarle de alimentación y nos entrega en sus terminales 6 y 2 las mediciones de los ejes X y Y respectivamente. Además cuenta con una terminal llamada self-test, ya que el sensor es capaz de realizar un autodiagnóstico e informarnos los resultados mediante este pin.

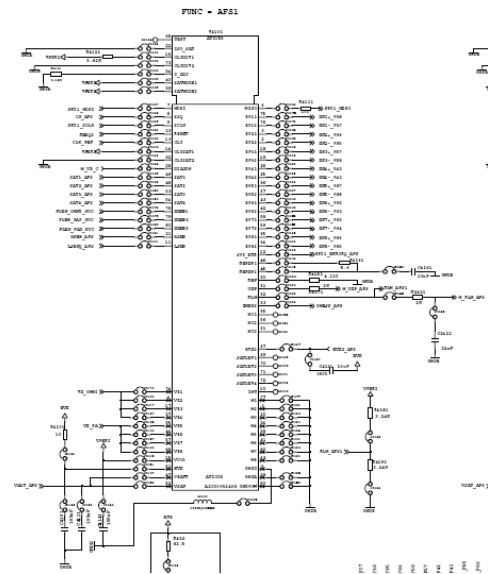


**Fig. 22. Diagrama de conexionado del sensor XY**

### 3.5.16 Firing ASIC

Este bloque está conformado por 3 microcontroladores conocidos como ATIC55. Estos micros son los encargados de detonar las bolsas de aire cuando el coche se encuentra en una situación de choque. Dos de estos micros manejan 8 airbags mientras que el último solo controla a cuatro. Es posible controlar todas las bolsas con un solo microcontrolador, sin embargo, en este caso es mejor controlarlos de micros diferentes para que si se llega a presentar una falla en uno de los micros no afecte a todas las bolsas si no solo a una parte de ellas, aunque el hecho de existir un error en este bloque ya es una situación muy crítica.

Cabe mencionar también que estos microcontroladores procesan un algoritmo también



**Fig. 23 Firing ASIC Circuit**

para detectar situaciones de choque, se supone que es una tarea del MUC, pero esto se hace para incrementar la seguridad de las decisiones. En la Fig. 23 observan las conexiones de uno de estos micros

### 3.5.17 PS Cap. Test

Este circuito se usa para hacer pruebas a unos capacitores que pertenecen al bloque del Pedestrian Protection System (PPS). Lo que se mide básicamente es su capacitancia y los tiempos de descarga. Es necesario el empleo de este circuito porque la función de los capacitores que se miden es muy importante por lo tanto es necesario monitorear su estado. Se trata principalmente de dos capacitores de 10mF, que tienen la función de brindar la energía necesaria para detonar las bolsas en caso de que al presentarse el choque el automóvil se quede sin energía. (Véase Fig. 24)

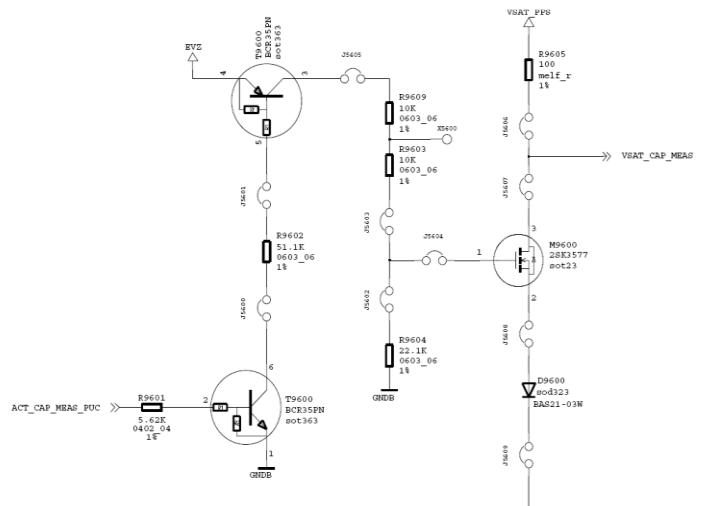


Fig. 24. Capacitor Test Circuit

### 3.5.18 Resistance Test

También existe un circuito que se emplea para realizar pruebas a algunas resistencias el cual consta de un amplificador operacional usado como comparador para determinar su valor. En la Fig. 25 observamos el circuito, cabe mencionar que las pruebas se realizan en modo cíclico con el fin de monitorizar y poder reportar errores a la unidad cuando se presente algún fallo. Se analizan aquellas resistencias que afectan de manera significativa el funcionamiento de la unidad.

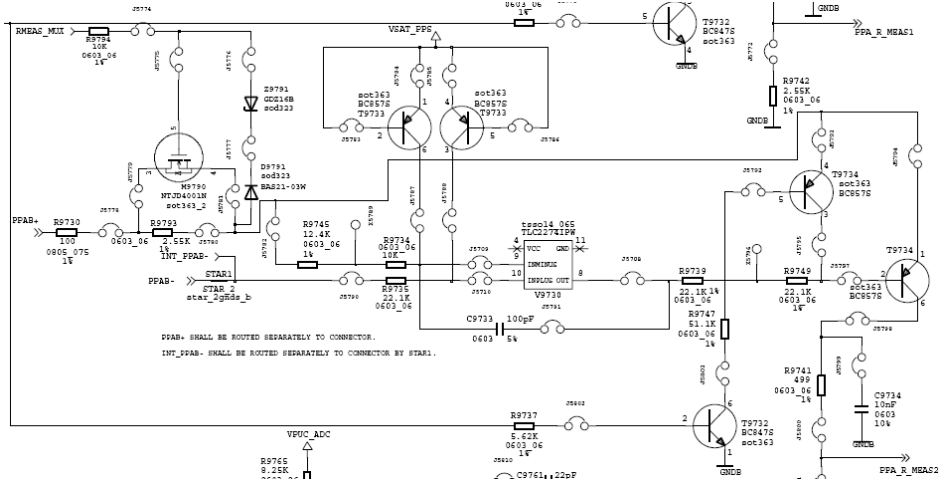
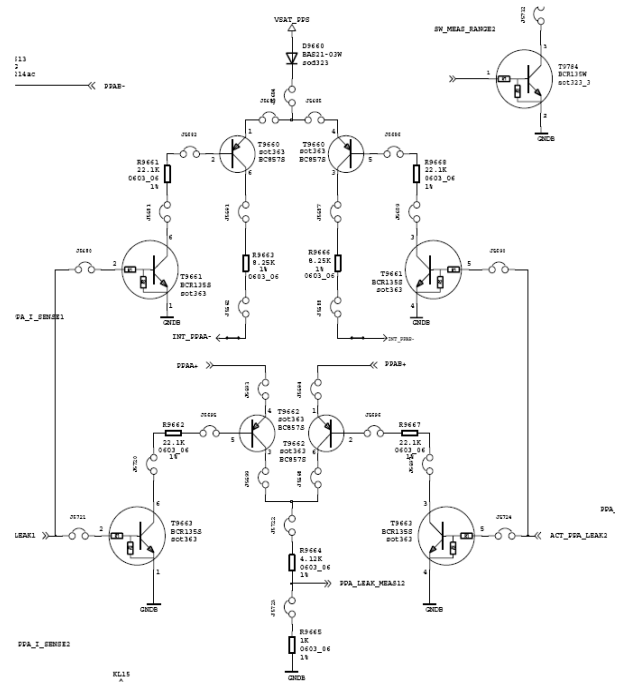


Fig. 25. Resistance Test Circuit

### 3.5.19 Leakage Test

La prueba de Leakage es aquella que se hace para detectar conexiones a tierra o a batería de algún componente que no deba tener conexiones a tierra. Esto se hace mediante el método de detección de fugas de corriente. El circuito de la Fig. 26 muestra las conexiones necesarias para realizar estas mediciones.



**Fig. 26. Leakage Test Circuit**

### 3.5.20 PUC

La unidad ARAMIS II cuenta con un sistema llamado Pedestrian Protection System (PPS), consiste en dotar al automóvil de la capacidad de poder amortiguar el golpe contra peatones cuando ocurre un atropellamiento mediante un ligero levantamiento del cofre, como se ve en la Fig. 27. El PUC (Pedestrian microcontroller) es el microcontrolador que se encarga de realizar esta tarea principalmente aunque también realizar otras como el monitoreo de las pruebas de medición. En la Fig. 27 se observa el conexionado de este micro.



**Fig. 27 Pedestrian Protection System**

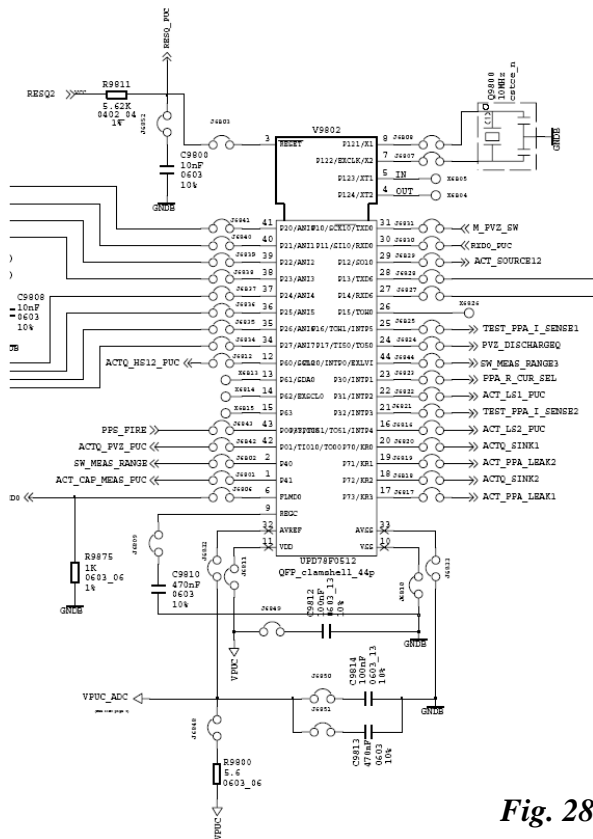


Fig. 28 Microcontrolador PUC

### 3.5.21 PVZ Discharging

Este circuito es muy simple su tarea es descargar los capacitores del PPS. Como se puede notar en la Fig. 29 la descarga está controlada por el PUC mediante la señal que se encuentra en la base del transistor T9807 llamada *PVZ DischargeQ*, que viene directamente del PUC.

