

**DISEÑO E IMPLEMENTACION DE SISTEMA DE DIAGNOSTICO PORTABLE  
PARA AACP Y MOUNTGRIP EN AERONAVES**

**EDNA CAROLINA MORIONES POLANÍA  
OSCAR LEONARDO MOSQUERA DUSSÁN**

**UNIVERSIDAD SURCOLOMBIANA  
FACULTAD DE INGENIERIA  
INGENIERIA ELECTRÓNICA**

**NEIVA**

**2010**

**DISEÑO E IMPLEMENTACION DE SISTEMA DE DIAGNOSTICO PORTABLE  
PARA AACP Y MOUNTGRIP EN AERONAVES DE COMBATE**

**EDNA CAROLINA MORIONES POLANÍA  
OSCAR LEONARDO MOSQUERA DUSSÁN**

**Trabajo de grado para optar el título de Ingeniero Electrónico**

**DIRECTOR**

**JAVIER HUMBERTO RUBIO**

**Ingeniero Electrónico**

**UNIVERSIDAD SURCOLOMBIANA**

**FACULTAD DE INGENIERIA**

**INGENIERIA ELECTRÓNICA**

**NEIVA**

**2010**

Nota de aceptación

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

## RESUMEN

El proyecto consiste en el Diseño e Implementación de un Sistema de Diagnostico Portable para los dispositivos AACP y Mountgrip usados en Aeronaves de combate, herramienta de gran importancia para la optimización de las pruebas de protocolo de seguridad usados en el mantenimiento de las aeronaves; para lograr el objetivo se diseño un sistema basado en microcontroladores, debido a sus características de fácil programación, versatilidad y alto desempeño en aplicaciones de control de sistemas.

Para realizar la interfaz humana (HID) del proyecto, se usaron componentes que permitieran interactuar fácilmente con el microcontrolador como: LCD, teclado matricial y una memoria externa, en la que se almacenan datos específicos de cada dispositivo (AACP y Mountgrip).

El sistema se ensambla en un maletín de uso industrial que soporta diferentes ambientes de trabajo, en el que se alojaron todos los componentes electrónicos como los antes descritos. La conexión entre el maletín y los dispositivos se realizo con dos (2) cables de conexión para cada uno de los dispositivos a diagnosticar. El sistema tiene dos opciones de alimentación; uno de ellos es mediante la red eléctrica AC usual para trabajar en el hangar de mantenimiento y mediante la batería del helicóptero 28VDC.

El sistema implementado puede ser usado de manera independiente o conectado a un PC el cual corre un aplicativo que facilita la configuración y uso del mismo, el programa fue hecho con lenguaje de programación Java de tal manera que fuera entendible y sencillo de manejar.

## ABSTRACT

The Project is about the design and implementation of a Portable Diagnostic System for devices AACP and Mountgrip that are used in Colombian combat aircraft. The Diagnostic System is an important tool in order to optimize the combat aircraft security protocol test. The system is based in microcontrollers, due to its characteristics of easy programming, versatility, and high performance in control systems applications.

With the purpose of develop the Device Human Interface(HID), electronic components were used to interact with the microcontroller, such as LCD (Liquid Crystal Display), matrix keypad and an extern memory that stores specified data for each device (AACP and Mountgrip).

The component previously mentioned will be into an industrial briefcase that stands different environmental work situations, two connection wires were designed to get connection between the Industrial briefcase and each device(AACP and Mountgrip). The System power supply has two options, one way is the A.C. electric net usually to operate inside the maintenance hangar, the other way is through the 28 VDC aircraft battery.

The System developed can be used in an independent way or connected to a PC that execute an application to easily configure the Diagnostic System, the program user interface was developed in Java, it's understandable and easy to manage.

## TABLA DE CONTENIDO

	pág.
<b>INTRODUCCION</b> .....	11
<b>1. REQUERIMIENTOS</b> .....	12
1.1 Microcontroladores.....	13
1.2 EEPROM 24LC512.....	16
<b>2. DISPOSITIVOS MOUNTGRIP Y AACP</b> .....	19
2.1 MOUNTGRIP.....	19
2.2 AACP.....	26
2.2.1 Diagrama de conexiones AACP.....	28
2.2.2 Configuración cables de conexión.....	30
2.2.3 Tabla de datos.....	34
<b>3. DISEÑO DE FIRMWARE Y HARDWARE</b> .....	36
3.1 FIRMWARE.....	36
3.1.1 Formato de tabla de datos.....	37
3.1.1.1 Número de dispositivos.....	38
3.1.1.2 Índice de la tabla de datos.....	38
3.1.1.3 Nombre del dispositivo.....	39
3.1.1.4 Fuente a conmutar.....	39
3.1.1.5 Nombre del contactor.....	39
3.1.1.6 Información del contactor.....	39
3.1.2 Funciones principales.....	40
3.1.3 Librerías utilizadas.....	44
3.1.3.1 Funciones y procedimientos básicos.....	44
3.1.3.2 Multifuncionalidad del teclado matricial.....	45
3.1.3.3 Manejo de cadenas de texto en memoria serial I2C y LDC.....	46
3.2 HARDWARE.....	46
3.2.1 Partes del sistema.....	47
3.2.2 Descripción por etapas.....	49
3.2.3 Fabricación cables de conexión.....	51
3.2.4 Ensamblando el sistema.....	53
<b>4. APLICATIVO INTERFÁZ USB</b> .....	55
4.1 NetBeans IDE.....	55
4.2 Librería jPicUSB.....	56
4.2.1 Ficheros que componen el paquete jPicUSB.....	56
4.2.2 Métodos principales disponibles en jpibus.jar.....	56
4.3 Software de la aplicación.....	57
4.4 Resultado del aplicativo.....	58
<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	61

BIBLIOGRAFÍA.....62  
ANEXOS.....63

## LISTA DE FIGURAS

	pág.
Figura 1. Pines microcontrolador 18F4550.....	16
Figura 2. Bits de direccionamiento.....	17
Figura 3. Selección de dirección de memoria.....	18
Figura 4. Mountgrip.....	19
Figura 5. Contactores dispositivos Mountgrip.....	20
Figura 6. Ojo de gato.....	20
Figura 7. Conector Amphenol Mountgrip.....	21
Figura 8. Conexiones internas de la empuñadura del control de arma.....	22
Figura 9. Cableado interno Mountgrip.....	26
Figura 10. AACP.....	26
Figura 11. Visualizadores AACP.....	27
Figura 12. Conector amphenol AACP.....	28
Figura 13. Pines 1-20 del conector amphenol AACP.....	28
Figura 14. Pines 30-54 conector amphenol AACP.....	29
Figura 15. Diagrama de bloques de la red serial implementada.....	32
Figura 16. Diagrama general del sistema de diagnostico portable.....	36
Figura 17. Diagrama del funcionamiento del sistema.....	37
Figura 18. Formato memoria EEPROM 24LC512.....	38
Figura 19. Mapa memoria EEPROM actualizada.....	39
Figura 20. Diagrama de flujo para agregar un nuevo dispositivo.....	41
Figura 21. Diagrama de flujo para realizar prueba.....	42
Figura 22. Diagrama de flujo para borrar tabla de datos.....	43
Figura 23. Maletín industrial.....	47
Figura 24. LCD 20x4.....	47
Figura 25. Teclado 4x4.....	48
Figura 26. Circuito impreso.....	48
Figura 27. Formas de energizar.....	49
Figura 28. Etapa de potencia.....	49
Figura 29. Etapa de alimentación.....	50
Figura 30. Etapa de señales análogas.....	50
Figura 31. Señales digitales.....	51
Figura 32. Funda de protección.....	52
Figura 33. Cables de conexión terminados.....	52
Figura 34. Circuito impreso sobre el maletín.....	53
Figura 35. LCD y teclado ensamblado.....	53
Figura 36. Sistema de diagnostico portable ensamblado.....	54
Figura 37. Comunicación microcontrolador y PC.....	57
Figura 38. QRead-Qwrite.....	58
Figura 39. Panel principal.....	58
Figura 40. Selección de contactor.....	59
Figura 41. Resultado de un contactor.....	59
Figura 42. Agregar dispositivos.....	60
Figura 43. Borrar tabla de datos.....	60

## LISTA DE CUADROS

	pág.
Cuadro 1. Características familia PIC18FXXXX.....	15
Cuadro 2. Conector J-1.....	23
Cuadro 3. Conexiones Armed (DS1).....	23
Cuadro 4. Conexiones Last_Round(A1DS2).....	24
Cuadro 5. Conexiones Power (S1).....	24
Cuadro 6. Conexiones G-Right (A1S2).....	24
Cuadro 7. Conexiones G-Left (A1S3).....	24
Cuadro 8. Conexiones Push to Fire (A1S4).....	25
Cuadro 9. Conexiones Push to Talk (A2S5).....	25
Cuadro 10. Conexiones AACP.....	29
Cuadro 11. AACP.....	31
Cuadro 12. Mountgrip.....	32
Cuadro 13. Máximo de pines E/S para el sistema de diagnóstico.....	33
Cuadro 14. Distribución multifuncional del teclado matricial.....	45

## LISTA DE TABLAS

	pág.
Tabla 1. Datos AACP.....	34
Tabla 2. Datos Mountgrip.....	35

## INTRODUCCION

Con este libro se pretende dar a conocer el proceso realizado para lograr el Diseño e Implementación de Sistema de Mantenimiento Portable Para AACP (Avionic and Panel control) y MOUNTGRIP (empuñadura y control de mando) en Aeronaves.

El uso de tecnología para el mantenimiento de aeronaves se lleva a cabo mediante la importación de dispositivos necesarios para tal fin. Por tanto, conlleva a las siguientes desventajas: costos elevados, dificultad de mantenimiento y reparación (limitados por el proveedor del equipo). Los motivos mencionados hizo que la Fuerza Aérea Colombiana se interesara por adquirir bancos de diagnóstico, para certificar sus dispositivos implementados en el Arpía III, para el control de armas.

El AH-60-L “ARPÍA III” es un helicóptero basado en la plataforma del UH-60-L “BLACK HAWK”, cuyos sistemas (AACP, Mountgrip y otros) fueron estudiados y seleccionados por el Comando de la Fuerza Aérea Colombiana, así pues la Fuerza Aérea sin invertir tanto como requiere la adquisición de un helicóptero de combate o ataque propiamente dicho, diseñó su propio helicóptero de combate adaptado a las exigencias de la guerra interna que afronta Colombia.

El libro presenta en el primer capítulo los requerimientos para el desarrollo del proyecto; aquí se describen las características de los microcontroladores, centrando el estudio en una familia PIC18F4550 de gama alta muy comercial en estos días. También se implementó una memoria EEPROM externa la cual se describirá en el mismo capítulo.

La segunda parte del libro abarca el segundo y tercer capítulo: el segundo capítulo comprende el estudio realizado de los dispositivos a diagnosticar; se hace una descripción de los contactores y sus características. El tercero describe el desarrollo del firmware, contiene los diagramas de flujo, que permiten la comprensión del mismo y el desarrollo del hardware implementado para el sistema portable. Aquí se explica cada una de las etapas necesarias para el circuito impreso y el proceso de manufacturado de los conectores o, como se denominó, “cables de conexión”.

En la cuarta y última parte del libro se muestra el desarrollo del aplicativo interfaz USB, implementada bajo el lenguaje Java con la interfaz IDE NetBeans, programas gratuitos; el aplicativo desarrollado permite la comunicación envió y recepción de datos entre microcontrolador y computador (PC), así el usuario puede trabajar desde un PC para mayor comodidad.

## 1. REQUERIMIENTOS

El uso de tecnología para el mantenimiento de aeronaves se lleva a cabo mediante la importación de dispositivos necesarios para tal fin; como los RIU<sup>1</sup> de General Electric. Por tanto, conlleva a las siguientes desventajas: costos elevados, dificultad de mantenimiento y reparación (limitados por el proveedor del equipo). Por los motivos mencionados se quiso diseñar e implementar un dispositivo útil para el mantenimiento de aeronaves con recursos que se pueden encontrar en el medio.

Cuando se inició con el diseño del sistema de diagnóstico portable, se planteó como punto de partida la necesidad de un microcontrolador, por su facilidad para ejecutar procesos programados en su memoria. Además, dispone de varios puertos de entrada y salida (E/S) según la familia que se elija.

En un inicio se planteó usar un microcontrolador y con multiplexores aumentar la cantidad de entradas/salidas, pero, en el mercado no se encuentran multiplexores con más de 8 canales, lo que hace la escalabilidad del sistema más compleja; por tanto, se implementó una red de microcontroladores con la capacidad de manejar entre 50 y 60 pines de entrada/salida, además de contar con las opciones de comunicación I2C, USB y serial.

El microcontrolador PIC18F4550 cumple los requisitos y se encuentra fácilmente en el mercado, por lo que se eligió para el desarrollo del sistema. Este microcontrolador cuenta con una limitante en su memoria EEPROM interna 256 bytes, debido a esto, fue necesario usar una memoria externa que se comunicara con el microcontrolador maestro por medio del bus I2C. El uso de esta memoria externa satisface el diseño de un sistema escalable, que permite almacenar varios dispositivos y hacer el respectivo diagnóstico.

Además de los requerimientos de software, era necesario cumplir con unas normas MIL para el diseño del hardware; dentro de estas normas se encuentran las Test method Standard electronic and electrical (MIL-SRD-202), Cables-Radio frequency-Flexible and semirigid (MIL-C-17-G) y Conectores eléctricos cilíndricos (MIL-DTL-38999); las cuales se explican a lo largo del libro.

Para la alimentación del sistema se hizo uso de reguladores de voltaje para estabilizar los valores necesarios que requieren los dispositivos a diagnosticar; la referencia utilizada se presenta en la sección de hardware, descripción por etapas.

---

<sup>1</sup> RIU "Remote Interface Unit": Diseñados para funcionar a una amplia gama de sensores y efectores de aeronaves.

## 1.1 MICROCONTROLADORES

Es un circuito integrado que permite realizar funciones similares a las de una computadora, pero no contiene dispositivos para la comunicación con humanos. Estos circuitos integrados fueron diseñados para el control de máquinas más que para interactuar con humanos.