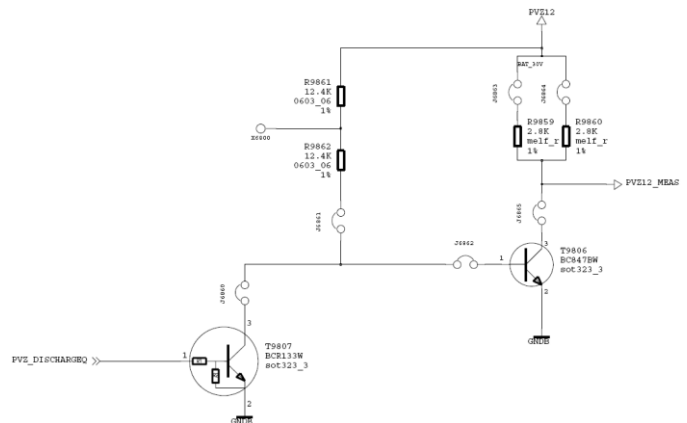
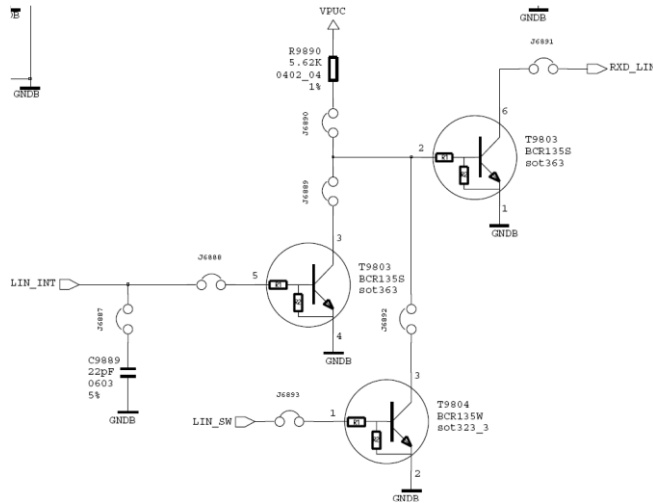


Fig. 29 PVZ Discharging Circuit

### 3.5.22 Line Communication PUC-MUC

Existe comunicación entre el PUC y MUC mediante un protocolo denominado LINE, que está implementado en la unidad pero solo se utiliza para comunicar a estos dos microcontroladores. Ver Fig. 30



**Fig. 30 Comunicación LINE**

## 4. FEA TESTING

La actividad principal que se realizó a lo largo de este trabajo fue la de llevar a cabo una serie de pruebas con las cuales se pretendía confirmar el correcto funcionamiento del ARAMIS II.

Las pruebas se realizaron en un módulo electrónico conocido como FEA (Failure Effect Analysis) este módulo contiene el montaje de la circuitería del ARAMIS II tal como se describió en el capítulo anterior, sin embargo este módulo nos permite más accesibilidad a cada componente de la circuitería en comparación con el módulo comercial del mismo sistema.

El módulo FEA como su nombre lo dice nos permite realizar fallas al módulo y analizar los efectos que estas fallas provocan en el sistema.

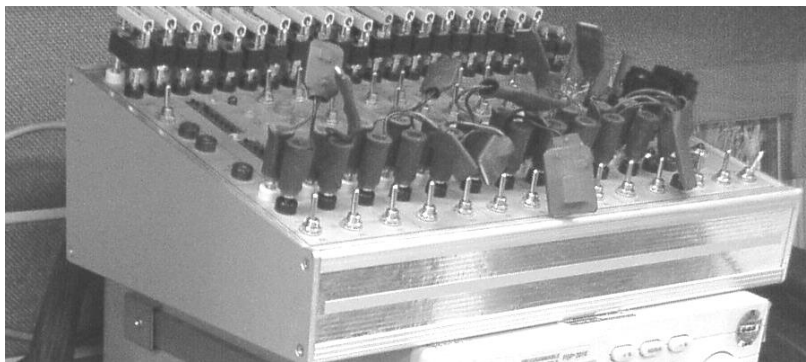
### 4.1 Ejecución de las fallas de cortos y abiertos

Para poder realizar las fallas, primeramente debemos de contar con un banco de pruebas, este banco es proporcionado por el responsable del proyecto y fue diseñado en base a un análisis completo del circuito del sistema. En su totalidad existen aproximadamente 3394 pruebas. Para poder realizarlas todas se necesita que varias personas realicen una parte de estas.

Es necesario también contar con herramienta de software y hardware para facilitar el diagnóstico de las fallas. A continuación se da una breve explicación del manejo y función de estas herramientas.

### 4.2 Descripción de herramienta de diagnóstico

La herramienta principal para realizar el diagnóstico de fallas es un módulo de simulación que cuenta con sensores de choque, presión, etc., conocido como LoadBox. También cuenta con las resistencias conocidas como Squibs que se encargan de detonar las bolsas de aire, en este caso son consideradas como los actuadores. El módulo tiene además una lámpara indicadora de fallas, un puerto de comunicación RS232 para establecer comunicación con la PC y algunas otras señales que necesita el sistema del airbag para su funcionamiento. (Véase Fig. 31)



*Fig. 31 LoadBox para ARAMIS II*



Existe un software llamado PCDiagNT el cuál usa el puerto RS232 de la PC con el fin de establecer comunicación con la Loadbox y así poder mostrar de forma gráfica los detalles de los errores que se pueden presentar.

Se cuenta también con un software de apoyo el cual permite ubicar un componente dentro de la tarjeta. Este se llama PCBCAM y mediante una búsqueda personalizada señala la posición física del componente. Este software es muy necesario ya que de otra forma sería muy complicado encontrar un componente dentro de otros 1000 aproximadamente.

### 4.3 Procedimiento para realizar las fallas

A continuación se dará una breve explicación del procedimiento que se siguió para realizar las fallas y la forma de utilizar el equipo de diagnóstico. Para iniciar es necesario conectar la loadbox al puerto RS232 de la PC. Después de esto se conecta la FEA mediante un arnés con la loadbox y además es necesario abrir el programa de diagnóstico PCDiagNT y el PCBCAM para poder localizar cada componente que se requiera.

**Paso 1:** El primer paso es elegir una prueba del banco de pruebas, este banco esta contenido dentro de un documento de Excel y contiene:

- El nombre y número de pin del componente al que se le realizará la prueba
- La función que realiza ese componente
- El tipo de fallo que se aplicará (Corto o Abierto)

También cuenta con campos que se deben llenar a la hora de realizar la prueba, estos campos son:

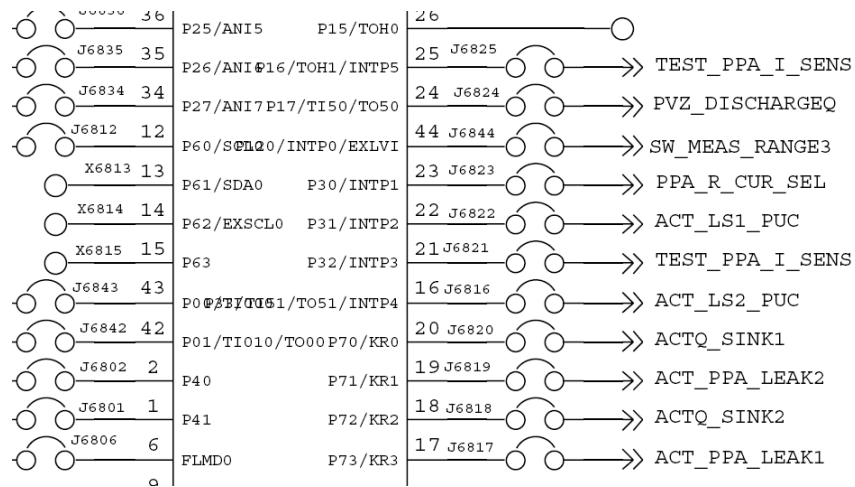
- Estado de la lámpara de Fallas, ON u OFF (Realiza la función de la lámpara ENGINE del automóvil)
- Códigos de errores que se produjeron
- Apreciación, si no se pudo realizar la falla se especifica porqué.
- Versión de software con el que se hizo el diagnóstico
- Iniciales de la persona que realizó la prueba

Para ejemplificar el procedimiento vamos a elegir una prueba aleatoria. Elegimos la prueba número 3161, que se refiere al componente V9802 (ver figura No. 32). Esta prueba indica que debemos hacer un corto (short) entre los pines 15 y 16 de dicho componente. La celda marcada con un \* corresponde a la función que realiza el componente, pero en este caso fue omitida.

•	3160	V9802-15	P63	Open
•	3161	V9802-15/16	*	Short
•	3162	V9802-16	ACT_LS2_PUC	Open

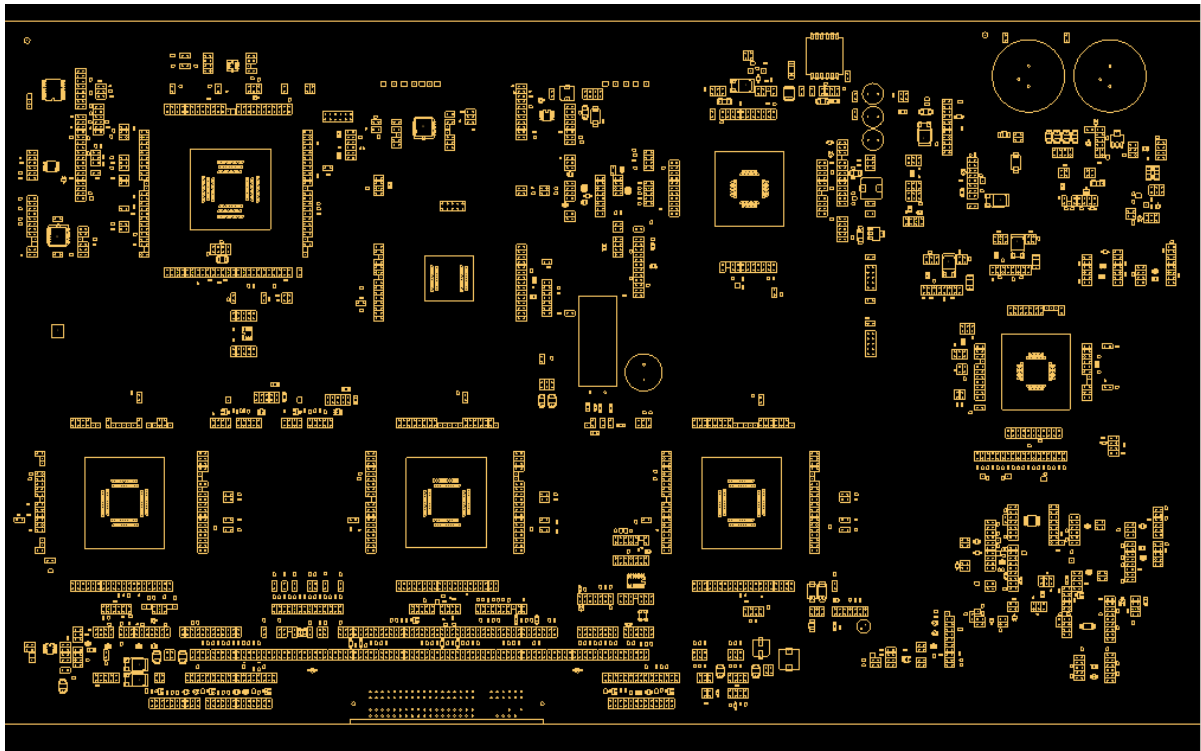
**Fig. 32 Prueba elegida del banco de pruebas**

**Paso 2:** Ya que sabemos a qué componente nos vamos a dirigir, realizamos una inspección en el diagrama esquemático para asegurarnos que este corto circuito no va afectar de manera significativa a la unidad. En la figura 33 se muestra la conexión del componente V9802 y sus pines 15 y 16. Como podemos observar no hay ningún problema aparente si se realiza el corto circuito. También es útil checar el número de Jumper de los pines, que para este caso es X6815 para el pin 15 y J6816 para el pin 16, esto permitirá la fácil ubicación de los pines en la tarjeta, ya que no hay acceso a ellos sino nada más a través de los Jumpers.



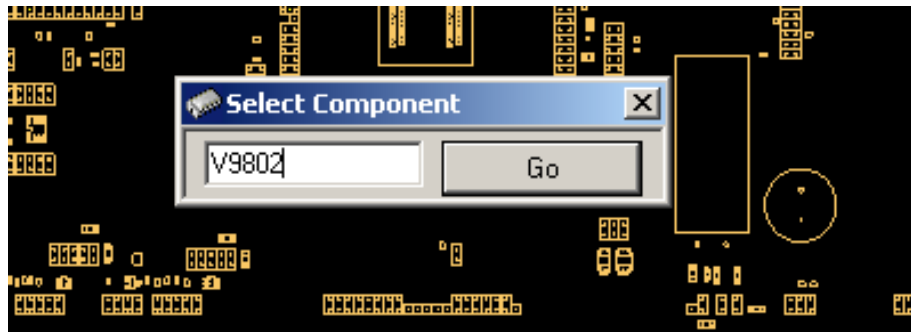
**Fig 33 Ubicación en el diagrama esquemático de los pines 15 y 16 del componente V9802**

**Paso 3:** El siguiente paso es ubicar el componente en la tarjeta. Para esto vamos a usar el software PCBCAM, el cual nos muestra inicialmente la tarjeta completa con la posición real de cada componente. (Véase fig. 34)



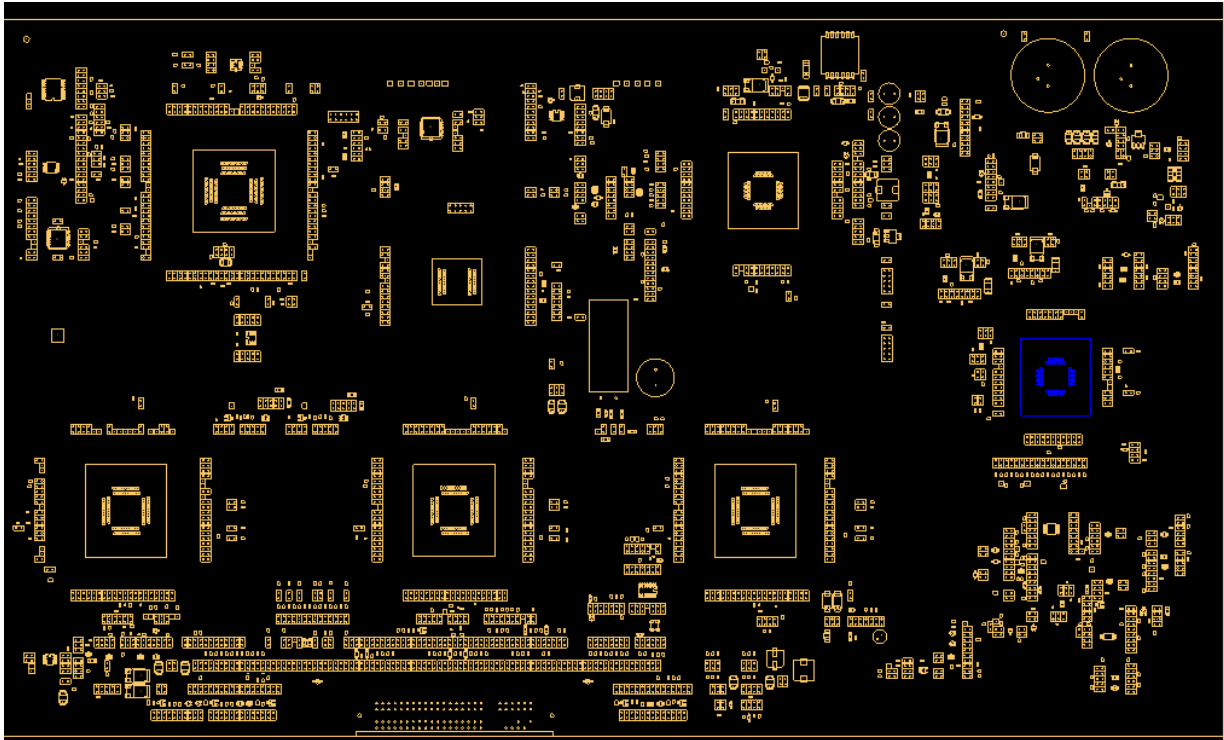
*Fig. 34 Airbag ARAMIS 2 (unidad de prueba) en PCBCAM*

Realizamos una búsqueda del componente, mediante el cuadro de dialogo **Select Component** en el PCBCAM (Fig. 35a)



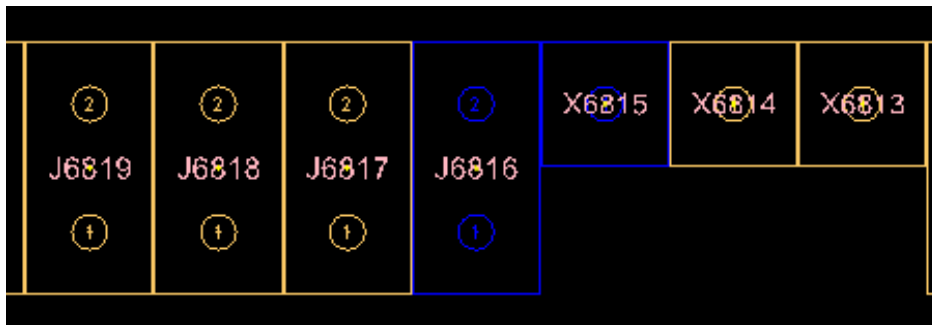
*Fig. 35a búsqueda del componente V9802*

Cuando damos clic en el botón **Go** el programa muestra en color azul el componente en cuestión. (Véase Fig. 35b)



*Fig. 35b V9802 seleccionado*

Ahora para localizar los pines 15 y 16, hacemos un zoom sobre el componente y recurrimos de nuevo al cuadro de dialogo **Select Component** para localizar los Jumpers X6815 y J6816. (Ver Fig. 36)



*Fig. 36 Jumpers J6816 y X6815*

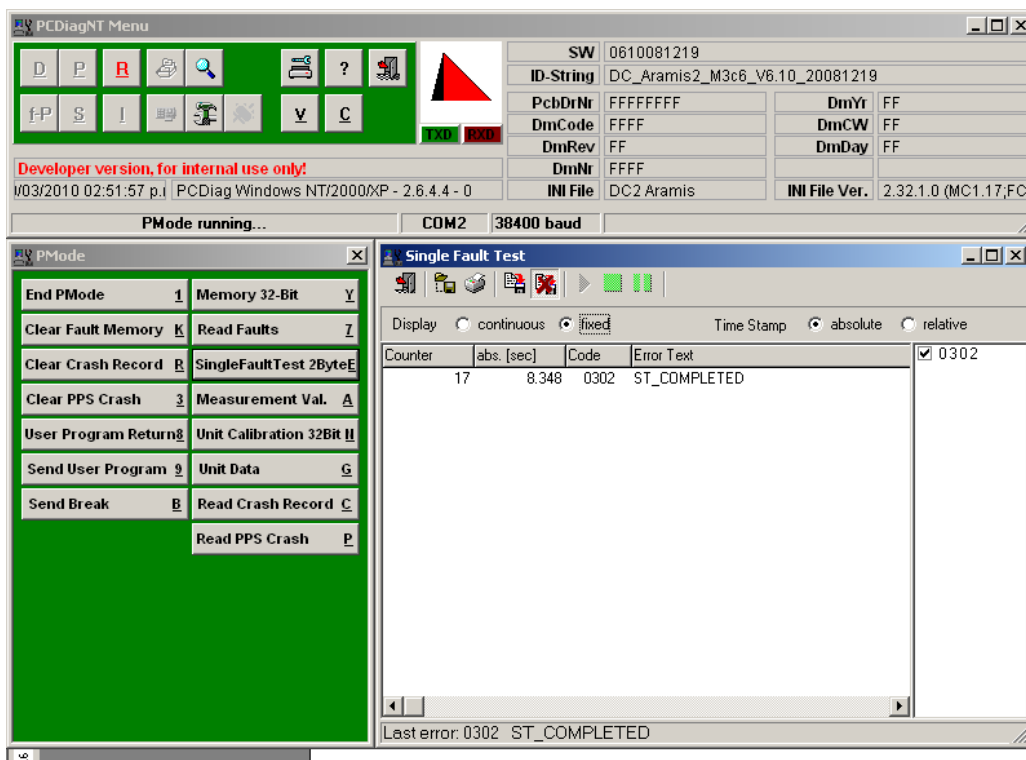
**Paso 4:** Después de tener ubicado los pines, se prosigue a realizar la falla y diagnosticarla. Existen dos modos de diagnóstico que la unidad realiza, y para los cuales hay que ejecutar la falla y diagnosticar posibles errores. Los modos son **Startup** y **Cyclic**. El modo Startup se ejecuta al arrancar el sistema, hace un diagnóstico inicial

de algunas características del airbag y no se vuelve a ejecutar hasta el próximo arranque. Por esta razón las fallas que se realizan en modo startup deberán ejecutarse antes de arrancar el sistema y analizarse una vez arrancado. Por otro lado, el modo Cyclic se comienza a ejecutar después de que termina el modo startup y de forma cíclica hasta detener el sistema. Así que las fallas en Cyclic pueden ejecutarse cuando el sistema haya arrancado y diagnosticarse mientras el sistema no se haya detenido.

Por esta razón una prueba específica se realiza dos veces, incrementado el número de pruebas al doble, con el fin de diagnosticar si se detecto algún error en modo startup y para diagnosticar también si se detectó algún error en modo Cyclic.