Características generales

Un microcontrolador dispone normalmente de los siguientes componentes:

- ✓ Procesador o UCP (Unidad Central de Proceso).
- ✓ Memoria RAM para Contener los datos.
- ✓ Memoria para el programa tipo ROM/PROM/EPROM.
- ✓ Líneas de E/S para comunicarse con el exterior.
- ✓ Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: conversores Digital/Analógico, etc.).
- ✓ Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que para su regulación incorporan un micro controlador, disponen de las siguientes ventajas:

- ✓ Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
- ✓ Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.
- ✓ Reducción del tamaño en el producto acabado: La integración del micro controlador en un chip disminuye el volumen, la mano de obra y los stocks.
- ✓ Mayor flexibilidad: las características de control están programadas, por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es, en definitiva, un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño, es posible montar el controlador en el propio dispositivo al que gobierna.

**1.1.1 MICROCONTROLADOR MICROCHIP:** Los microcontroladores de Microchip son muy reconocidos en el mundo y el número 1 en ventas. Algunas ventajas son:

- Amplia gama: Gran variedad de familias que permite adaptar el uC (microcontroladores) a las necesidades de cada aplicación.
- Herramientas de desarrollo comunes.

- Gran variedad de unidades funcionales embebidas (temporizadores, USART, I2C, SPI, unidades de comparación/captura/PWM, Convertidores A/D, USB, receptores/transmisores de RF, Ethernet, etc...).
- Precios competitivos
- Buen soporte (datasheet, libros, notas de aplicación, seminarios y mucha información disponible en internet)

#### Familias de Microcontroladores

- PIC10: de 8 bits, de bajo coste, 6 pines y bajas prestaciones.
- PIC12: de 8 bits, de bajo coste de 8 pines y bajas prestaciones.
- PIC16: de 8 bits, gran variedad de pines y prestaciones medias.
- PIC 18: de 8 bits, con gran variedad de numero de pines y prestaciones medias/altas.
- PIC 24: de 16 bits.
- dsPIC'S

De las familias disponibles se decidió trabajar con la familia PIC 18 por su gran variedad de numero de pines E/S (Entrada/Salida) y porque permite prestaciones media/alta, además de las ventajas escritas anteriormente.

#### Características fundamentales de la familia PIC 18:

- Arquitectura RISC avanzada Harvard: 18 bit con 8 bit de datos.
- 77 interrupciones
- Desde 18 a 80 pines
- Hasta 64K bytes de programa (hasta 2 Mbytes en ROMless)
- Hasta 3968 bytes de RAM y 1Kbyte de EEPROM
- Frecuencia máxima de reloj 40MHz
- Pila de 32 niveles
- Múltiple fuentes de interrupción
- Periféricos de comunicación avanzados (CAN y USB).

En el Cuadro 1 se describen las características de la familia PIC 18FXXXX

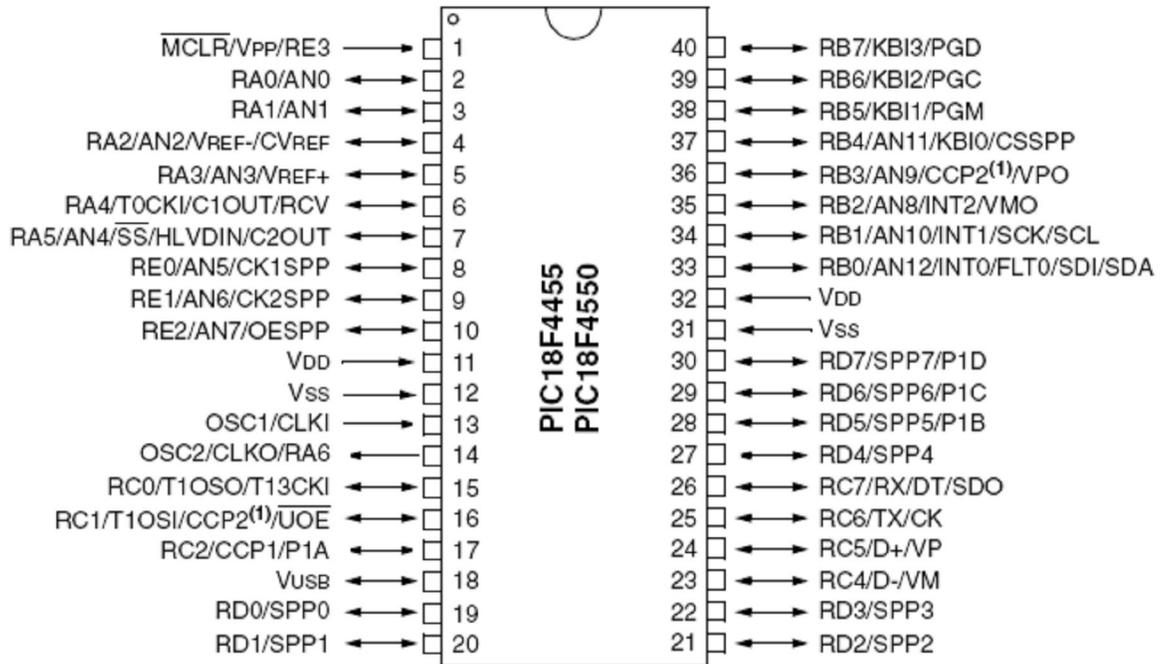
**Cuadro 1. Características Familia PIC18FXXXX**

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Operating Frequency	DC – 48 MHz			
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Data Memory (Bytes)	2048	2048	2048	2048
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/Compare/PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1	1	1	1
Streaming Parallel Port (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Comparators	2	2	2	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

Fuente: PIC18F2455/2550/4455/4550, Data Sheet 28/40/44-Pin, pág. 9

- **MICROCONTROLADOR PIC18F4550:** en la Figura 1 se puede observar el diagrama de conexionado del microcontrolador 18F4550

**Figura 1.** Pines microcontrolador 18F4550



Fuente: PIC18F2455/2550/4455/4550, Data Sheet 28/40/44-Pin, pág. 2

Las descripciones de los puertos E/S (Entradas/Salidas) se puede remitir a las hojas de datos que provee Microchip.

## 1.2 EEPROM 24LC512

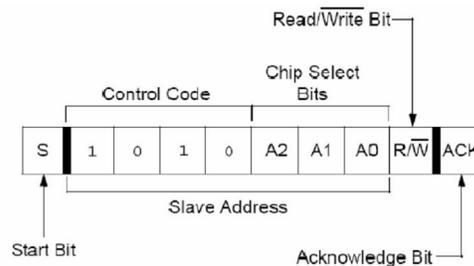
El circuito integrado 24LC512 es una memoria serie del tipo 64K x 8 EEPROM, es decir, una memoria eléctricamente borrable de 512 Kbits, capaz de operar en un amplio rango de tensiones (2.5V a 5.5V) y a una frecuencia máxima de reloj de 400KHz.

El 24LC512 puede operar de dos modos diferentes, en cuanto a transmisión de datos se refiere. Un dispositivo que envía datos sobre el bus se define como transmisor, mientras que un dispositivo que recibe datos a través del mismo se define como receptor. El 24LC512 puede funcionar como emisor o bien como receptor. Típicamente el 24LC512 funcionará como parte de un sistema maestro-esclavo, donde el papel de maestro será desempeñado usualmente por un microprocesador y la memoria estudiada será el esclavo. El maestro controlará el bus, a través del cual memoria y microprocesador quedan comunicados, generando la señal de reloj (SCL), controlando el acceso al mismo y generando la señalización necesaria como las condiciones de comienzo y de parada.

**1.2.1 Direccionamiento de los dispositivos:** pueden existir varios dispositivos conectados a un mismo bus I2C, que funcionarán de una manera correcta siempre que cumplan las especificaciones del y los formatos de trama establecidos. En el caso de la memoria 24LC512, estas tramas toman una forma determinada, como se expone a continuación:

Al iniciar es generado y enviado el bit de Start por parte del máster, que es seguido por el primer byte de la comunicación, denominado byte de control. En el caso de la memoria 24LC512 el byte de control presenta la siguiente estructura (Figura 2):

**Figura 2.** Bits de direccionamiento



Fuente: compilador C CCS y Simulador Proteus para Microcontroladores PIC, pág 201.

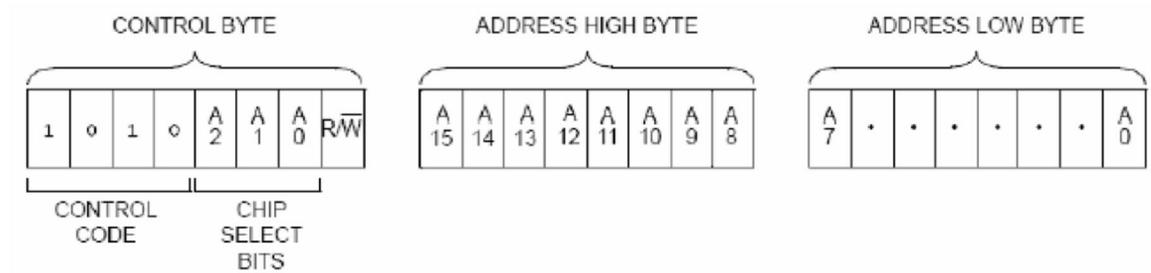
Los cuatro primeros bits conforman el código de control y siempre toman un valor fijo que en el caso del 24LC512 es '1010', tanto para las operaciones de lectura como de escritura (Figura10). Los tres bits siguientes son los que permiten la selección de chip o dispositivo y se denominan A2, A1 y A0 respectivamente. Estos tres últimos bits pueden tomar cualquier valor, por lo que son usados para seleccionar y acceder hasta 8 dispositivos 24LC512 conectados al mismo bus.

El bit siguiente dentro del byte de control anteriormente expuesto, define el tipo de operación a realizar. Cuando este bit presenta un nivel lógico 1 se realiza una operación de lectura, mientras que si se encuentra a nivel bajo, se realizará una operación de escritura.

A continuación son enviados dos bytes que definen la dirección del dato a leer o escribir. Estos 16 bits permiten direccionar hasta 64K posiciones de memoria, lo que en el caso del 24LC512 supone 512 Kbits, pues se trata de una memoria 64K x 8. Esta cantidad de posible memoria a direccionar, puede extenderse hasta 4 Mbits haciendo uso de los bits de Chip Select, que permite la conexión de hasta ocho dispositivos de este tipo al mismo bus (En este caso 4Mbits=512Kbits x 8). Después de la condición de comienzo, el dispositivo estudiado chequea el bus,

buscando la existencia del identificador apropiado. Si recibe un '1010' y una combinación de chip select adecuada, el dispositivo esclavo genera una señal de reconocimiento o ACK. Dependiendo del valor del bit de lectura/escritura, la operación a realizar será una u otra sobre la dirección de memoria determinada por los dos bytes siguientes ver Figura 3.

**Figura 3.** Selección de dirección de memoria



Fuente: compilador C CCS y Simulador Proteus para Microcontroladores PIC, pág 201.

## 2. DISPOSITIVOS MOUNTGRIP Y AACP

Estos dos dispositivos son utilizados en aeronaves militares, que inicialmente fueron construidos para el reconocimiento o transporte de tripulación, pero los técnicos de la FAC (Fuerza Aérea Colombiana) poco a poco fueron ensamblando por sus propios medios contenedores de armas y cohetes; obteniendo su propia versión Blackhawk denominada AH-60L Arpía II / III. La versión III es una versión con mejor electrónica para el control de tiro.

### 2.1 MOUNTGRIP

Es una empuñadura que controla el arma de referencia 219F981 (Figura 4). Sus contactores de control accionan armas de asalto; esta empuñadura cuenta con un contactor central que activa todo el dispositivo llamado Power (Figura 5), si este no se acciona, nada se activará aun cuando se presione. Tiene dos visualizadores llamados “ojos de gato” que se activan y desactivan directa e indirectamente. El ojo de gato con título Armed o armado (Figura 5); se activa al presionar el contacto central indicando que todo el sistema está activo; este visualizador si se desea desactivar se gira al lado izquierdo y deja de alumbrar.

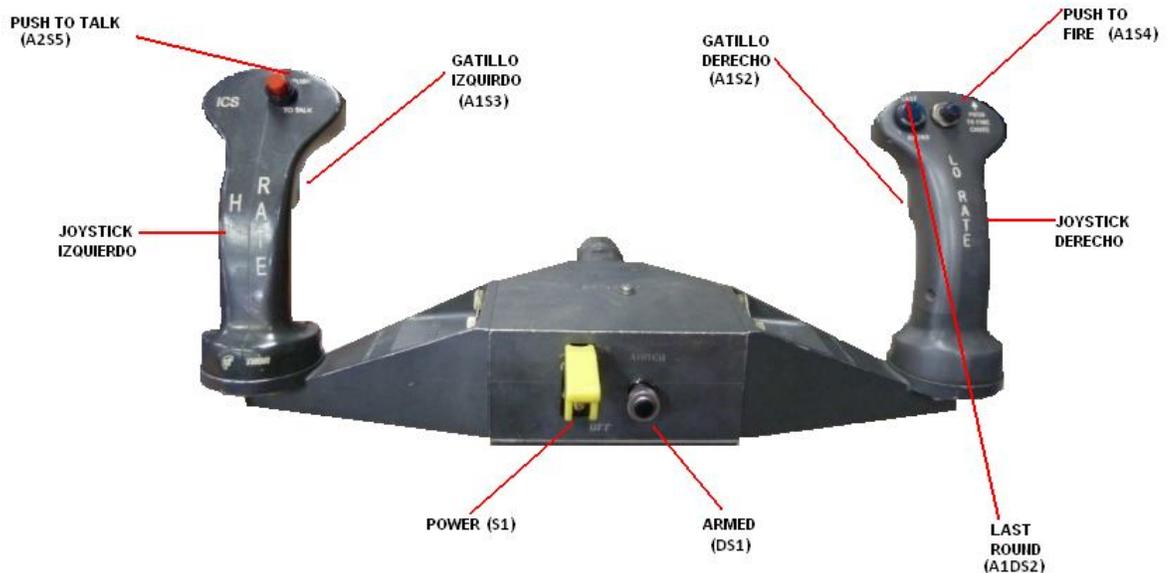
**Figura 4.** Mountgrip



El Joystick derecho llamado Lo-Rate tiene tres (3) elementos: un visualizador “ojo de gato” llamado Last Round o última ronda (Figura 5), se acciona desde otro dispositivo ajeno a esta empuñadura; este visualizador le indica al piloto que tiene la última carga de munición e igual que al ojo de gato Armed, se desactiva rotando a la izquierda ó mediante el contactor Push To Fire (Figura 5), que se acciona para dar la orden de hacer uso de la última carga de munición, activa las municiones de reserva. En el mismo joystick derecho se encuentra un gatillo que al ser accionado dispara los cohetes con bajo alcance (Figura 5).