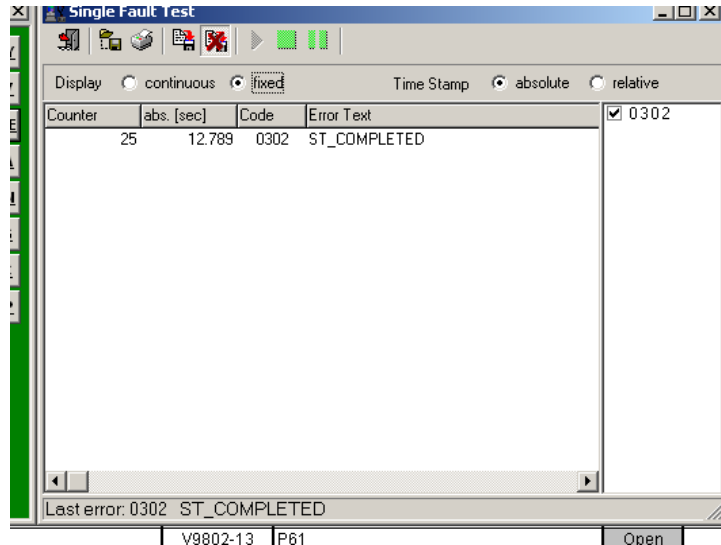
Continuando con nuestro ejemplo, vamos a realizar la prueba en modo Cyclic primeramente. Para esto son necesarias dos cosas:

- Encender el módulo de simulación (Se entiende que tenemos ya conectada la unidad de Airbag al módulo)
- Abrimos el software de diagnóstico PCDiagNT y checamos que no haya ningún error en la pantalla principal. El software debe arrojar algo como lo que se muestra en la fig. 37. El único código que debe mostrar es el 0302, que indica que el modo de Startup se completo sin errores (ST\_Completed).



*Fig. 37 PCDiagNT, software de diagnóstico*

Ahora bien, como la prueba es en modo Cyclic podemos ejecutar la falla y diagnosticar los posibles errores cuando la unidad permanece prendida. Para nuestro ejemplo he puesto un corto circuito mediante cables en los pines 15 y 16 del V9802, y la figura 38 muestra el diagnóstico correspondiente.



**Fig. 38 Diagnóstico de la falla short 15,16-V9802**

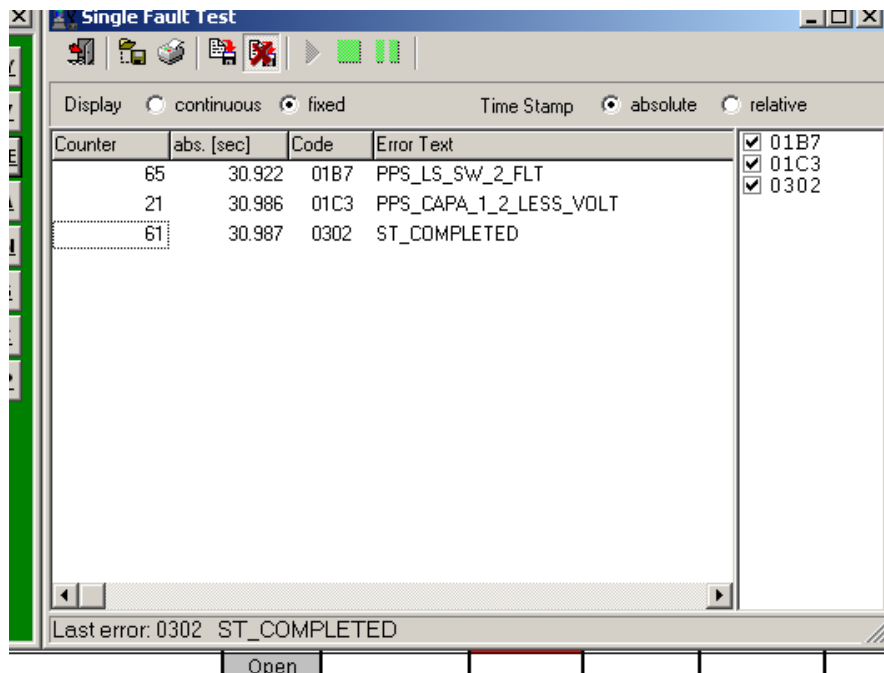
Como se puede observar no se diagnóstico ninguna falla, así que debemos volver al banco de pruebas y llenar los campos que se requieren. En la figura 39 vemos que el campo AWL se puso OFF ya que es el valor que arrojó la lámpara a la hora de realizar la prueba, el campo Failure Recognition Cycle test tiene un "-" ya que no se obtuvo ningún error, los campos siguientes se llenaron con datos de la versión de software y el último se llenó con el nombre de quien realizó la prueba.

AWL	PAD LAMP	Failure Recognition Cycle test	Firing possible?		Appreciation	Track							
			Yes	No		MuC SW Version	SuC SW Version	PuC SW Version	Par file	HW Version	Test Date	Done By	Ch
OFF		-				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		EDMA	
OFF		-				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		OSSES	
OFF		01B9				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		OSSES	

**Fig. 39 Resultados de la prueba realizada**

Por último vamos a realizar la misma prueba pero en modo Startup. Así que procedemos a apagar la unidad y ejecutar la falla. Enseguida encendemos la unidad y observamos el diagnóstico. (Véase Fig. 40)

Como podemos notar, se arrojaron 2 errores, el primero con código 01B7 que se había ejecutado 65 veces hasta el segundo 30 y el siguiente que se había ejecutado 21 veces en 30 segundos, también se pudo observar físicamente que la lámpara encendió cuando se provocó este error. Estos resultados hay que anotarlos en el banco de pruebas (Figura 41). Con esto habremos terminado una prueba y hay que repetir el proceso para las pruebas que faltan.



*Fig. 40 Diagnóstico de la prueba short 15,16 V9802 en Startup*

AWL	PAD LAMP	Failure Recognition Startup test	Firing possible?		Appreciation	Track						
			Yes	No		MuC SW Version	SuC SW Version	PuC SW Version	Par file	IHW Version	Test Date	Done By
OFF		-				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		EDMA
ON		01B7, 01C3				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		OSES
OFF		012D				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		OSES
OFF		01B9				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		OSES
OFF		01BE				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		OSES
OFF		01C1				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		OSES
OFF		-				Aramis2	Aramis2	Aramis2	MCG_5W	07 / 49		OSES

*Fig. 41 Resultados de la prueba short 15,16 V9802 en Startup*

En mi caso realice aproximadamente 752 pruebas en 2 semanas, debido a que las demás ya habían sido completadas por otras personas del departamento. Aunque el trabajo continuó con la elaboración del reporte de los resultados el cual menciono adelante, entre otras actividades.

#### 4.4 Elaboración del reporte del resultado de las pruebas

Las pruebas que se realizaron son útiles principalmente para el departamento de producción quien recibe el diseño ya depurado de la unidad de Airbag y diseña su proceso de producción el cual incluye algunas pruebas funcionales. Para nuestro caso se aplicarán las siguientes pruebas de producción:

- **AOI (Automatic Optical Inspection).** Esta prueba consiste en realizar un escaneo a través de cámaras sobre las tarjetas que han sido reproducidas. Principalmente se analiza que no haya faltado montar ningún componente, que no haya pistas rotas y que no haya soldaduras mal hechas. Esta prueba se hace automáticamente con un equipo especializado, sin embargo hay componentes que no pueden ser detectados en esta inspección. Existe un documento que elabora el encargado de cada proyecto en donde especifica que componentes del diseño en cuestión pueden ser detectados en esta prueba, los componentes que no pueden ser detectados son omitidos de dicho documento.

Para la elaboración de nuestro reporte vamos a usar este documento para corroborar que los componentes que se usan en el diseño del airbag realmente se pueden detectar, en caso contrario, deberá reflejarse en el reporte a realizar.

- **ICT (In Circuit Test).** Esta es la segunda prueba de producción a la cual se someterán las unidades reproducidas de airbags. Consiste en realizar pruebas eléctricas como por ejemplo: medición de valores nominales de resistencias, capacitores, bobinas, etc. También se realiza un reset a todos los microcontroladores. Y se carga el código del Microcontrolador APACE y algunos datos en la memoria EPROM
- **DTA (Diferencial Termic Analysis).** Esta prueba consiste en checar el desempeño de las unidades reproducidas, mediante la evaluación de algunos parámetros de los microcontroladores, estabilidad de voltajes y corrientes, estado de los sensores, etc., pero bajo una temperatura baja relativamente que está en el rango de  $-40^{\circ}\text{C} \pm 3^{\circ}\text{C}$  y después de esto se somete a una temperatura elevada que está contemplada en  $85^{\circ}\text{C} \pm 3^{\circ}\text{C}$ . En esta prueba se pueden detectar errores que se registren en la memoria de fallas (EPROM) o en el análisis que la unidad realiza cíclicamente (Single Fault Test, **SFT**).
- **SHAKER.** Esta prueba consiste en poner a prueba el funcionamiento de las unidades bajo condiciones de movimientos bruscos. Se hace uso de una fixtura para generar los movimientos de tal forma que se simulen los movimientos que se generan en un choque o volcadura. Se prueban principalmente los sensores Pitch-Over, Rollover, sensores de velocidad, entre otros. A diferencia del DTA en



esta prueba solo se pueden detectar fallas que se hayan registrado en la memoria de fallas (EPROM).

- **ROLLOVER.** Esta prueba es muy parecida a la del Shaker ya que se simulan choques para poner a prueba el sistema. Sin embargo, en este caso se prueba:
- la funcionalidad del sistema en su totalidad y se analizan tiempos de respuesta, retardos, etc. y se hacen los ajustes pertinentes. En esta prueba no se detecta ningún error ya que se trata más bien de realizar las calibraciones necesarias en cuanto a tiempo.
- **FINAL TEST.** Es la prueba final, principalmente se realiza la carga de los códigos de todos los microcontroladores, se hace un clean up de la memoria EPROM y se le cargan datos como: No. de Serie, Configuración del sistema, etc. En esta fase final se pueden detectar errores que se registren tanto en la memoria de fallas como en el SFT.

Estas pruebas de producción pueden o no realizarse dependiendo del diseño que se esté produciendo. Esto se decide en base a algunos parámetros obtenidos en las pruebas de cortos y abiertos y en algunos otros documentos. Para hacer notar esto vamos a ver un ejemplo de cómo se realizó el reporte de la unidad ARAMIS 2.

El reporte del resultado de las pruebas de cortos y abiertos cuenta con los siguientes campos de información que deben completarse:

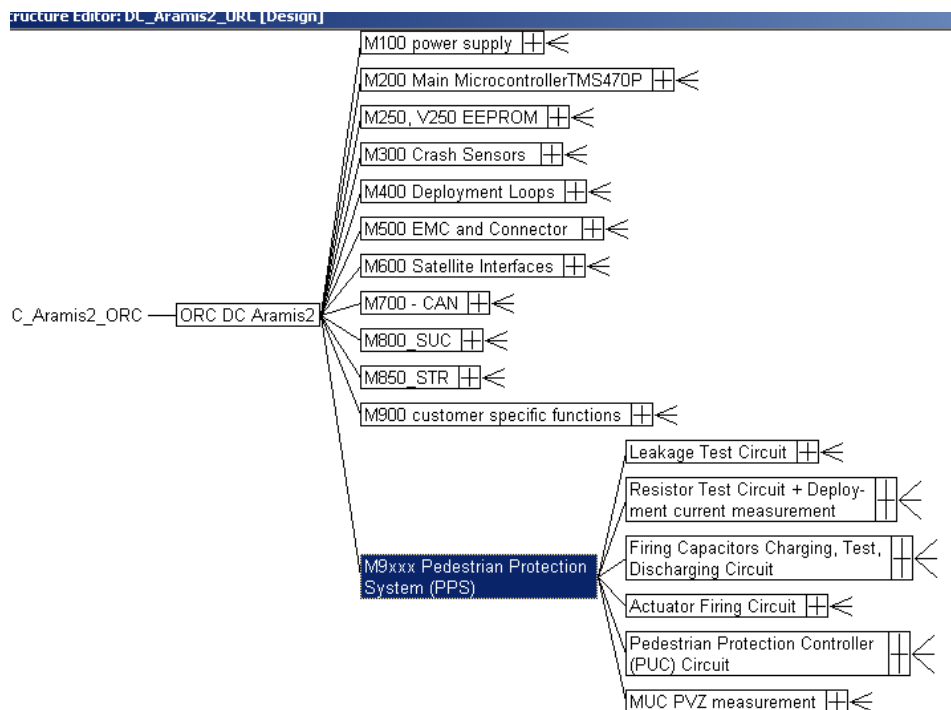
- **No. de Prueba**
- **Nombre y No. de Pin** del componente a reportar
- **Falla que se aplico**
- **AOI.** Se especifica si se puede aplicar esta prueba al componente.
- **ICT.** Se especifica si se puede aplicar esta prueba al componente.
- **DTA.** Se especifica si se puede aplicar esta prueba al componente.
- **Shaker.** Se especifica si se puede aplicar esta prueba al componente.
- **Rollover.** Se especifica si se puede aplicar esta prueba al componente.
- **Final Test.** Se especifica si se puede aplicar esta prueba al componente.
- **Severity.** Es un número indicativo que esta en el rango de 1-10 y que indica que tan severa puede ser una falla.
- **Ocurrence.** Al igual que Severity está en el rango de 1-10 e indica la posibilidad que ocurra.
- **Detectability.** Se encuentra en el rango de 1-10 e indica que tan posible es detectar la falla.

- **RPN.** Este es un número que se obtiene multiplicando los tres números anteriores (Severity, Ocurrence, Detectability) y se puede encontrar en un rango de 1 a 1000. Este número no debe tener valores altos porque entonces estaría dando a entender que la falla es muy crítica y que es necesaria una corrección para eliminarla.

| Existe un software conocido como **IQ-RM** el cual se usa para clasificar de forma ordenada y jerárquica todos los componentes que incluyen un diseño en particular, mediante este software podemos obtener dato sobre la unidad como son los valores de Severity, Ocurrence y Detectability. Este software está basado en un método organizacional conocido como FMEA (Failure Mode Effect Analisys) que tiene como objetivos:

- Facilitar la creación y actualización de nuevos proyectos
- Contribuir a la homogeneidad de la calidad, comparando y estandarizando los proyectos
- Facilitar el manejo de los fallos individualmente
- Clasificar los errores por severidad, Detectabilidad y Ocurrencia
- Entre otros.

En la figura 42, se muestra la forma en que se maneja cada proyecto.



**Fig. 42 El FMA permite manejar los proyectos como un organigrama**

Vamos a comenzar con nuestro ejemplo del llenado del reporte. En la Fig. 43 se muestra el formato que debemos completar.

ARAMIS 3 FAULT COVERAGE ANALYSIS														
MODULE 100 POWER SUPPLY														
No. Test	Component	Failure	Function	BOB VOS COMPONENT	AOI (PTS)	ICT (PTS)	DTA	Shaker	Rollover	Final Test	DFMEA SEVERITY	DFMEA OCCURENCE	DFMEA DETECTABILITY	REP
Module 100														
1	C110	open			X									
2	C110	short		C110										
3	C115	open												
4	C115	short		C115										
5	C116	open												
6	C116	short		C116										
7	C1161	open												
8	C1161	short												

*Fig. 43 Formato de reporte final*

**Paso 1:** Como primer paso vamos a elegir para este ejemplo el capacitor C110. Debemos completar la información para las dos fallas que se realizaron sobre este componente que son open y short.

**Paso 2:** El primer campo se el AOI en el cual se debe indicar si el C110 se puede detectar en esta prueba. Para esto tenemos que consultar el documento donde se registran todos los componentes que pueden ser detectados en esta prueba. En este caso el C110 si se encuentra en la lista, así que marcamos la celda para indicar que si se puede detectar. (Véase Fig. 43)

**Paso 3:** El campo de ICT se llena en base al siguiente criterio: Si el componente fue detectado en AOI puede se omitido para la prueba de ICT, pero los componentes que no pudieron ser detectados tienen que ser forzosamente analizados en ICT. Si el componente tiene una funcionalidad importante deberá pasar por las dos pruebas (por ejemplo los microcontroladores). En base a este criterio podemos marcar al capacitor C110 con un X, ya que si pudo ser detectado en AOI y su funcionalidad no se considera severamente afectada. (Véase Fig. 43)

**Paso 4:** El campo de DTA puede ser completado en base a los posibles errores que se hayan encontrado en las pruebas realizadas anteriormente. Como las pruebas se realizan tanto en corto como en abierto existen dos resultados de DTA para el C110.

A partir de este paso se usa la siguiente regla: Si la prueba en Startup muestra alguna falla esta se registra en la memoria de fallas (EPROM) y si es en Cyclic se registrará en SFT.

Si se revisa el banco de pruebas y en específico los resultados del C110 nos daríamos cuenta que no tuvo errores en modo startup ni en modo Cyclic para la falla open, sin embargo para la falla short se encontraron errores en ambos modos. El campo DTA queda de la forma que se muestra en la figura 44.

**Paso 5:** El campo Shaker se llena de forma parecida al campo DTA, es decir, en base a los errores obtenidos en ambos modos y en ambas fallas para el C110, además considerando que en la prueba del Shaker solo se detectan errores que se registren en la memoria de fallas y tomando en cuenta la regla mencionada en el paso 4, podemos decidir los valores que se muestran en la Fig. 44.

**Paso 6:** Se mencionó que para la prueba del roll-over no se pueden detectar ningún error ya sea que este se registre la memoria de fallas o en SFT. Sin embargo en este caso se marcó con una ✓ ya que en este componente está relacionado con la alimentación de la batería y afecta el funcionamiento de la unidad considerablemente. Los resultados para este campo se pueden ver en Fig. 44.

**Paso 7:** En la prueba final se siguen considerando los mismos criterios, así que no se podrá detectar nada para el open del C110 ya que no se presentaron fallas, sin embargo, si se podrán detectar errores para el short del C110 debido a que presento errores en ambos modos (Starup y Cyclic).

**Paso 8:** Para los campos **Severity, Ocurrence y Detectability** es necesario consultar los resultados que arroje el IQ-RM. Estos valores fueron asignados anteriormente por los diseñadores del circuito. Nosotros solo vamos a tomarlos y anotarlos en el reporte. Los valores para el C110 se observan en la Fig. 44.

**Paso 9:** El último campo RPN es la multiplicación de los 3 valores obtenidos en el paso 8. No es necesario calcularlo, en vez de eso se usa la función de Excel para calcular operaciones matemáticas automáticamente entre celdas.

Este es el procedimiento que se siguió para realizar el reporte en su totalidad que constó aproximadamente de 3355 pruebas. Es un proceso tedioso y repetitivo pero que es necesario para el departamento de producción.

3 FAULT COVERAGE ANALYSIS													
00 POWER SUPPLY													
Component	Failure	Function	BOH VOS COMPONENT	AOI (PIS)	ICT (PIS)	DTA	Shaker	Rellover	Final Test	DFMEA SEVERITY	DFMEA OCCURRENCE	DFMEA DETECTABILITY	REM
C110	open						X	X	X	3	1	4	32
	short		C110	✓	X	✓	✓	✓	✓	3	2	2	36
C115	open												

*Fig. 44 Resultados para el C110 del reporte final de resultado de pruebas*

## 4.5 Detección, Análisis y Corrección de fallas

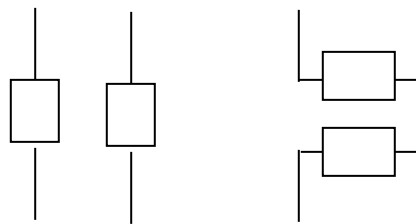
Otra de las actividades que se realizaron, se trató de revisar una tarjeta FEA que se encontraba en malas condiciones, así que, se procedió a realizar una inspección y se encontraron los siguientes errores:

- Cortocircuito entre VPERI y Ground.
- Cortocircuito entre VPERIREF y Ground
- Cortocircuito entre EVZ y Ground.

Estos cortos circuitos estaban causando que la unidad no pudiera establecer comunicación con la PC. Así que se procedió a buscar la causa de los cortos y corregirlos.

Para el primer cortocircuito se encontró una pista dañada mediante una inspección bajo microscopio, así que se mandó a los laboratorios a reparar y con esto desapareció la falla. El segundo cortocircuito se corrigió también al reparar la pista dañada, esto se debe a que VPERIREF depende de VPERI, en realidad son el mismo voltaje solo que VPERIREF es el resultado de filtrar la señal de VPERI.

El segundo error fue muy complicado de corregir, tomó mucho tiempo en localizar la falla. Lo que sucedió es que el técnico quien se encargó de montar los componentes en la placa se equivocó al montar dos resistencias superficiales (R9859 y R9860), estas se localizan en el sub-bloque de PVZ Discharging que se analizó en el punto 3.5.21. El error del técnico fue conectar las resistencias como se muestra en la Fig. 45 (b) cuando lo correcto era como en (a).