El Joystick izquierdo o Hi-Rate tiene dos elementos, Push to Talk (Figura 5); este permite al piloto comunicarse con la cabina central. El Gatillo izquierdo al ser presionado dispara los cohetes con largo alcance (Figura 5).

**Figura 5.** Contactores dispositivo Mountgrip



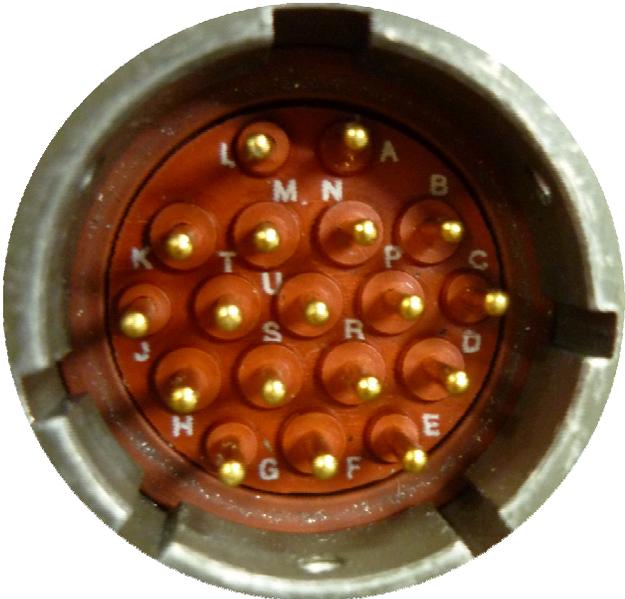
Ojos de Gato: es un visualizador y a la vez un contactor pulsador, son objetos muy poco comunes en el mercado, se podría decir que son útiles en la aeronáutica, sus descripciones son las siguientes: son muy similares a un bombillo, con una función muy especial; tienen una perilla giratoria que abre y cierra simulando ser un ojo, por esto su nombre. El bombillo funciona con un voltaje entre 12.5V y 28V con 7.5 Ohmios de resistencia interna en frío (Figura 6).

**Figura 6.**Ojo de gato



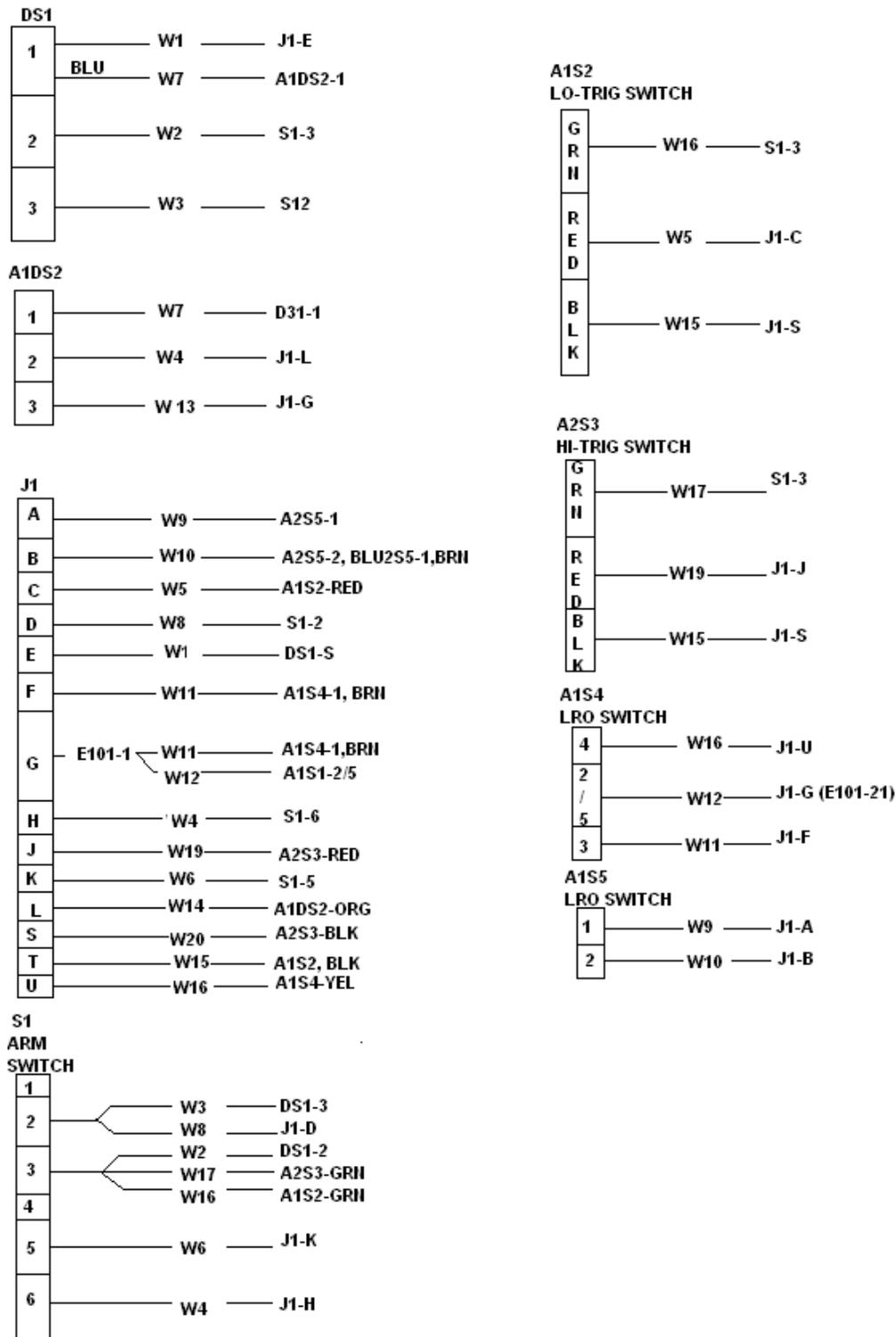
Esta empuñadura tiene un conector Amphenol macho de referencia M85049/31-14W. Los primeros datos indican que es un componente backshells de tipo circular y la segunda parte de la referencia, indica la forma como están distribuidos los pines (Figura 7).

**Figura 7.** Conector Amphenol Mountgrip



En la Figura 8 se presentan las conexiones internas de los contactores que componen el dispositivo Mountgrip, esta información se adquirió previo a las mediciones; debido que en un inicio no se pudo acceder a trabajar con los dispositivos directamente.

**Figura 8.** Conexiones internas de la empuñadura del controlador de arma



Fuente: Fuerza Aérea Colombiana

Para realizar las primeras pruebas de escritorio en la base aérea de Rio Negro, lugar donde se tenía acceso a estos dispositivos, era necesario llevar claro las mediciones a realizar, motivo por el cual se realizaron los cuadros mostrados a continuación:

En el Cuadro 2 se pueden observar los datos de las trayectorias de cada uno de los pines del conector J1, que hace referencia al Amphenol M85049/31-14W del Mountgrip; permitiendo la conexión con el conjunto controlador de arma GAU 19A (Gatling Gun) del Helicóptero Arpía III.

### Cuadro 2. Conector J1

Señal Enviar	Activar contactor(es)		Leer Retorno
J1A	A2S5(1,2)		J1B
J1B	A2S5(2,1)		J1A
J1C	A1S2(R,B)		J1S*(o T)
J1D	S1(2,5)		J1K
J1E	DS1(1,2)	S1(2,5)	J1H
J1F	A1S4(B,Y)		J1U
J1G	E101-1	A1S4(B,Y)	J1U
	E101-1	A1S4(W,B)	J1F
J1H	S1(6,5)		J1K
J1J	A2S3(R,B)		J1S
J1K	S1(5,6)		J1H
J1L	A1DS2(O,G)		J1G
J1S	A2S3(B,R)		J1J
J1T	A1S2(B,R)		J1C
J1U	A1S4(Y,W)		J1G

El Cuadro 3 contiene la información del visualizador y contactor Armed, en él se muestran los pines por los cuales es activado y además de esto, los contactores con los que se relaciona, es decir, es afectado o afecta al ser presionado, en este caso con el contactor Power (S1).

### Cuadro 3. Conexiones Armed (DS1)

DS1			
Señal Enviar	Activar contactor(es)		Leer Retorno
J1E	DS1(1,2)	S1(3,2)	J1D
J1E	DS1(1,3)		J1D
J1D	DS1(2,3)	S1(3,5)	J1K

El Cuadro 4 contiene la información del visualizador y contactor Last\_Round, en el se muestran los pines por los cuales es activado. En este caso se puede ver que no está relacionado con ningún otro contactor.

#### Cuadro 4. Conexiones Last\_Round (A1DS2)

A1DS2			
Señal Enviar	Activar contactor(es)		Leer Retorno
J1L	A1DS2(1,2)		J1E
J1G	A1DS2(1,3)		J1E
J1L	A1DS2(2,3)		J1G

Las conexiones del contactor Power se observan en el Cuadro 5, aquí están los pines con los cuales interactúa, observando que tiene varios pines en común, haciendo de este el contactor maestro.

#### Cuadro 5. Conexiones Power (S1)

S1			
Señal Enviar	Activar contactor(es)		Leer Retorno
J1D	S1(2,3)	DS1(1,2)	J1E
J1D	S1(2,5)		J1K
J1D	S1(2,6)		J1H
J1K	S1(3,5)	DS1(1,2)	J1E
J1H	S1(3,6)	DS1(1,2)	J1E
J1K	S1(5,6)		J1H

El Cuadro 6 muestra las conexiones a las que está implicado el contactor pulsador Gatillo derecho o como se nombró G-Right; al igual en el Cuadro 7 se muestran las conexiones del contactor pulsador Gatillo izquierdo o G-Left.

#### Cuadro 6. Conexiones G-Right (A1S2)

A1S2			
Señal Enviar	Activar contactor(es)		Leer Retorno
J1C	A1S2(R,B)		J1S
J1C	A1S2(G,R)	A2S3(G,R)	J1J
J1S	A1S2(G,B)	A2S3(G,R)	J1J

#### Cuadro 7. Conexiones G-Left (A1S3)

A2S3			
Señal Enviar	Activar contactor(es)		Leer Retorno
J1J	A2S3(R,B)		J1S
J1J	A2S3(G,R)	A1S2(G,R)	J1C
J1S	A2S3(G,B)	A1S2(G,R)	J1C

En el Cuadro 8 está la información recopilada de la Figura 5 acerca del contactor pulsador Push to Fire.

**Cuadro 8.** Conexiones Push to Fire (A1S4)

<b>A1S4</b>			
<b>Señal Enviar</b>	<b>Activar contactor(es)</b>		<b>Leer Retorno</b>
J1U	A1S4(Y,W)		J1G(E101-21)
J1U	A1S4(Y,B)		J1F
J1G	A1S4(W,B)		J1F

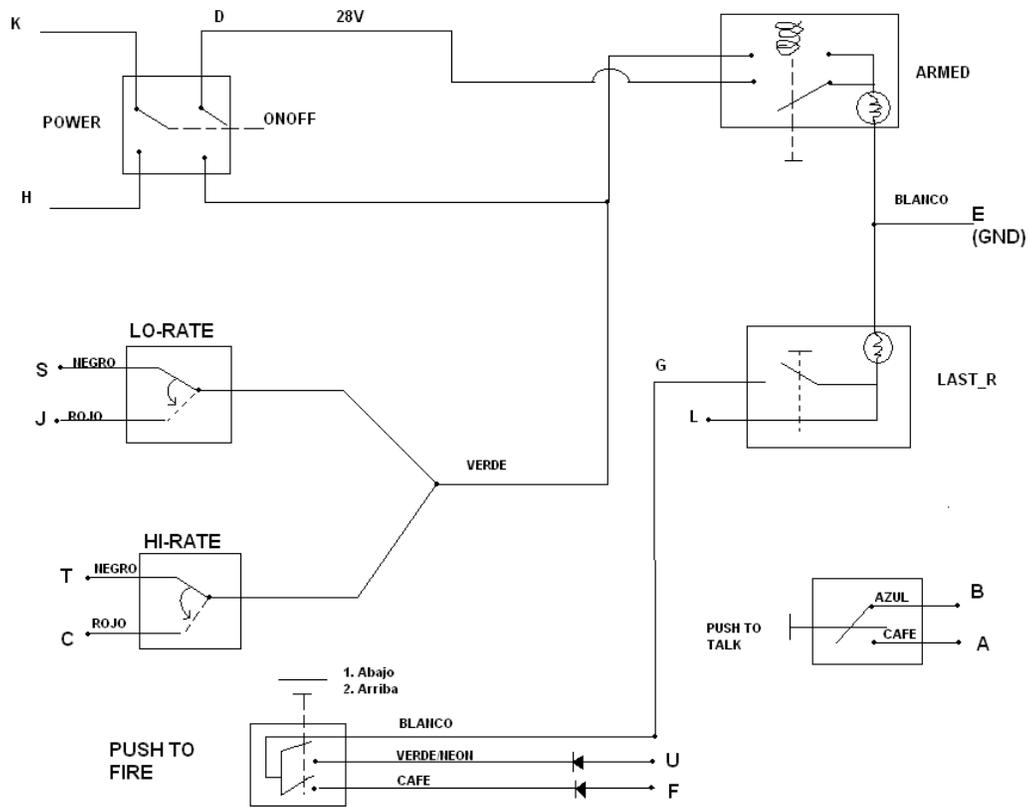
El contactor pulsador Push to Talk es el más independiente de todos los otros contactores, pues se puede observar que tiene un par de pines para ser activado.

**Cuadro 9.** Conexiones Push to Talk (A2S5)

<b>A2S5</b>			
<b>Señal Enviar</b>	<b>Activar contactor(es)</b>		<b>Leer Retorno</b>
J1A	A2S5(1,2)		J1B

En el momento de realizar las primeras pruebas de escritorio con los dispositivos, en la base aérea de Rio Negro y comparando la información extraída de la Figura 6 con las tablas antes mostradas, se obtuvo un diagrama más explícito de cada uno de los contactores y su interacción en conjunto. Esta información se puede observar en la Figura 9.

**Figura 9.** Cableado interno Mountgrip



## 2.2 AACP

Estas siglas traducen Avionic Armament Panel Control (Panel de Control de armamento Aviónico). Este dispositivo está ubicado en la consola central y se activa mediante el contactor principal, denominado M-ARM (Figura 10).

**Figura 10.** AACP



A continuación se da una breve explicación de cada uno de los contactores de la Figura 10.

M-ARM: es un contactor de tres (3) posiciones: la posición ARM controla la alimentación del sistema de armas; SAFE informa que el sistema es inofensivo y OFF, que todo el dispositivo está desactivado.

ROCKETS: es un contactor de dos (2) posiciones: la ON activa la coherería y la posición OFF, la desactiva.

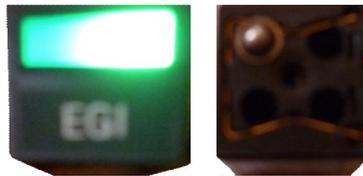
L-RANGE: es un contactor de tres (3) posiciones, que activa telémetro láser y lo acciona en dos modos: LAST, activa el laser en modo último eco, FIRST, activa el láser en modo primer eco y OFF, desactiva esta opción.

L-MARK: contactor de dos (2) posiciones: posición ON, que activa el sistema EOP -Electro Optic Payload (Carga útil electro óptico)- y posición OFF, que desactiva esta función.

GUNS: contactor de dos (2) posiciones ON/OFF, activa y desactiva el voltaje de las ametralladoras.

Los siete (7) elementos que se ven en la Figura 10, son visualizadores que se alimentan a 5V: M-Avion es el contactor maestro, el cual bloquea o desbloquea la interrupción de estos. Cada uno de estos contactores visualizadores debe ser presionado para ser activado, dando paso a la simbología del casco del piloto (Figura 11).

**Figura 11.** Visualizadores AACP



Cada uno de los visualizadores contiene 4 bombillos internos muy pequeños, que se alimentan a 5V c/u; estos alumbran en parejas debido a que están en paralelo. Los bombillos de la parte inferior son para los títulos del panel y los de la parte superior para el indicador (luz verde).

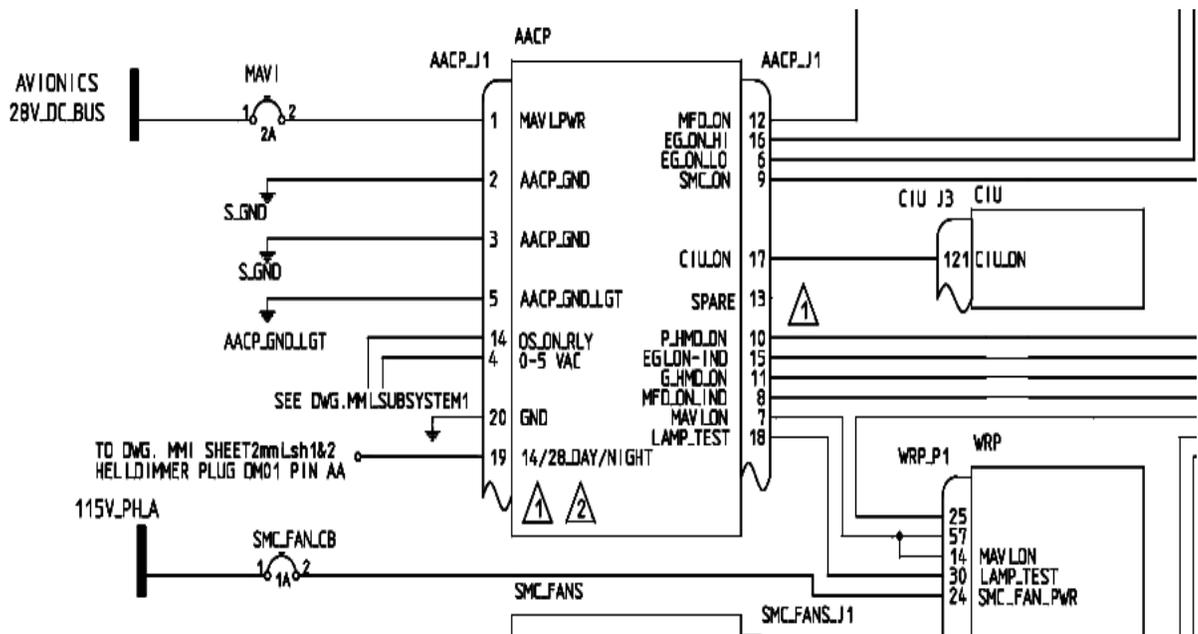
Este panel de control tiene un conector Amphenol de referencia JD38999/20WE35PN; los primeros datos indican que es un componente backshells de tipo circular y la segunda parte de la referencia, indica la forma de cómo están distribuidos los pines (Figura 12).

**Figura 12.** Conector Amphenol AACP



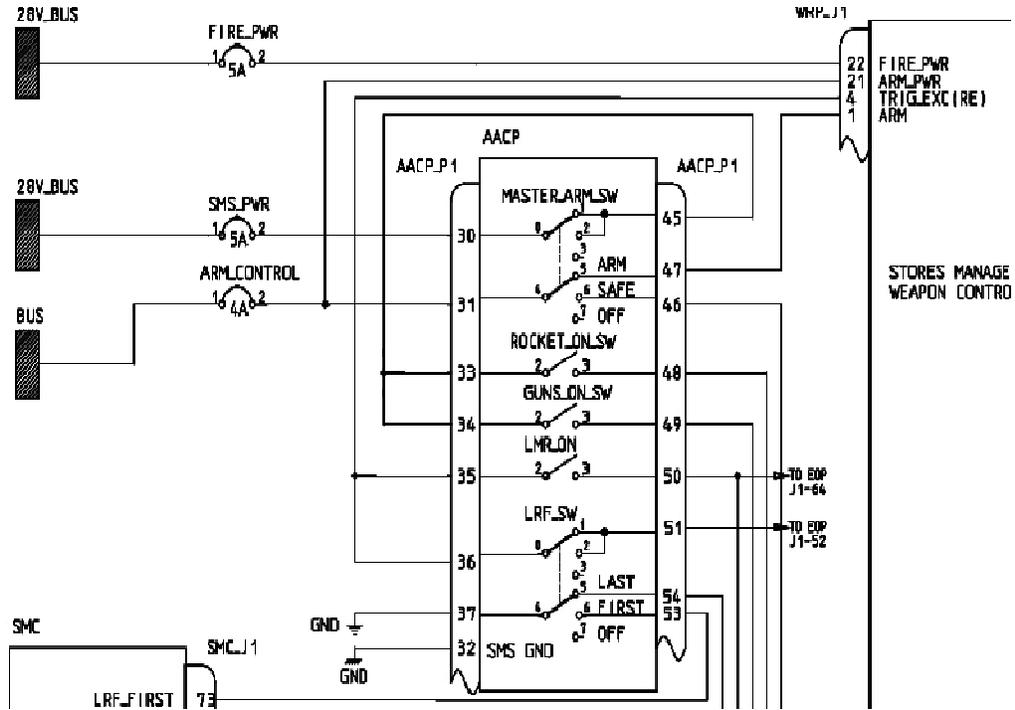
**2.2.1 Diagrama de conexiones AACP:** en las siguientes Figuras 13 y 14 se pueden observar la distribución y conexión de los pines del Amphenol de la Figura 12.

**Figura 13.** Pines 1-20 del conector Amphenol de AACP



Fuente: Pdf. WMS (Weapon Management System), pág. 2.

**Figura 14.** Pines 30-54 Conector Amphenol del AACP



Fuente: Pdf. WMS (Weapon Management System), pág. 8.

Como se puede ver, se tiene un pin para envío de señal y otro para retornar. En las primeras pruebas de escritorio se diseñó unos cuadros para cada uno de los dispositivos; la información mostrada en el Cuadro 10 corresponde a las conexiones del AACP.

**Cuadro 10.** Conexiones AACP

AACP	GND	VCC	Master_ARM		ARM/SAFE/OFF			ROCKET	GUNS	LMR	LRF_SW		LAST/FIRST/OFF		
			1(45)	2(45)	(47)	(46)	(OFF)	48	49	50	51	51	54	53	OFF
1		X													
2	X														
3	X														
5	X														
20	X														
30			X	X											
31					X	X									
33								X							
34									X						
35										X					
36											X	X			
37													X	X	
32	X														

**2.2.2 Configuración cables de conexión:** al comparar los conectores Amphenol de las Figuras 4 y 10, se puede observar que las distribuciones de pines y sus referencias son diferentes: el conector Amphenol del Mountrgrip cuenta con tan solo 18 pines con denominación A, B C, D, E, F, G, J, K, L, M, N, O, P, R, S, T, U y el AACP de 55 pines, con distribución numérica de 1-55. El conector a usar en el maletín sin duda es el Amphenol JD38999/20WE35PN macho, el cual tiene 55 pines con el que se podría hacer el diagnóstico tanto para AACP como para el Mountgrip.

Ya elegido el conector para el maletín, era necesario crear una conexión entre el dispositivo y el maletín, lo que se denominó “cables de conexión”, con conectores Amphenol hembra en sus extremos, para realizar el diagnóstico de los dispositivos. En los siguientes cuadros se puede ver cómo se conectaron los pines de cada uno de los conectores de los dispositivos a diagnosticar, para entender los datos finales que se utilizaron para el firmware del dispositivo.

En el Cuadro 11 se muestra la equivalencia entre el conector AACP y el conector del maletín.

**Cuadro 11. AACP**

Nombre sw	AACP	MALETIN
TITULOS	4	1
TITULOS	5	8
LUz-SW	20	2
LUZ-SW	19	9
SMC	9	31
P-HMD	10	32
G-HMD	11	35
EGI	15	36
M-AVION	1	33
MFD	12	34
EOP	14	37
RETORNO DE SW	17	38
TALK	30	15
TALK	45	16
M-ARM	31	17
M-ARM	47	18
SAFE	46	19
ROCKETS	33	20
ROCKETS	48	21
GUNS	34	22
GUNS	49	23
L-MARK	35	24
L-MARK	50	25
L-RANGE-OF1	36	26
L-RANGE-OF1	54	29
LAST	37	28
LAST	54	29
FIRST	53	30
FIRST	37	28

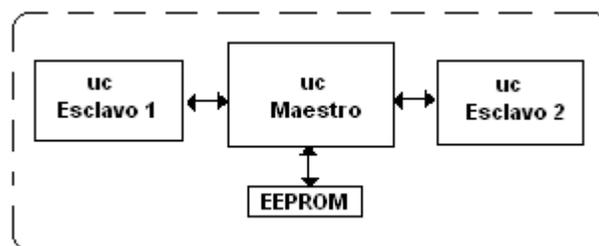
En el Cuadro 12 se muestra la equivalencia entre el conector del Mountgrip y el conector del maletín.

**Cuadro 12.** Mountgrip

	MOUNTGRIP	MALETIN
ARMED	D	18
ARMED	E	1
LAST_R	L	8
LAST_R	E	1
P-FIRE	F	22
P-FIRE	U	23
LO-RATE	C	17
LO-RATE	S	20
HI-RATE	J	19
HI-RATE	T	21
P-TALK	A	15
P-TALK	B	16
POWER	H	25
POWER	K	24

Como se puede ver en los Cuadros 11 y 12, el maletín conserva una equivalencia que debe respetarse para los dos dispositivos. La cantidad de señales digitales y análogas las determinan la cantidad de pines E/S del microcontrolador. Para el sistema se utilizaron 3 microcontroladores. Para lograr la escalabilidad, estos están configurados de la siguiente manera (Figura 15):

**Figura15.** Diagrama de bloques de la red serial implementada



- El microcontrolador maestro manipula la LCD, el teclado, la EEPROM 512Kb, la comunicación con la interfaz grafica de usuario (GUI) y los dos esclavos. En consecuencia el microcontrolador deja muy pocos pines de E/S para el propósito de diagnostico.
- Los dos (2) microcontroladores restantes son los esclavos que almacenan una tabla de datos que se explica en el Firmware (capitulo 3). Los esclavos permiten tener 24 pines de E/S disponibles para las pruebas de diagnostico.

En el Cuadro 13 se pueden ver las equivalencias del conector del maletín con los microcontroladores a utilizar, el J-1 es el Amphenol del maletín con 55 pines para el diagnóstico de los dispositivos.

**Cuadro 13.** Máximo de pines E/S para el sistema de diagnóstico

Conector Maletín J-1	Pines de uC PIC's	Conector Maletín J-1	Pines de uC PIC's
1	GND	28	RC1-S1
2	GND	29	RC2-S1
3	GND	30	RD0-S1
4	GND	31	RD1-S1
5	GND	32	RD2-S1
6	GND	33	RD3-S1
7	GND	34	RD4-S1
8	RA1-M	35	RD5-S1
9	RA2-M	36	RD6-S1
10	RA3-M	37	RD7-S1
11	RA4-M	38	RE0-S1
12	RA5-M	39	RE1-S1
13	RE1-M	40	RE2-S1
14	RE2-M	41	RE3-S1
15	RA0-S1	42	RA0-S2
16	RA1-S1	43	RA1-S2
17	RA2-S1	44	RA2-S2
18	RA3-S1	45	RA3-S2
19	RA4-S1	46	RA4-S2
20	RA5-S1	47	RA5-S2
21	RB0-S1	48	RB0-S2
24	RB3-S1	49	RB1-S2
25	RB4-S1	52	RB4-S2
26	RB5-S1	53	RB5-S2
27	RC0-S1	54	RB5-S2
		55	RC0-S2

Los primeros 14 pines del conector del maletín se dejan para las señales análogas, se implementan para el accionamiento de los visualizadores que funcionan a un voltaje superior al que soportan los microcontroladores (5V). La numeración 15 en adelante son pines que se utilizan como E/S para el propósito de diagnóstico de los contactores que manejan señales iguales a 5 Voltios.

**2.2.3. Tabla de datos:** el microcontrolador maestro ejecuta una lista ordenada de los contactores a diagnosticar, esta lista está almacenada en las localidades de la memoria EERPOM externa, los datos están en la Tabla1 y Tabla 2.

La información que contiene la memoria EEPROM es la que indica al microcontrolador maestro el tipo de prueba que debe hacer según el contactor a diagnosticar. Hay tres (3) clases que puede ejecutar el microcontrolador, la clase cero (0) para los contactores pulsadores en estado activo, clase uno (1) es para los contactores visualizadores sin retorno, y clase dos (2) para los contactores en estado desactivado (OFF). La salida indica el envío de la señal de prueba y el retorno realiza la lectura de la señal de prueba. La numeración de salida y retorno corresponde a los pines de los microcontroladores esclavos. El retorno para los contactores de clase uno (1) no aplica (N.A).