**Fig. 45 (a) Forma Correcta (b) forma incorrecta**

Al corregir esta falla se procedió a programar los microcontroladores de la unidad (este proceso se detalla en la siguiente sección) para tratar de establecer la comunicación con la PC. El objetivo era dejar funcionando la unidad sin ningún error o falla, sin embargo, aunque se pudo establecer comunicación con la PC esta vez,

todavía persistían algunos errores que estaban siendo detectados en el diagnóstico por computadora, pero este mismo diagnóstico arroja una clave la cual nos habla de la causa del posible error, esto definitivamente facilita la tarea.

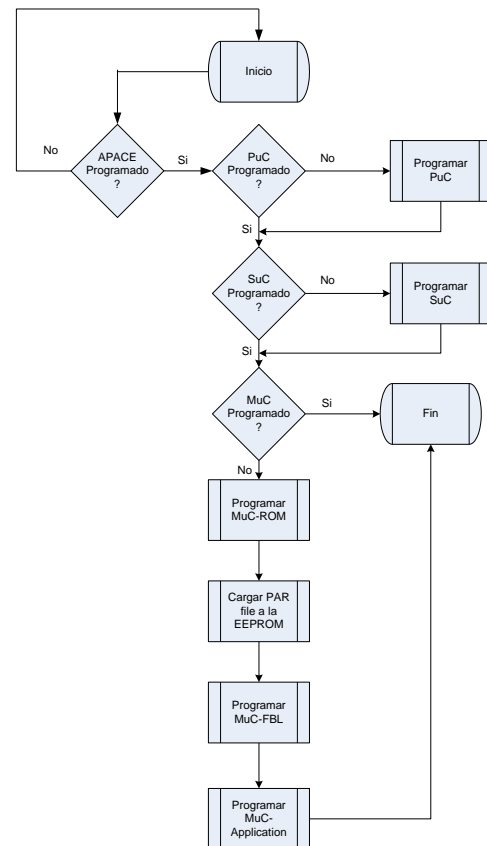
Se consiguió el objetivo en aproximadamente 7 días, siendo los cambios físicos que se hicieron sobre la unidad (como la reparación de la pista dañada y el cambio de las resistencias) los procesos más lentos ya que los técnicos del laboratorio siempre tienen mucha carga de trabajo.

## 4.6 Programación de microcontroladores de unidad ARAMIS II

Una de las tareas que se deben realizar sobre una unidad ARAMIS II consiste en programar los microcontroladores que esta unidad tiene para que tenga un correcto funcionamiento. La programación no consiste en escribir el diseñar el código para cada micro (esto es tarea del departamento de Software), sino en tomar esos código y almacenarlos en las memorias de cada microcontrolador. Enseguida se muestra el procedimiento que se siguió para esta actividad.

### 4.6.1 Diagrama de Flujo de programación

En la Fig. 46 muestra el flujo de programación para los microcontroladores de ARAMIS II (MuC, PuC y SuC):

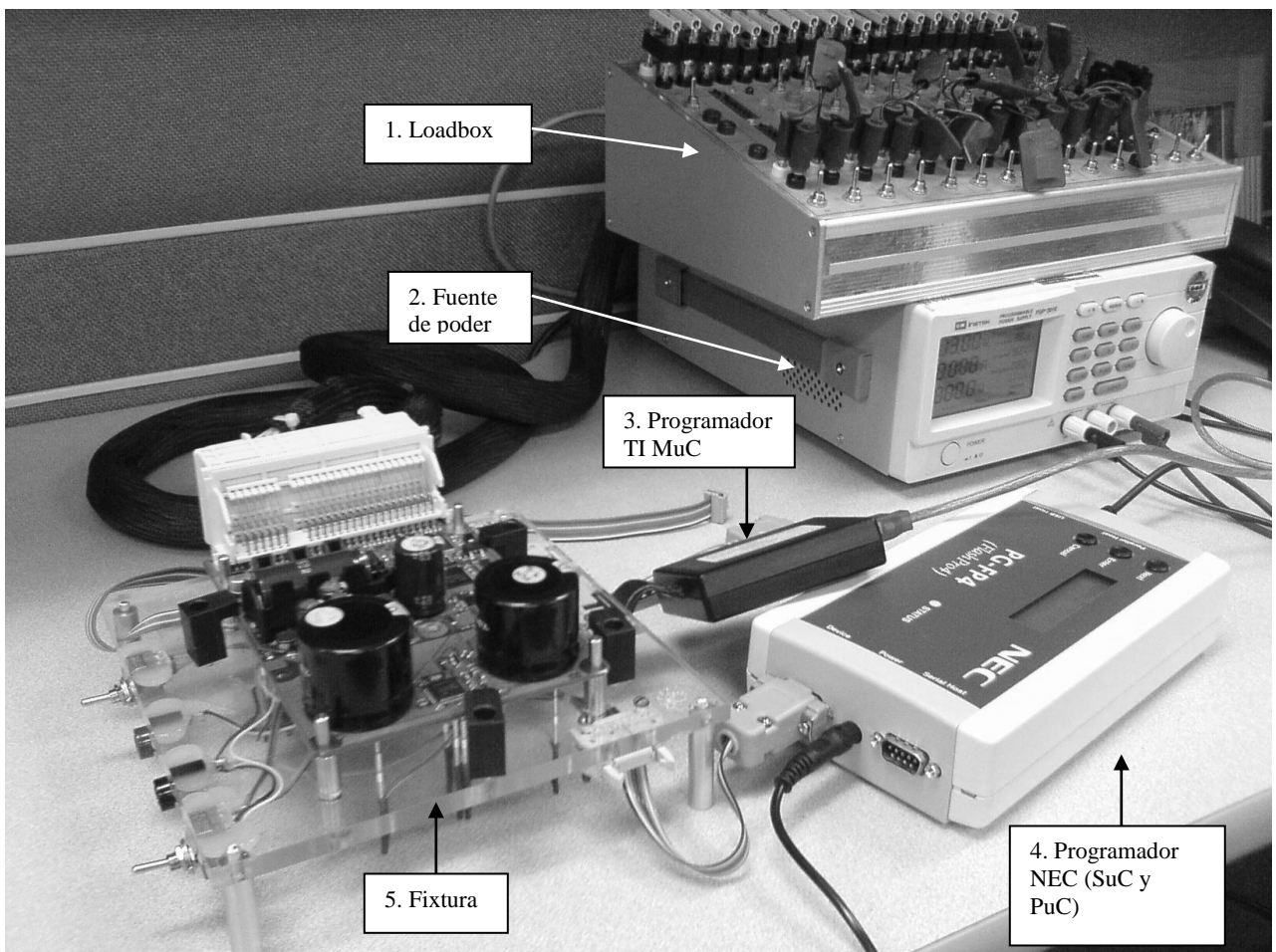


*Fig. 46 Diagrama de Flujo*

#### 4.6.2 Hardware Utilizado

- **Loadbox:** Módulo de simulación donde se encuentran montados sensores, actuadores, indicadores y permite también establecer comunicación con la PC.
- **Fuente de Poder:** Es la fuente de alimentación del loadbox
- **Programador TIMuc:** Programador para el MUC
- **Programador NEC:** Este se utiliza para programar el SUC y PUC

En la Fig. 47 se observa una unidad Aramis II montada sobre una fixtura, en nuestro caso la unidad que usamos es una versión expandida (como se mencionó anterior mente) así que no es necesaria la fixtura ya que esta versión cuenta con conectores para los programadores.

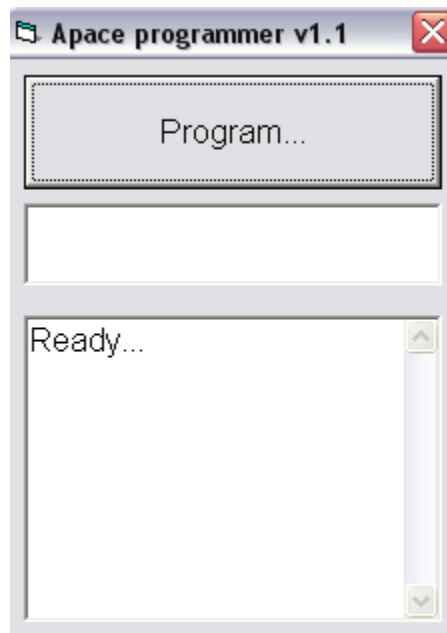


*Fig. 47 Equipo utilizado para programar los Microcontroladores*



### 4.6.3 Programación del APACE

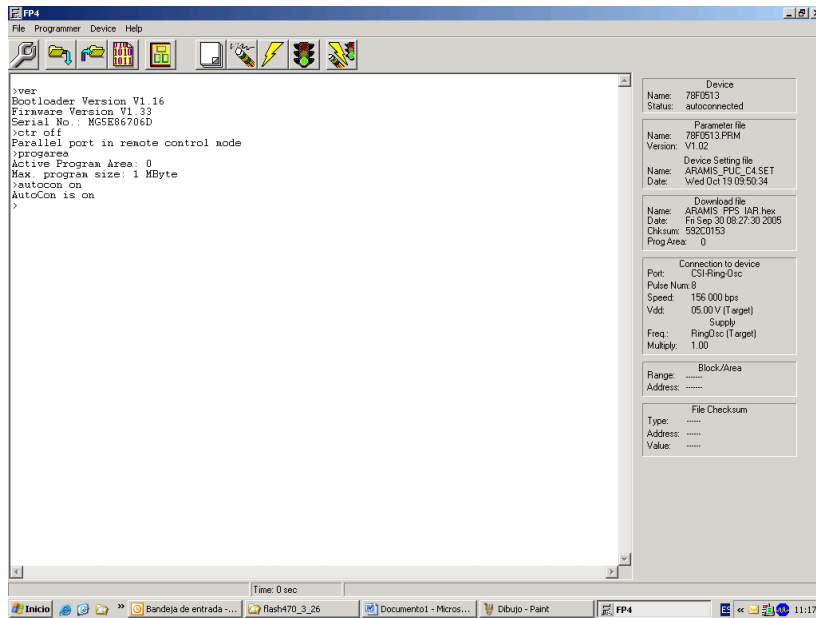
Para la programación del APACE, se necesita conectar el programador al conector de la FEA y tener conectada la unidad al loadbox. Para establecer comunicación con el Apace y cargarle el código se hizo uso de un software llamado Apaceprogrammer (Fig. 48) en la ventana de este software se ingresa la palabra reservada A9 29 y luego se da clic en el botón program. Lo que pasa es que al recibir ese código el apaceprogrammer envía hacia el Apace una serie de comandos SPI los cuales contienen la información del código.



*Fig. 48 Ventana principal del Apaceprogrammer*

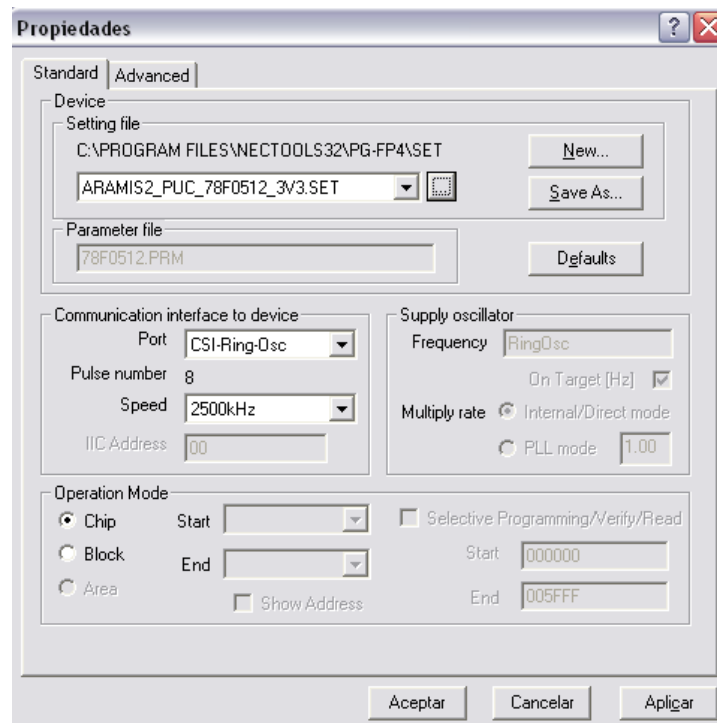
### 4.6.4 Programación del microcontrolador PUC

Para este microcontrolador usamos el programador NEC (Fig. 43) conectado al conector del PUC que está sobre la unidad. Es necesario también tener conectada la unidad a la loadbox. En este caso se hace uso del software FP4. (44a)



**Fig. 44 (a) Ventana principal del FP4**

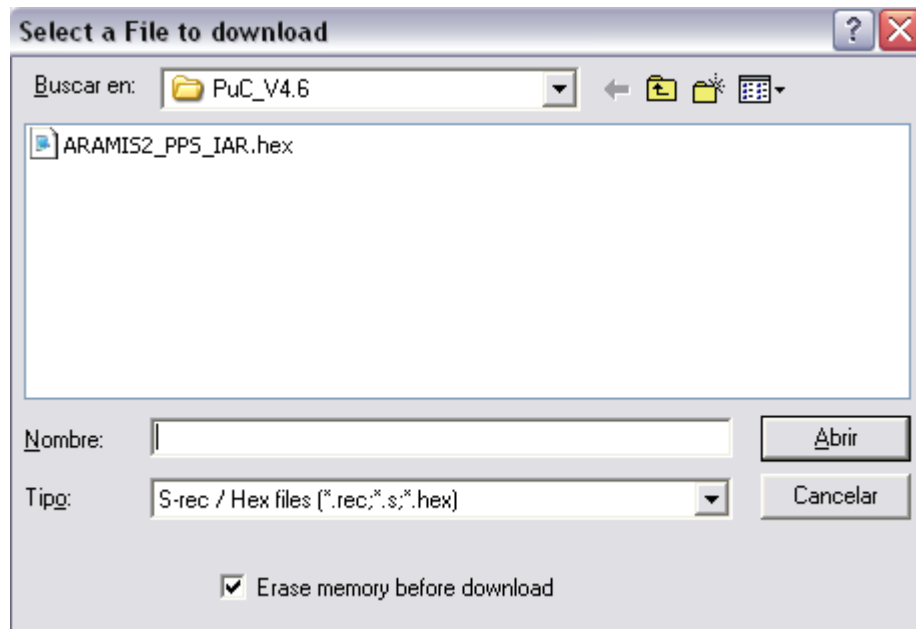
El primer paso es elegir el dispositivo que se va a programar y configurarlo. Para esto debemos dar clic en el menú Device y elegir la opción setup, entonces nos aparecerá la ventana de propiedades. (Fig. 44 b)



**Fig. 44 (b) Ventana de propiedades del FP4**

En esta ventana podemos ver en el área de Setting File, que hemos elegido el archivo ARAMIS2\_PUC\_78F0512\_3V3.SET el cual contiene la configuración del dispositivo que se desea programar y que es útil para el software.

Ahora debemos regresar a la ventana principal y abrir el menú File y dar clic en la opción download. Entonces elegimos el archivo .hex que contiene el código que se debe cargar en el PUC. (Fig. 44c)



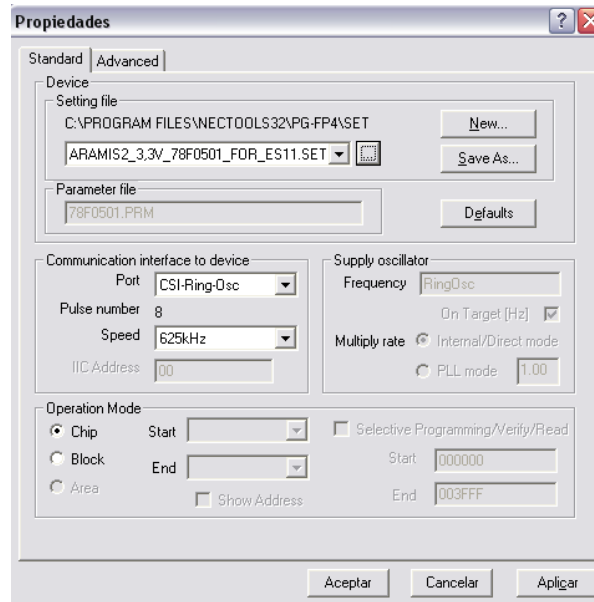
*Fig. 44 (c) Selección de archivo .hex para el PUC*

Por último volvemos a la ventana principal y damos clic en el botón **"Blank check, erase, program and verify a device"** 

El software muestra al final de la operación una leyenda que indica el 100% completado.

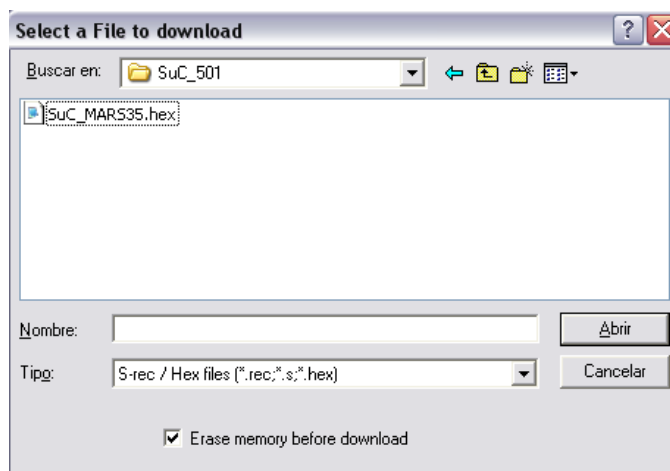
#### 4.6.5 Programación del microcontrolador SUC

La programación de este microcontrolador es muy parecida a la del PUC con la diferencia que se cargan archivos de configuración y código hexadecimal diferentes. Por ejemplo, el archivo de configuración utilizado en este caso es ARAMIS2\_3,3V\_78F0501\_FOR\_ES11.SET (ver figura 45a)



*Fig. 45 (a) Ventana de Configuración para el SUC*

El archivo .hex que se cargó en el SUC es el Suc\_MARS35.hex (Fig. 46 b)

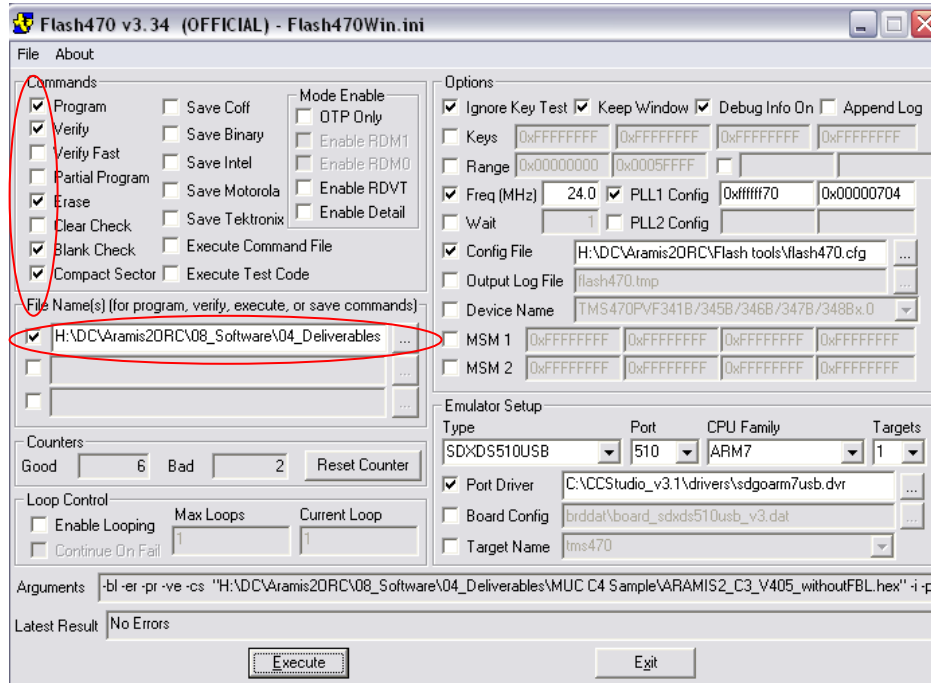


*Fig. 45 (b) Selección de archivo .hex para el SUC*

## 4.6.6 Programación del microcontrolador MUC

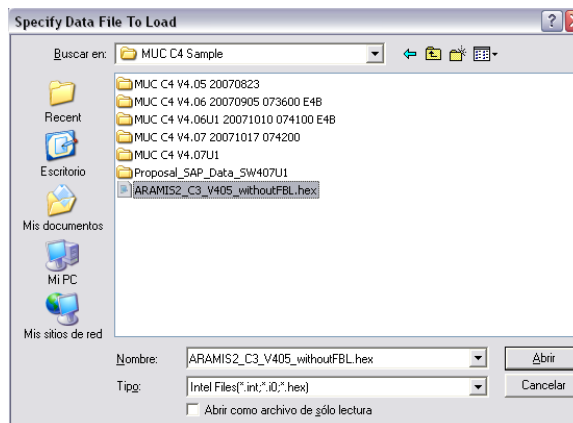
### 4.6.6.1 Programación de MUC\_ROM

Para poder realizar el flashing de la memoria ROM del MUC vamos a necesitar de un software llamado Flash470. (Ver Fig. 46 a)



*Fig. 46 (a) Ventana principal del Flash470*

Debemos verificar que las casillas marcadas con un círculo rojo en la Fig., anterior estén seleccionadas y ahora debemos cargar el archivo .hex para la rom, como se muestra en la fig. 46 b.