**Tabla 1. Datos AACP**

SW AACP			SW/Visualizadores AACP		
M-ARM	Clase	2	M-AVION	Clase	0
	Salida	10		Salida	26
	Retorno	N.A		Retorno	17
SAFE	Clase	0	MFD	Clase	0
	Salida	10		Salida	27
	Retorno	12		Retorno	31
ARM	Clase	0	SMC	Clase	0
	Salida	10		Salida	24
	Retorno	11		Retorno	31
ROCKETS	Clase	0	EGI	Clase	0
	Salida	13		Salida	29
	Retorno	14		Retorno	31
LAST	Clase	0	G-HMD	Clase	0
	Salida	21		Salida	28
	Retorno	22		Retorno	31
FIRST	Clase	0	EOP	Clase	0
	Salida	21		Salida	30
	Retorno	23		Retorno	31
L-MARK	Clase	0	P-HMD	Clase	0
	Salida	17		Salida	25
	Retorno	18		Retorno	31
GUNS	Clase	0	Pilotos	Clase	1
	Salida	15		Salida	2
	Retorno	16		Retorno	N.A

**Tabla 2. Datos Mountgrip**

Mountgrip		
ARMED	Clase	1
	Salida	3
	Retorno	N.A
LAST-ROUND	Clase	1
	Salida	1
	Retorno	N.A
P-FIRE	Clase	0
	Salida	15
	Retorno	16
LO-RATE N.A	Clase	0
	Salida	14
	Retorno	13
LO-RATE Acticado	Clase	0
	Salida	10
	Retorno	13
HI-RATE N.A	Clase	0
	Salida	13
	Retorno	14
HI-RATE Acticado	Clase	0
	Salida	12
	Retorno	14
P-TALK	Clase	0
	Salida	8
	Retorno	9
POWER	Clase	0
	Salida	18
	Retorno	17

### 3. DISEÑO DEL FIRMWARE Y HARDWARE

#### 3.1 FIRMWARE

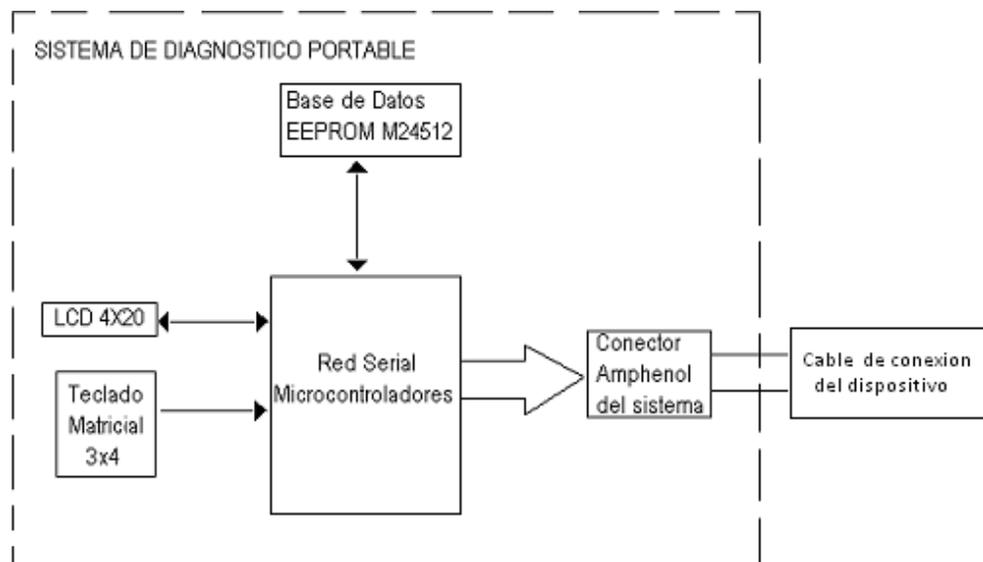
En el diseño del firmware del proyecto se tiene en cuenta básicamente la creación y el manejo de una tabla de datos dinámica, lo cual quiere decir que la información almacenada se puede modificar, borrar o actualizar con el tiempo.

La tabla de datos es almacenada en una EEPROM externa (24LC512), la cual contiene la información necesaria para realizar la prueba de un determinado dispositivo.

Para la gestión de la tabla de datos es empleada una red serial de 3 microcontroladores, uno de ellos trabaja en modo maestro el cual es encargado de ingresar y leer la información almacenada en la EEPROM externa y compartirla con los dos microcontroladores esclavos, los cuales tienen como función interpretar la información y realizar las pruebas de lectura pertinentes para cada dispositivo, en la Figura 16 se observa el diagrama general del proyecto. El microcontrolador Maestro también se encarga de manejar los periféricos que se necesitan y principalmente ellos son:

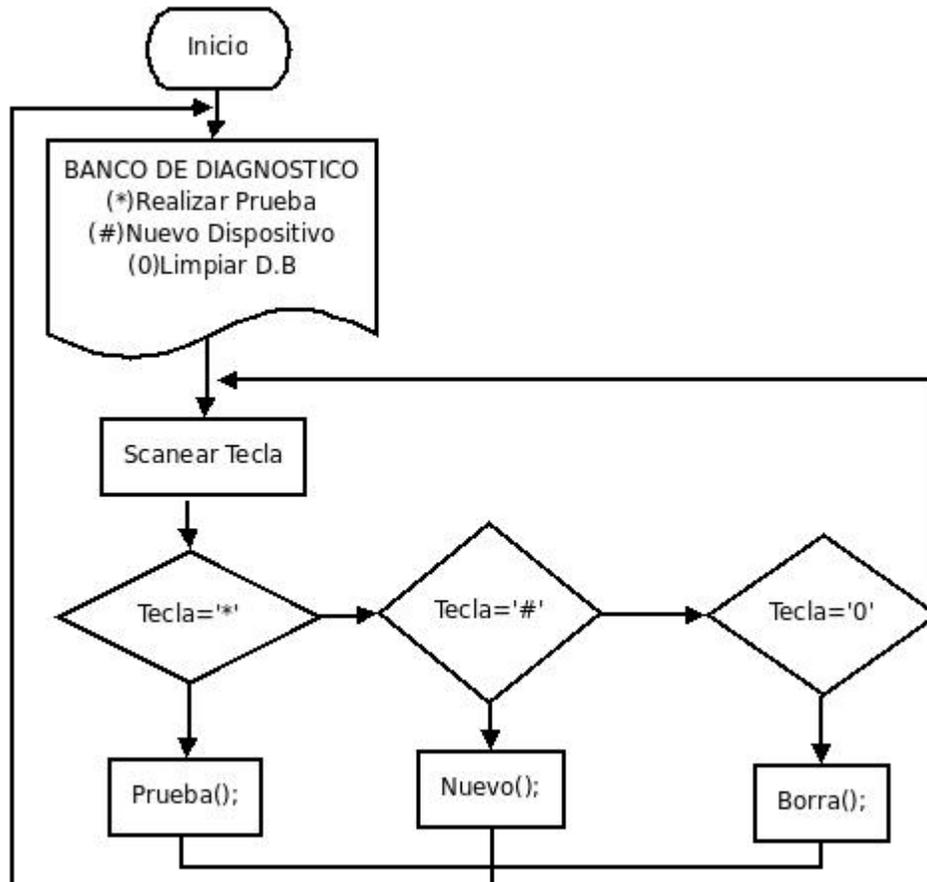
- Teclado matricial de 3X4, necesario para ingresar información y navegar a través del programa.
- LCD 4x20 (pantalla de cristal liquido de cuatro líneas y 20 caracteres, símbolos por línea.)
- Memoria EEPROM externa serial 24LC512, manejada mediante protocolo I2C.

**Figura 16.** Diagrama general del sistema de diagnóstico portable.



El diagrama de la Figura 17, muestra el funcionamiento del sistema:

**Figura 17.** Diagrama del funcionamiento del sistema



Como se puede apreciar en la Figura 17, el firmware del proyecto está compuesto básicamente de tres funciones. A grandes rasgos cada una realiza lo siguiente:

Prueba(), se lleva a cabo la lectura de la tabla de datos y se realiza la prueba para cada dispositivo elegido.

Nuevo(), se agrega información correspondiente al funcionamiento de un dispositivo nuevo a la tabla de datos.

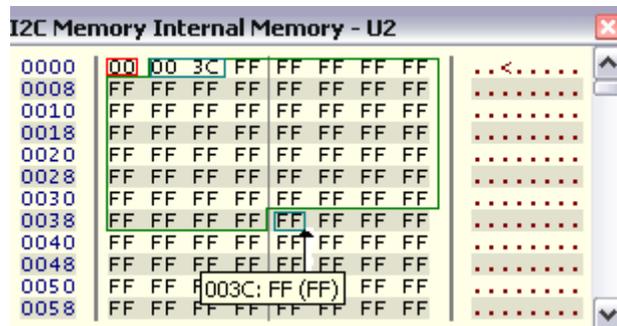
Borrar(), se elimina y reorganiza información de los dispositivos almacenados en la tabla de datos.

**3.1.1 Formato de tabla de datos:** el formato utilizado está inspirado en la estructura interna de un libro, es decir cuando se abre un libro en sus primeras páginas se encuentra el índice o tabla de contenido y en las páginas siguientes se encuentra el contenido del mismo, si se desea consultar la información de un

capítulo específico se busca en el índice y de esta forma ubicar la página en la cual está la información necesitada.

La tabla de datos almacenada en la memoria serial I2C 24LC512 se puede ver en la Figura 18, que corresponde al mapa de memoria de la EEPROM externa cuando el sistema se inicializa y no se ha agregado ningún dispositivo.

**Figura 18.** Formato memoria EEPROM 24LC512



**3.1.1.1 Número de dispositivos:** el recuadro rojo en la Figura 18, localidad 0x0000 en memoria se relaciona con el número de dispositivos y con el índice de la tabla de datos, para este caso tiene un valor de cero, ya que el sistema no tiene agregado ningún dispositivo.

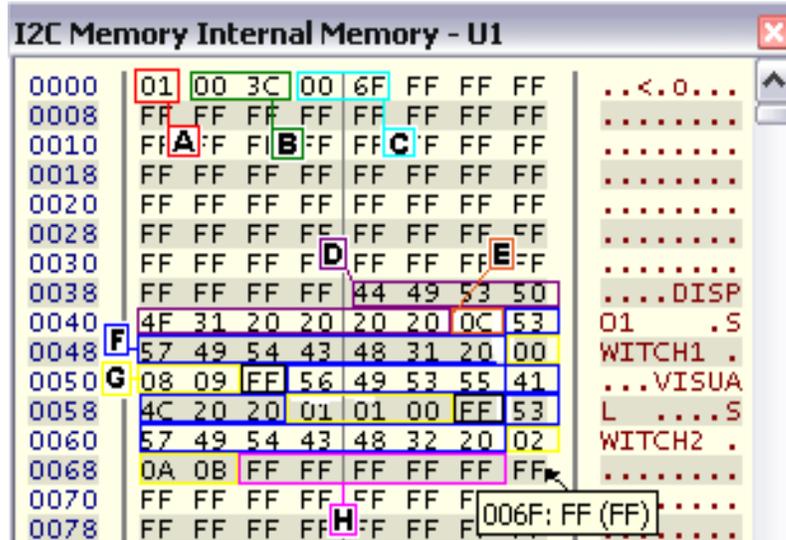
Al momento de agregar el primer dispositivo su valor será de uno (1), ya que indicara en qué dirección se encuentra el byte LSB del índice correspondiente al primer dispositivo. A medida que se añadan dispositivos, el valor almacenado en la localidad 0x0000 se incrementa de dos en dos, debido a que se utilizan dos bytes para almacenar los valores del índice de la tabla de datos.

**3.1.1.2 Índice de la tabla de datos:** espacio de 60 localidades de un (1) byte en memoria EEPROM externa, reservado para almacenar las direcciones donde inicia la tabla de datos correspondiente para cada dispositivo, el formato corresponde a 16 bits (2 bytes) por lo que se prevé un número máximo de 30 dispositivos.

Como se observa en la Figura 18, las localidades 0x0001 y 0x0002 son utilizadas para almacenar la dirección donde debe iniciar la tabla de datos correspondiente al primer dispositivo.

La Figura 19 presenta el mapa de memoria EEPROM externa cuando tiene la información correspondiente a un dispositivo de ejemplo que ha sido agregado

**Figura 19.** Mapa memoria EEPROM actualizada



Como se puede observar (Figura 19) en la dirección 0x003C, señalado con la letra D, se encuentra el inicio de la tabla de datos correspondiente para el dispositivo de ejemplo, la organización de memoria es la siguiente:

**3.1.1.3 Nombre del dispositivo:** las primeras 10 localidades de memoria son reservadas para almacenar el nombre del dispositivo. Como se puede ver en la imagen marcado con la letra D, cada byte corresponde con el código ASCII de cada carácter del nombre. Si se llama n al valor que indica el índice de la tabla de datos para cada dispositivo; el nombre del dispositivo inicia en la dirección de memoria n.

**3.1.1.4 Fuente a conmutar:** es necesario utilizar más de una fuente debido a los diferentes niveles de tensión requeridos por los visualizadores en cada uno de los dispositivos (Mountgrip=28V, AACP=5V). El valor almacenado en la localidad de memoria n+1, marcado con la letra E, corresponde con el valor de la fuente que se debe utilizar para probar los visualizadores de cada modulo, para este ejemplo se puede ver almacenado el valor 0x0C es decir el decimal 12, lo cual indica que los visualizadores del correspondiente dispositivo necesitan 12 voltios para su correcto funcionamiento.

**3.1.1.5 Nombre del contactor:** después del espacio en memoria ocupado por el nombre y el valor de la fuente a conmutar, las siguientes 8 localidades son usadas para almacenar el nombre del contactor, como se puede observar en la Figura 19, marcado con la letra F; El nombre del contactor inicia en la localidad de memoria “m”, donde m=n+9+1, y termina en la localidad de memoria m+8.

**3.1.1.6 Información del contactor:** básicamente se almacenan 3 parámetros para cada contactor de un determinado dispositivos, estos son:

Clase de contactor: corresponde al tipo de prueba que se debe realizar, se almacena en la localidad de memoria m+9, los tipos de clase son los siguientes.

- Clase 0, prueba de contactor activo, se debe detectar que el contactor no esté abierto cuando se activa.
- Clase 1, indicadores visuales, retorno mediante el ojo humano, el cual indicara si enciende o no el indicador.
- Clase 2, prueba de contactor sin activar, se debe detectar que el contactor no esté en corto cuando no se activa.

Salida: corresponde al pin del microcontrolador que se utiliza para enviar la señal de prueba al contactor del dispositivo, se almacena en la localidad de memoria m+10.

Retorno: corresponde al pin del microcontrolador que se utiliza para leer el retorno de la señal en el contactor del dispositivo a probar, se almacena en la localidad de memoria m+11.

Como se observa en la Figura 19, el recuadro marcado con la letra G, en localidad de memoria m+12, después de la información correspondiente a cada contactor se tiene una dirección de memoria vacía para señalar que el siguiente dato corresponde al inicio de un nuevo contactor. También en el recuadro marcado con la letra H, se observa consecutivamente cinco (5) localidades de memoria vacías que indican el fin del dispositivo y el inicio de un futuro nuevo dispositivo que se necesite agregar, si se mira el recuadro marcado con la letra B, el cual tiene almacenado el valor 0x006F el cual corresponde con la dirección de inicio para un futuro nuevo dispositivo, ya que este es parte del índice de la tabla de datos

**3.1.2 Funciones Principales:** la Figura 20 presenta el diagrama para agregar un nuevo dispositivo.

**Figura 20.** Diagrama de flujo para agregar un nuevo dispositivo

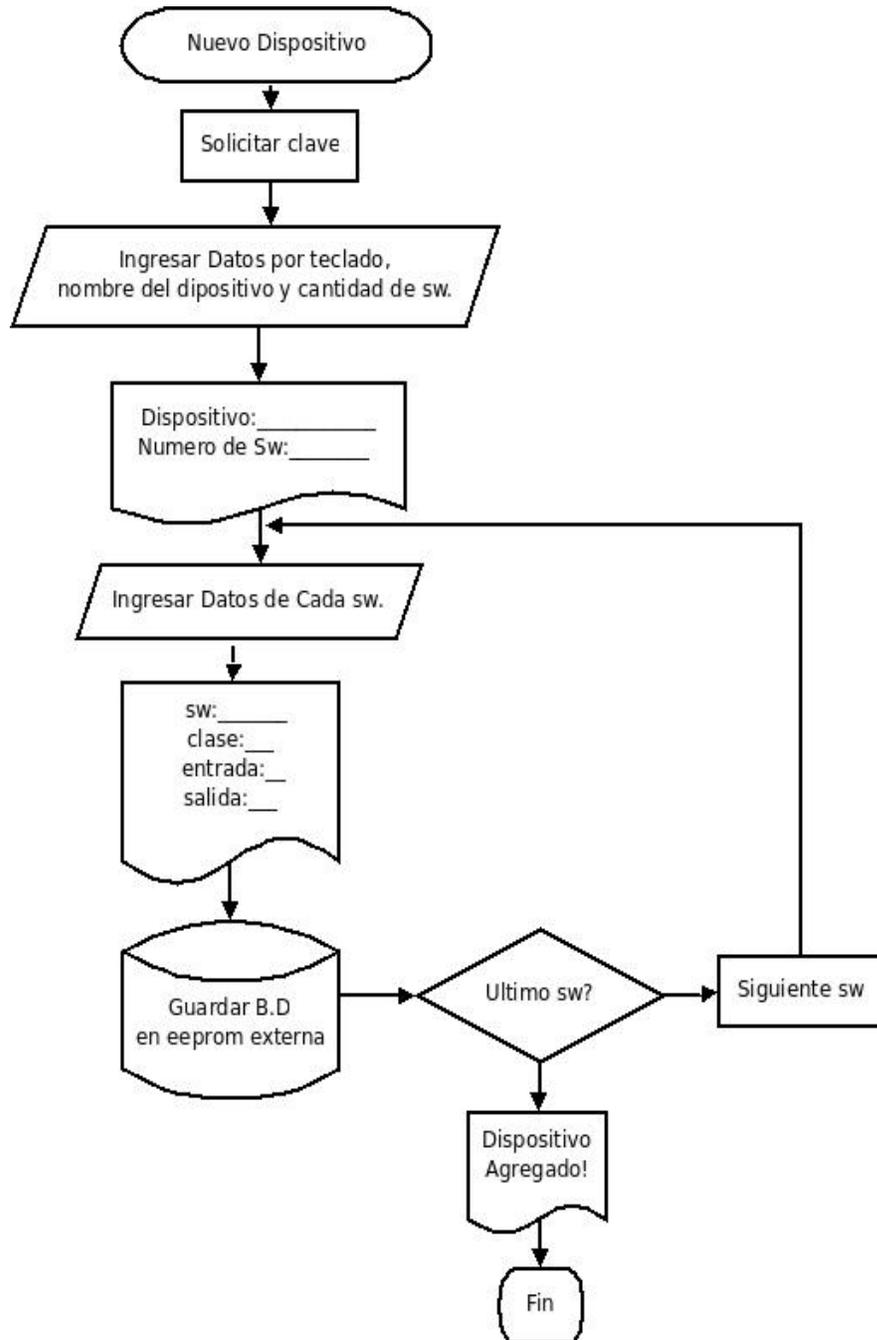
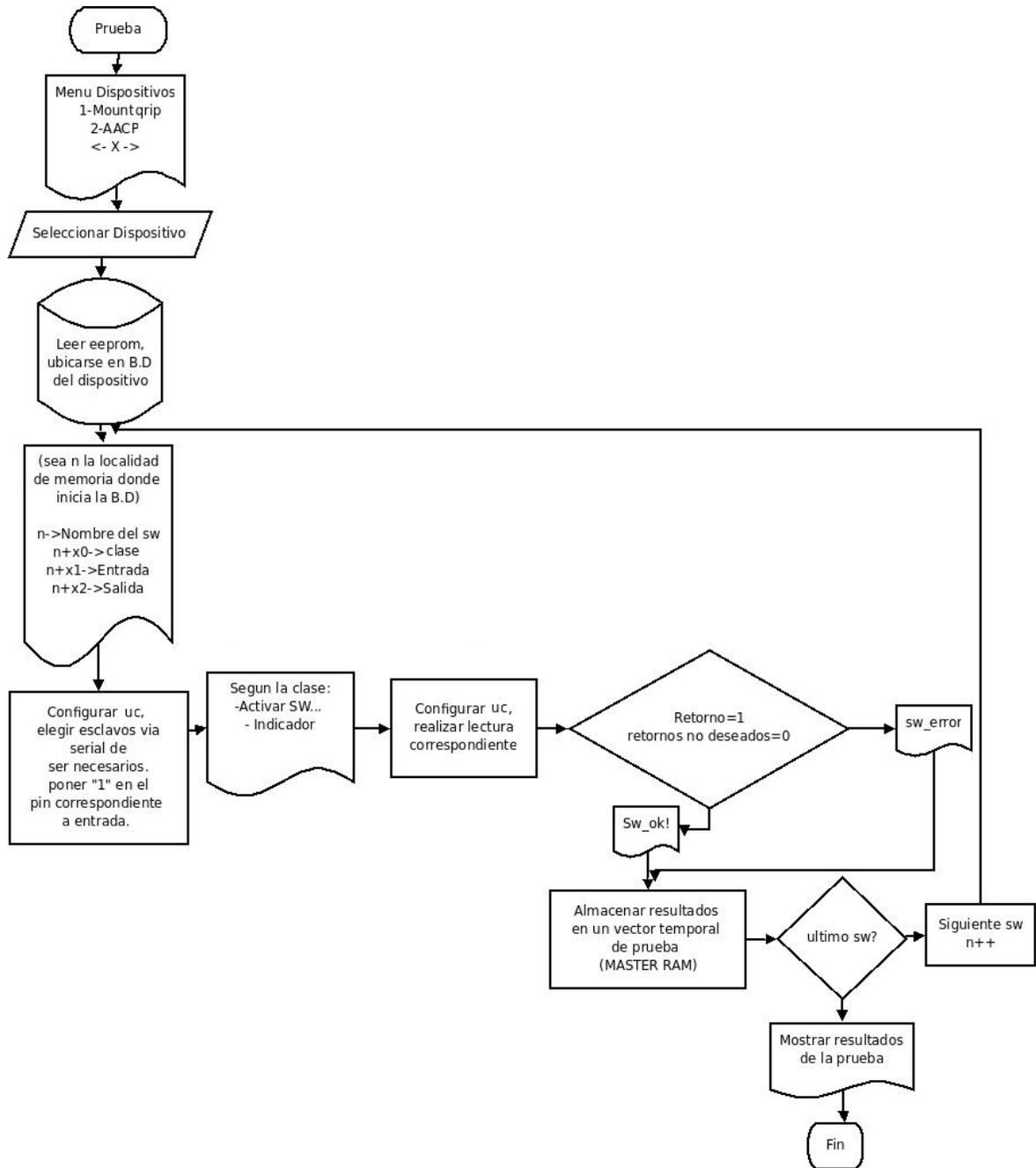


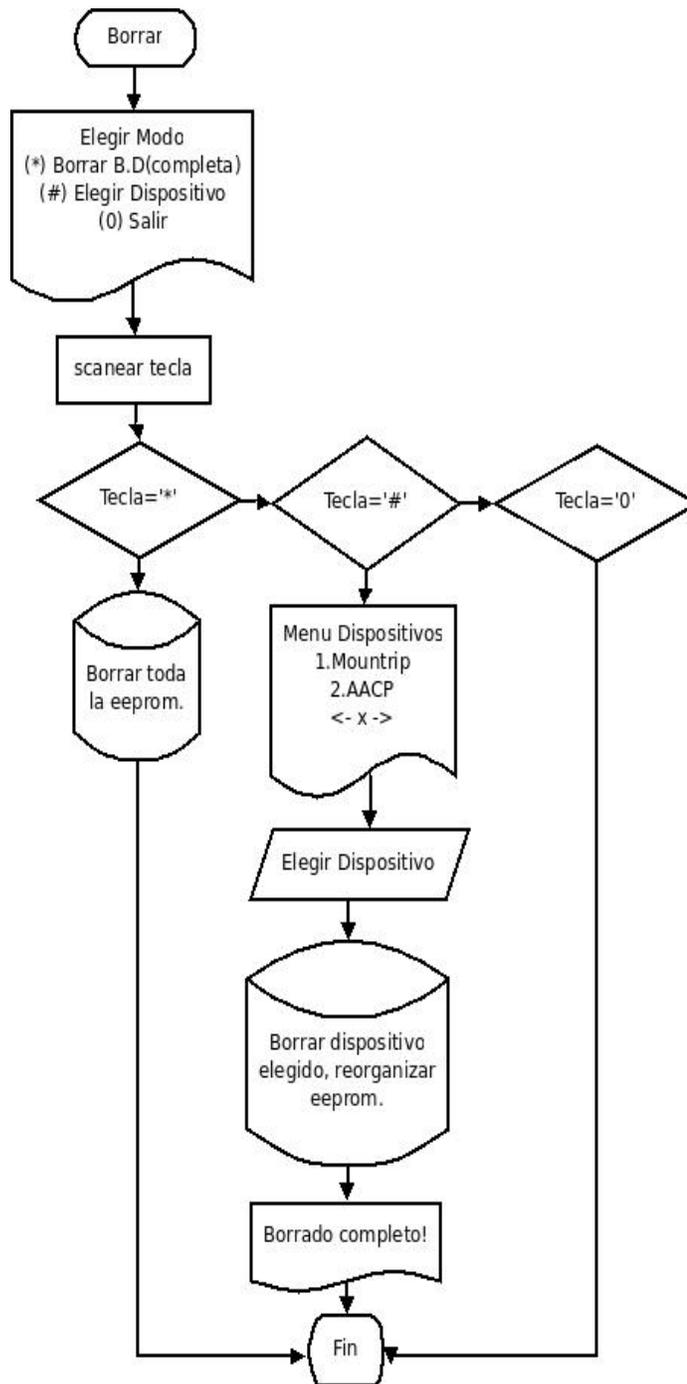
Diagrama de flujo para realizar la prueba de un determinado dispositivo agregado previamente (Figura 21)

**Figura 21.** Diagrama de flujo para realizar prueba



El diagrama de flujo para borrar información de la tabla de datos se observa en la Figura 22.

**Figura 22.** Diagrama de flujo para borrar tabla de datos



Para la codificación de las funciones, el compilador C de Custom Computer Services Incorporated (CCS) fue escogido debido a que permite desarrollar programas en C enfocado a microcontroladores de Microchip, con las ventajas que supone tener un lenguaje desarrollado específicamente para un microcontrolador,

su facilidad de uso, su claro entorno de trabajo y la posibilidad de compilar en las tres familias de gamas baja, media y alta, le confieren una versatilidad y potencia muy elevadas.

**3.1.3 Librerías utilizadas:** cada una de estas funciones hace uso de librerías, algunas proporcionadas por el software CCS PICC, otras creadas para el desarrollo del proyecto, en dichas librerías se encuentran procedimientos para manejo de periféricos y demás tareas que son usadas con regularidad.

**3.1.3.1 Funciones y procedimientos básicos:** procedimientos que son llamados con frecuencia por otras funciones, en la librería se encuentran las siguientes funciones:

- Función para esperar hasta adquirir tecla presionada, en la variable global k se encuentra el caracter, y en x el valor numérico de tipo int:

```
x= tecla_k() ;
```

- Función para escribir números: si se desea borrar un número, se lleva a cabo presionando la tecla #.

```
void escribir(char limit,int parada)
```

*limit* : carácter correspondiente con la tecla para dejar de escribir.

*parada*: entero correspondiente a la longitud limite de números que se escribirá.

- Funciones para escribir y leer un dato en la EEPROM I2C 24LC512, estas funciones se encuentra en uno de los ejemplos del uso del compilador CCS PICC.

Para escribir un byte en la EEPROM externa:

```
void write_ext_eeprom(long int address, BYTE data)
```

*address*= localidad de memoria en la que se desea almacenar la información.

*data*= Dato de 8 bits que se necesite almacenar.