*Fig. 46 (b) Selección de archivo .hex para la rom*

A continuación damos clic en el botón execute para iniciar la programación y al final una ventana nos aparecerá indicando que no hubo errores. (Fig. 46 c)

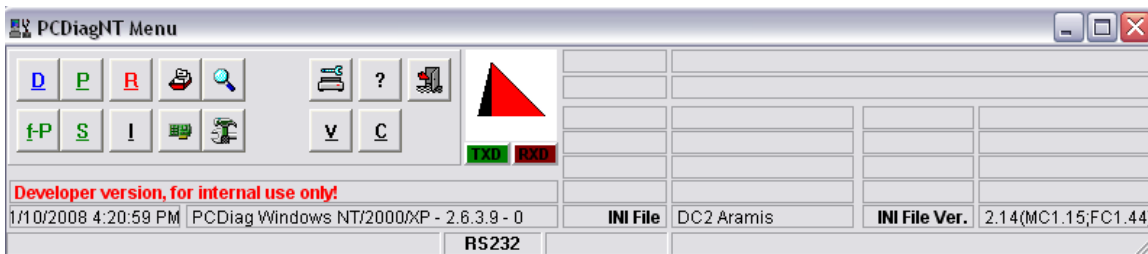
```

c:\ti\flash470\v334\Fflash470.exe

Compacting Sector from 0x0003C000 to 0x0003FFFF
 0 total compaction pulses
Compacting Sector from 0x00040000 to 0x00047FFF
 0 total compaction pulses
Compacting Sector from 0x00048000 to 0x0004FFFF
 0 total compaction pulses
Compacting Sector from 0x00050000 to 0x00057FFF
 0 total compaction pulses
Compacting Sector from 0x00058000 to 0x0005FFFF
 0 total compaction pulses
Checking blank from 0x00000000 to 0x0003FFFF
Checking blank from 0x00040000 to 0x0005FFFF
Programming from 0x00000000 to 0x000000FF
 Avg pulses 1.00, Max pulses 1
Programming from 0x00000120 to 0x0000811F
 Avg pulses 1.00, Max pulses 3
Programming from 0x00008144 to 0x0003FFFF
 Avg pulses 1.03, Max pulses 3
Verify from 0x00000000 to 0x000000FF
Verify from 0x00000120 to 0x0000811F
Verify from 0x00008144 to 0x0003FFFF
Elapsed execution time 15.80 seconds.
Total elapsed time 18.38 sec
No Errors
Press Enter to exit
  
```

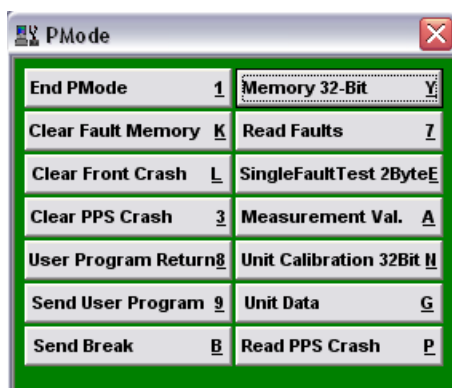
*Fig. 46 (c) Ventana que indica el 100% completado del flashing*

Ahora vamos a configurar la EEPROM usando el PCDIAG (Fig. 46 d)



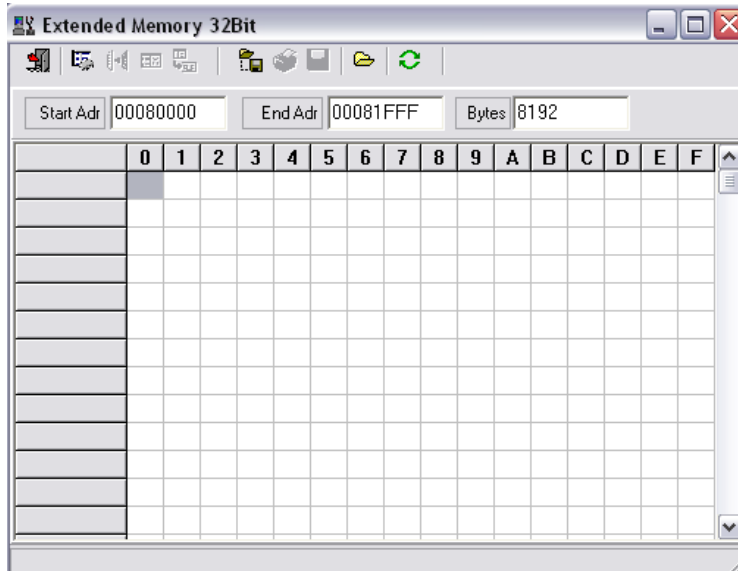
*Fig. 46 (d) Ventana principal del PCDIAG*

Damos clic en la letra P para acceder al Modo P-Mode. En la ventana que aparecerá (Fig. 46 E) elegimos el botón Memory 32-Bit



*Fig. 46 (e) Ventana de P-Mode*

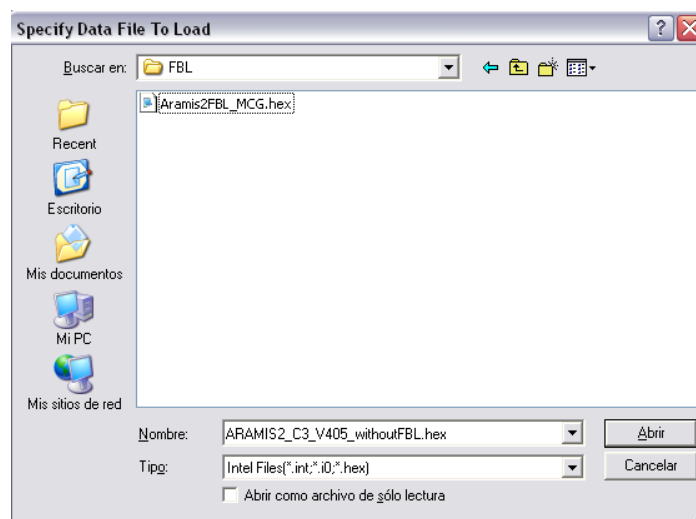
La ventana mostrada en la Fig. 46 F aparecerá, en está elegimos el archivo .txt el cual contiene la configuración de la EEPROM damos clic en el botón Write y con esto se finaliza el proceso de configuración.



**Fig. 46 (F) Mapa de memoria de la ROM**

### 3.6.6.2. Programación de MUC-FBL

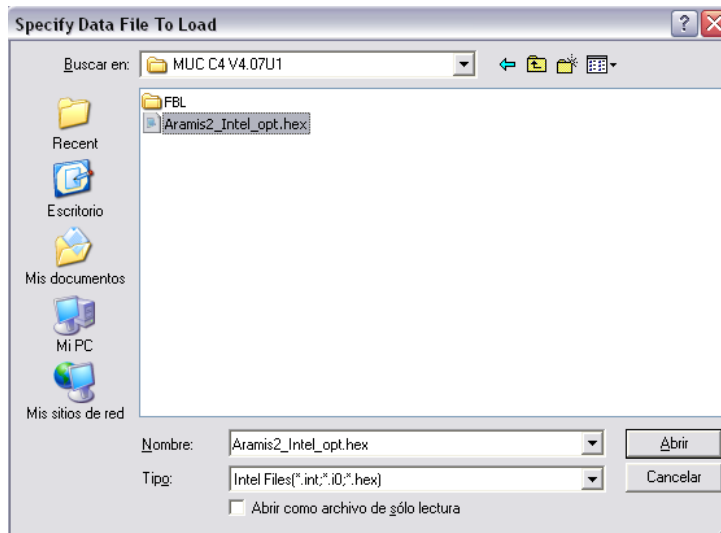
Este proceso es muy parecido al anterior. Se hace uso de Flash470 también, lo único que varía es que ahora debemos cargar un archivo .hex diferente, el cual se muestra en la Fig. 47



**Fig. 47 Selección de archivo para el MUC-FBL**

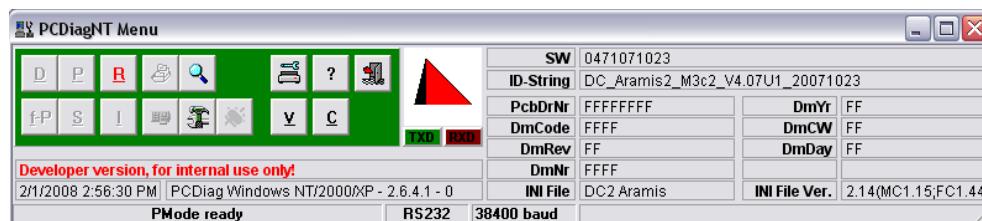
### 3.6.6.3 Programación de MUC-Application

Este es el último paso que se requiere para tener toda la unidad programada. Seguimos usando Flash470 pero ahora vamos a cargar un archivo diferente. (Ver Fig. 48 A).



*Fig. 48 (a) Selección de archivo para el MUC-Application*

Después de haber escrito el archivo en el microcontrolador vamos a verificar que todo se haya realizado bien. Para esto vamos a usar nuevamente PCDIAG. Vamos a entrar a P-Mode otra vez y entonces debemos ver una ventana como la que se muestra en la Fig. 48 B. Debemos verificar que la leyenda ID-String coincida con la de la imagen.



*Fig. 3.6.3 (b) Comprobación del flashing*



## 5. CONCLUSIONES

La realización de este trabajo se concluyó satisfactoriamente y podemos hablar de que fue un éxito ya que actualmente se está realizando una producción semanal de 3,000 unidades ARAMIS II en las líneas de producción de la planta conocida como la Tijera en Guadalajara, Jalisco. Producción que se pretende mantener por los próximos 2 años ya que así ha sido requerida por Mercedes Benz.

Este proyecto me ha permitido poner en práctica mis conocimientos adquiridos durante mi proceso de preparación profesional en el Instituto Tecnológico de Tuxtla Gutiérrez, a través de las materias que se me fueron impartidas. Es por esto que la culminación de este trabajo representa también un gran éxito profesional personal ya que además de permitirme acreditar mi Residencia Profesional ahora también cuento con experiencia laboral en este campo tan interesante como es el ramo electrónico automotriz.

## 7. BIBLIOGRAFÍA

### **Child Anthropometry for Improved Vehicle Occupant Safety**

Matthew Reed

Kathleen DeSantis Klinich

### **Automotive Safety Handbook, Second Edition**

Lothar Wech

Ulrich W. Seiffert

### **Rear Impact, Side Impact, and Rollover, 2010**

HandBook

### **Advances in Side Airbag Systems**

Donald E. Struble

SAE International

<http://auto.howstuffworks.com>

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

[airbagindustries.com](http://airbagindustries.com)

[www.airbagcenter.com](http://www.airbagcenter.com)