Para leer un byte de la EEPROM externa:

```
BYTE J = read_ext_eeprom(long int address)
```

- Las siguientes funciones son usadas para escribir en la memoria externa I2C. A diferencia de las anteriores, estas escriben y leen datos en formato de 16 bits, es decir usa dos bytes de memoria para el almacenamiento de un solo dato, como consecuencia es un poco más lenta, aunque humanamente imperceptible y hace uso de las funciones anteriores para escribir y leer un byte en memoria.

```

write_eeprom_16b(long int address, long* val)
long int U= read_eeprom_16b(long int direccion)

```

**3.1.3.2 Multifuncionalidad del teclado matricial:** librería que permite escribir letras y números mediante la pulsación repetitiva de una tecla en un teclado matricial, para lograr esto se hace uso del timer 1 y la interrupción por desbordamiento.

A Las teclas del 1-9 se le asignan 3 letras y el correspondiente valor numérico, la tecla 0 funciona como “barra espaciadora” y las teclas '\*' y '#' tienen funciones específicas, de la siguiente forma:

**Cuadro 14.** Distribución multifuncional del teclado matricial.

Tecla	Caracteres / Funciones				
1	A	B	C	1	
2	D	E	F	2	
3	G	H	I	3	
4	J	K	L	4	
5	M	N	O	5	
6	P	Q	R	6	
7	S	T	U	7	
8	V	W	X	8	
9	Y	Z	-	9	
0	Espacio				0
#	Borrar, desplazar ->				#
*	Validar, desplazar<-				*

Se debe tener en cuenta la frecuencia de trabajo, para el caso la frecuencia de cristal utilizada fue de 20Mhz. Es necesario habilitar interrupciones globales y de timer 1 en la rutina principal (main) del programa.

El procedimiento se basa en incrementar la coordenada en x en la cual se escribirá el próximo carácter una vez pasado un determinado tiempo sin que se haya pulsado una tecla o cuando una tecla distinta es presionada

```

letras(char limit,int cord_x,int cord_y,int stop);

```

*limit=caracter para parar de escribir, por ejemplo '\*'.*  
*cord\_x,cord\_y=cordenadas en la lcd para iniciar a escribir.*  
*stop=numero de caracteres limites que se pueden escribir.*

**3.1.3.3 Manejo de cadenas de texto en memoria serial I2C y LCD:** esta librería permite guardar y leer cadenas de texto, bien sea que la información se encuentre en memoria externa o en pantalla LCD.

Se debe definir la siguiente variable como global:

```
char texto[9];
```

- Función para almacenar cadena de texto en eeprom:

```
guardar_eeprom_texto(long int ADD,int W, char cadena[]);
```

*ADD: dirección de eeprom a guardar el primer caracter de la cadena*

*W: longitud de la cadena.*

*cadena[]: cadena que se desea guardar.*

- Función para Leer cadena de texto desde memoria externa, la lectura se almacena en la variable `texto[]`:

```
leer_eeprom_texto(long int ADD, int W);
```

*ADD: dirección en memoria a leer el primer caracter de la cadena*

*W: longitud de la cadena.*

- Función para leer cadena de texto en lcd:

```
leer_lcdtext(int x,int y,int W);
```

*x,y: coordenadas en la lcd donde inicia la cadena.*

*W: longitud de la cadena.*

## **3.2 HARDWARE**

El sistema se diseñó teniendo en cuenta las normas MIL-STD-202-G, donde se establecen “*los métodos uniformes para comprobar los componentes electrónicos y eléctricos, entre los que incluye pruebas ambientales para determinar la resistencia a los efectos nocivos de los elementos naturales y las condiciones que rodean las operaciones militares, y física y pruebas eléctricas. A los efectos de esta norma, el término "los componentes" incluye artículos tales como condensadores, resistencias, interruptores, inductores y otros.*”<sup>2</sup>

---

<sup>2</sup> Department Of Defense Test Method Standard Electronic and Electrical Component Parts. Norma MIL-SRD-202, <http://www.dscc.dla.mil/Downloads/MilSpec/Docs/Mil-Std-202/std202.pdf>, pag.7

### 3.2.1 Partes del sistema:

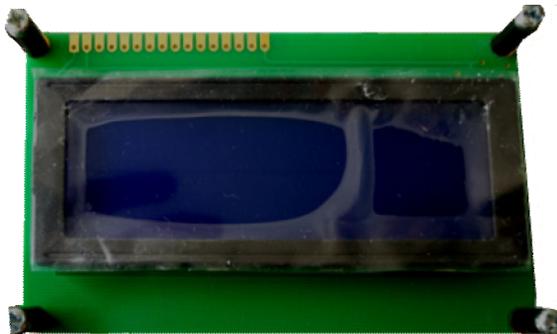
- Maletín industrial: se implementó un maletín que soportara ambientes de trabajo fuertes; este maletín que se observa en la Figura 23, está hecho de policarbonato, el cual es resistente a diferentes ambientes de trabajo, asegurando así que no se maltraten los materiales que guarda en su interior, cumpliendo con ello la norma anteriormente mencionada.

**Figura 23.** Maletín industrial



- LCD y Teclado: para la interacción con el microcontrolador se empleó un LCD de 20 columnas por 4 filas, en el cual se visualizan todos los eventos del sistema y un teclado de 4 columnas por 4 filas con el cual se ejecutan los procesos que corren en el mismo (Figuras 24, 25).

**Figura 24.** LCD 20x4

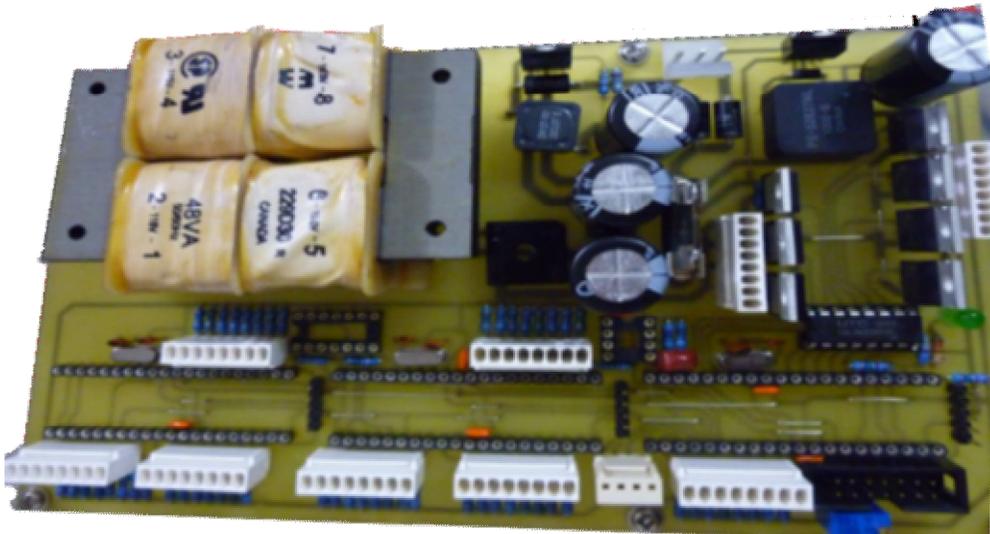


**Figura 25.** Teclado 4x4



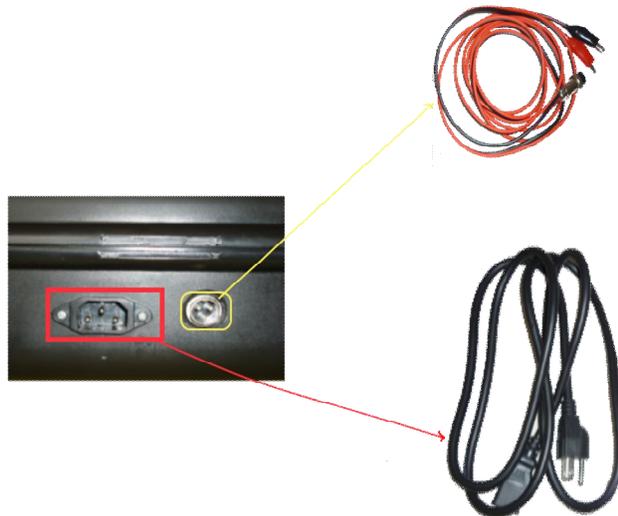
- Circuito impreso: aquí se puede visualizar la placa madre del sistema. Como se puede observar hay una etapa de potencia, rectificación, transistores para regulación de los niveles de voltaje, Mosfet que disparan las señales análogas (visualizadores de alto voltaje) y los sockets para la red serial de los microcontroladores. (Figura 26)

**Figura 26.** Circuito impreso



- Alimentación del sistema: el sistema cuenta con dos formas para energizar el circuito impreso. En la Figura 27 se puede observar que el cable negro es para conectar a la red 120VAC para trabajar en el taller de mantenimiento y el otro cable es para conectar directamente a la batería de la aeronave.

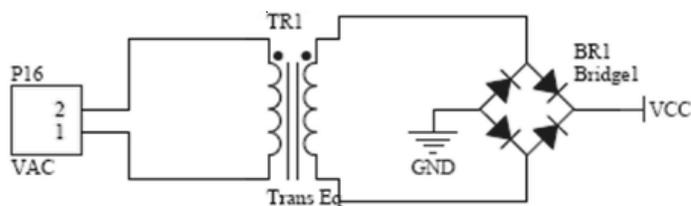
**Figura 27.** Formas de energizar



### 3.2.2 Descripción por etapas

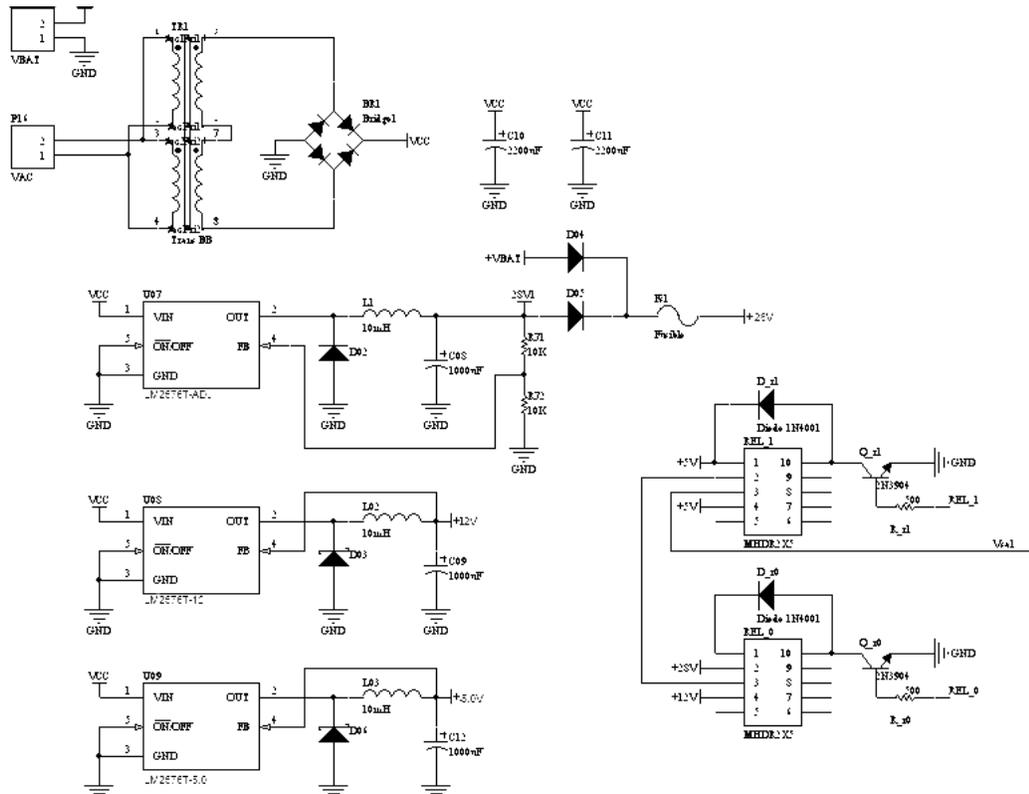
- Etapa de potencia: se utilizó un transformador de 48VA, que recibe la señal AC 120-110VA. En la Figura 28 se pueden ver sus conexiones:

**Figura 28.** Etapa de potencia



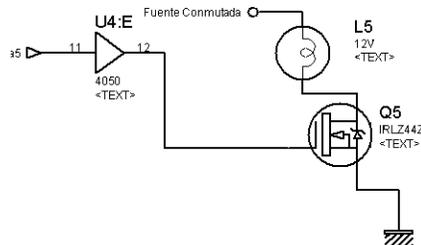
- Etapa de alimentación: con la señal DC tomada del puente rectificador (Figura 29), era necesario reguladores de voltaje que permitieran estabilizar la señal 5V para alimentar los microcontroladores y los voltajes ajustables 5V, 12V y 28V para los visualizadores de los dispositivos a diagnosticar. El regulador usado es el LM2576; este permite diseñar la fuente conmutada que se requiere para las pruebas de los elementos visualizadores a 5V, 12V y 28V, entregando un máximo de 3A que es lo que el fabricante garantiza. En la Figura 28 se ven las conexiones respectivas.

**Figura 29.** Etapa de alimentación



- Etapa de señales de prueba: para enviar las señales de alto voltaje se utilizó un mosfet IRLZ44, que al recibir la señal del microcontrolador activa el visualizador que esté conectado mediante el cable de conexión, a los extremos de la fuente conmutada y el mosfet, como se puede ver en la Figura 30.

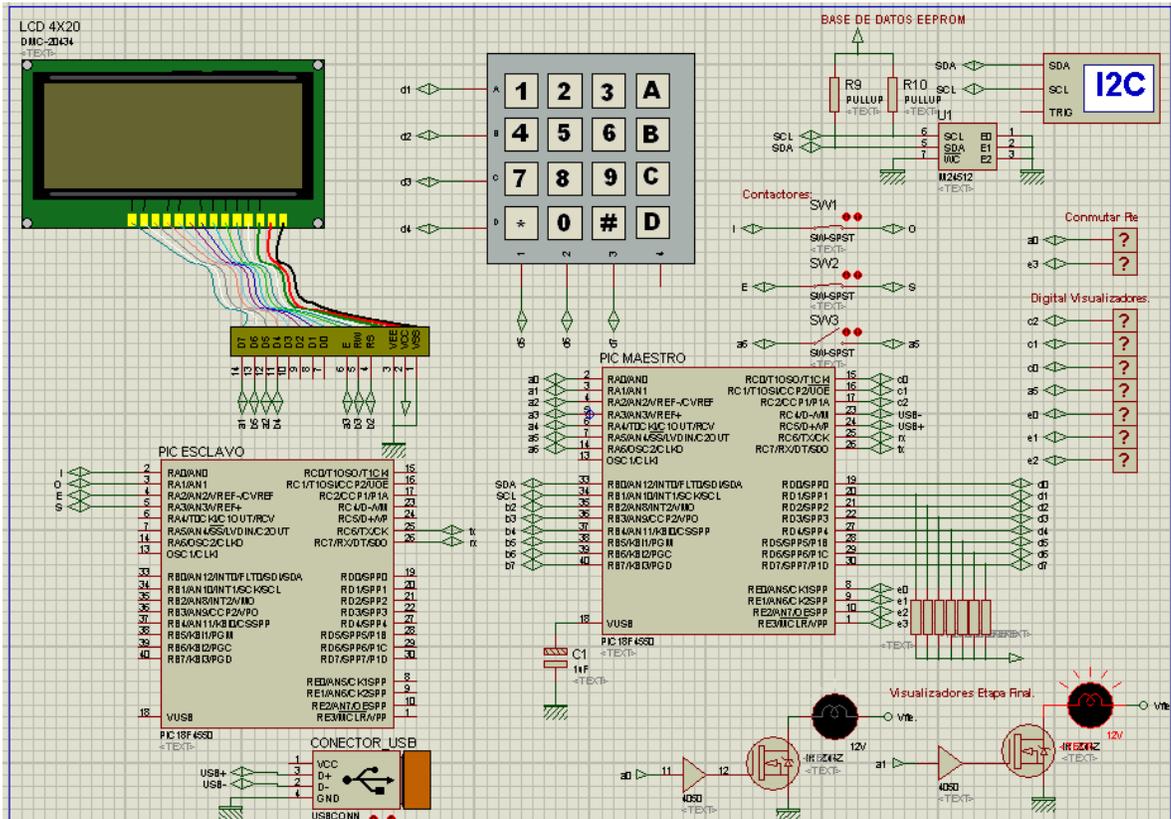
**Figura 30.** Etapa señales análogas



El buffer que se muestra en la Figura 30, permite controlar el manejo de corriente asegurando la activación de mosfet.

En la Figura 31 se puede observar el diagrama esquemático de la red microcontroladores maestro/esclavo, el teclado, LCD y EEPROM.

Figura 31. Señales digitales



**3.2.3 Fabricación cables de conexión:** para realizar el cable de conexión se usó un tipo de cable que satisface las normas MIL-C-17-G<sup>3</sup>. La referencia RG188 A/U<sup>4</sup> cumple con las especificaciones, cubre los cables flexibles y semirrígidos con núcleos de dieléctrico sólido y semisólido, con conductores internos individuales, dobles resistente al fuego. Los cables eléctricos conectados a los Amphenol están recubiertos por una malla metálica que protege el cable eléctrico en zonas explosivas (Figura 32).

<sup>3</sup> <http://www.wbdg.org/ccb/FEDMIL/c17g.pdf>

<sup>4</sup> RG-Cables acc. to MIL-C-17F and MIL-C-17G, [www.draka.co.th/en/Communication/.../RG%20Cables/RG188AU.pdf](http://www.draka.co.th/en/Communication/.../RG%20Cables/RG188AU.pdf)

**Figura 32.** Funda de protección



**Fuente:** <http://www.directindustry.es/prod/ad-vigano/funda-de-proteccion-para-cable-electrico-en-zonas-explosivas-25422-137822.html>

En la Figura 33 se puede ver los cables de conexión terminados, tienen aproximadamente 1 metro de longitud y sus conectores Amphenol hembra en sus extremos.

**Figura 33.** Cables de Conexión terminados



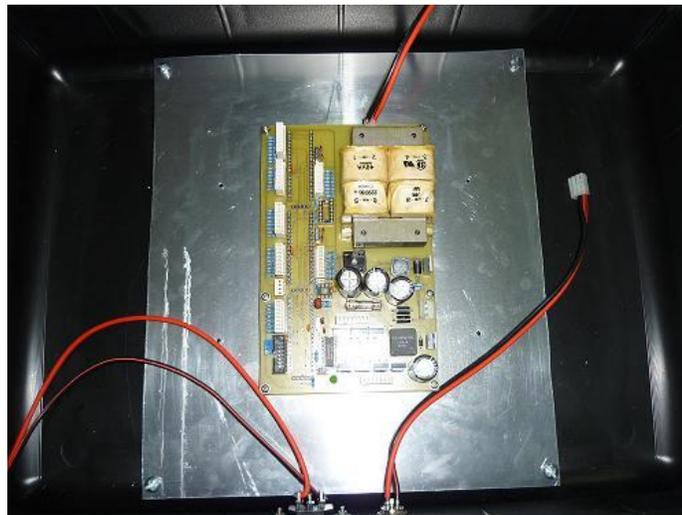
Los conectores MIL-DTL-38999 se implementaron para cumplir los requisitos más rigurosos de la industria militar y aeroespacial; algunas de las características de los conectores Amphenol son:

- Blindaje EMI (Materiales absorbentes ó láminas metálicas con adhesivos conductivos para absorber, disminuir y/ó atenuar la interferencia causada por electromagnetismo ó la generada por radiofrecuencia, eliminando la necesidad de sujetadores mecánicos).
- Resistencia de humedad: además del sello diedro, el souffler común principal y la junta posterior en el enchufe se fijan para el lacre de la humedad entre las mitades del conector

- Resistencia a la corrosión.
- Cadmio monótono verde oliva sobre el niquelado en la cáscara de aluminio (Resiste 500 horas de exposición del aerosol de sal) o el níquel no electrolítico

**3.2.4 Ensamblando el sistema:** la Figura 34 muestra el circuito impreso sobre el maletín, fijada por tornillos a una placa metálica

**Figura 34.** Circuito impreso sobre el maletín



Las Figuras 35-36 muestran el sistema de diagnostico portable puesto en marcha.

**Figura 35.** LCD y Teclado ensamblados



**Figura 36.** Sistema de diagnóstico portable ensamblado



## 4. APLICATIVO INTERFAZ USB

El puerto paralelo y puerto serie ya están obsoletos, por tal motivo se quiso trabajar con el puerto USB que es el más común en los computadores portátiles y de escritorio actuales.

Estudiando acerca de qué plataforma implementar para el desarrollo del aplicativo, se decidió trabajar sobre JAVA por su versatilidad y eficiencia, pero el lenguaje java se complica cuando se quiere acceder a las librerías dinámicas (dll). En los foros como todopic, unicrom entre otros se puede encontrar mucha documentación para solucionar problemas, esto hace parte de las ventajas que tiene programar sobre esta plataforma; es muy implementada por estudiantes universitarios en el desarrollo tecnológico.

Para programar en lenguaje java, fue necesario utilizar un entorno que permitiera trabajar con objetos, haciendo más fácil la programación de la aplicación; encontrando el software NetBeans el más adecuado. NetBeans es un entorno de desarrollo integrado (IDE) para Java, parecido a la de Delphi. Xelfi que fue el primer entorno de desarrollo integrado escrito en Java.

### 4.1 NetBeans IDE

Para facilitar el lenguaje JAVA se hizo uso de un entorno que permitió el manejo de objetos, NetBeans IDE es un entorno de desarrollo visual de código abierto para aplicaciones programadas mediante Java.

La programación mediante NetBeans se realizó a través de componentes de software modulares, también llamados módulos. NetBeans cuenta con una página web que pone al servicio de los usuarios muchos de estos módulos para realizar diferentes aplicaciones no solo para escritorio, también para páginas web.

Para que NetBeans funcione es necesario instalar previamente el software de java que es gratuito, se puede descargar de la red fácilmente para trabajar con él. También se debe instalar un kit de desarrollo JDK disponible en la red (<http://netbeans.org/>), con estos dos instaladores se puede empezar a crear el aplicativo.

## 4.2 LIBRERIA jPicUSB<sup>5</sup>

La librería jPicUSB facilita la comunicación entre el microcontrolador y el computador. JPicUSB es una clase java, que mediante el uso de interfaces nativas (JNI), permite a una aplicación Java hacer llamados a una librería dinámica, en este caso (jpicusb.dll). JPicUSB.dll es una librería que implementa todas las funciones de la API USB de Microchip (mpusbapi.dll), con la diferencia de que está especialmente recompilada para permitir que la clase jPicUSB haga llamados a sus funciones.

### 4.2.1 Ficheros que componen al paquete jPicUSB: ficheros que componen al paquete jPicUSB

- jpicusb.jar – Interfaz java con la librería dinámica jpicusb.dll. Esta debe agregarse a la librería del archivo creado para que se ejecute correctamente.
- jpicusb.dll – Librería dinámica que contiene todas las funciones de la API USB de Microchip, que debe ser instalado previamente en el PC.
- javadoc – Carpeta con la documentación de la clase. (Descripción de todos los métodos). Es de mucha ayuda, orienta de forma correcta sobre la función de los comandos.

### 4.2.2 Métodos principales disponibles en jpicusb.jar: en la clase iface del paquete jPicUSB se distinguen 3 tipos de métodos como son

- Métodos de inicialización varios: Cargan la librería dinámica y establecen valores por defecto.  
**static void load( )**: carga la librería dinámica jpicusb.dll. Este método debe ser llamado solo una vez, antes de cualquier llamada a otro método.
- Métodos generales: Lectura y Escritura de datos a un dispositivo USB, apertura de una conexión, etc.  
**static void set\_vidpid(java.lang.String s)**: establece el VID&PID que Microchip provee para el PC
- Métodos rápidos: Imitan a los métodos generales, pero con menos código para agilizar el desarrollo.  
**static void set\_instance(int i)**: establece la instancia utilizada por defecto en toda llamada a un método rápido.

---

<sup>5</sup> Oñativía, Gerónimo Isidro. Universidad Nacional de Tucumán – Facultad de Ciencias Exactas – Ingeniería en Computación. geronimox@gmail.com - desarrollos@divideandconquer.com.ar - www.divideandconquer.com.ar

### 4.3 SOFTWARE DE LA APLICACIÓN

El programa principal contiene todos los objetos necesarios para la aplicación.

Al implementar la comunicación USB la transmisión se realiza por envío y recepción de paquetes de datos; todos los comandos (Public byte) son para iniciar la comunicación entre el microcontrolador y computador (Figura 37).

Comandos:

```
public byte cmd_enviarDatos = 88;  
public byte cmd_recibirRespuesta = 98;  
public byte cmd_CERRAR = 87;
```

**Figura 37.** Comunicación microcontrolador y PC

```
40 public Datos() throws Exception(  
41     try (  
42         iface.load();  
43     } catch (Exception e) (  
44         System.out.println(e.getMessage());  
45     )  
46     mdbc=new MyDBConnection(); //Inicializo la variable de  
47     mdbc.init();//Realizo la conexion  
48     Connection conn=mdbc.getMyConnection(); //Obtengo la cc  
49     stmt= conn.createStatement(); //Inicializo la variable  
50     //configuramos el vid_pid del dispositivo asi podemos t  
51     iface.set_vidpid("vid_04d8&pid_000b");  
52     //configuramos la instancia del dispositivo asi podemos  
53     iface.set_instance(0);  
54     initComponents(); //inicializo los componentes, este  
55     inicializarventanas(); //Funcion para iniciar la posici  
56     crearBaseDatos();//Funcion q envia comandos a mysql y c  
57     inicializarComboBox();//Funcion para iniciar el menu de  
58 }
```

Las líneas de código de la Figura 38 son las explicadas en el javadoc; QWrite permite escribir un dato con cierta cantidad de paquetes a una velocidad establecida. QrRead permite leer lo que el microcontrolador envía.

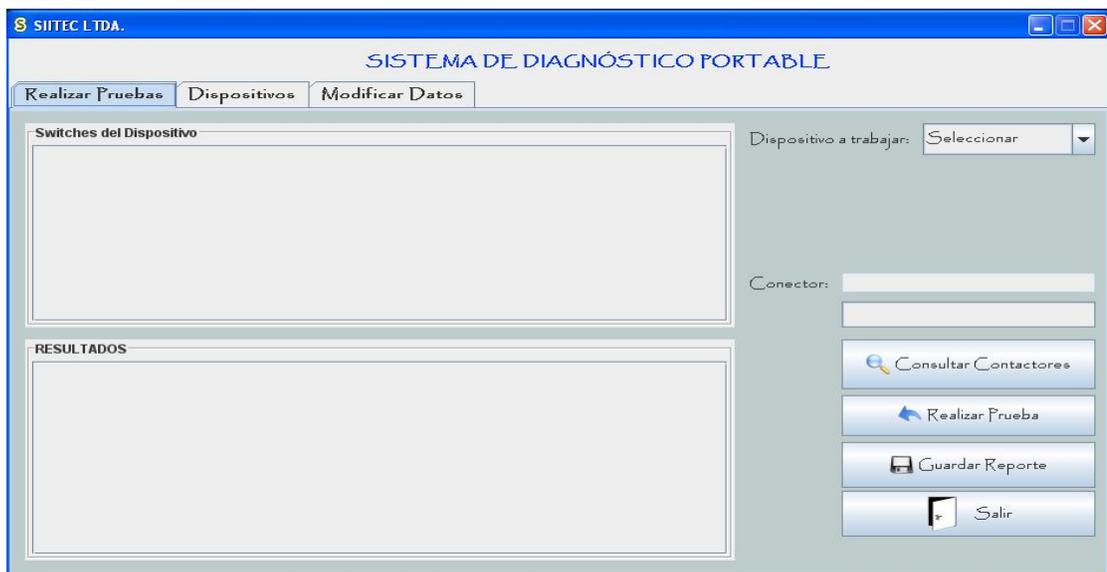
Figura 38. QRead-QWrite

```
Fuente  Diseño  [Icons]
66      }
67
68      public void obtenerResultado(byte command) {
69          byte[] out = {command};
70          iface.QWrite(out, 1, 1000);
71      }
72
73      public void send_command(byte command, byte param1, byte param2) {
74          byte[] out = {command, param1, param2};
75          iface.QWrite(out, 3, 1000);
76      }
77
78      public byte[] read_response(int maxbytes) {
79          return iface.QRead(maxbytes, 1000);
80      }
81
82      private void enviar_datos(String entradaS,String salidaS) {
83          byte in = Byte.valueOf(entradaS);
84          byte out = Byte.valueOf(salidaS);
85          send_command(cmd_ESC, in, out);
86      }
87
```

#### 4.4 RESULTADOS DEL APLICATIVO

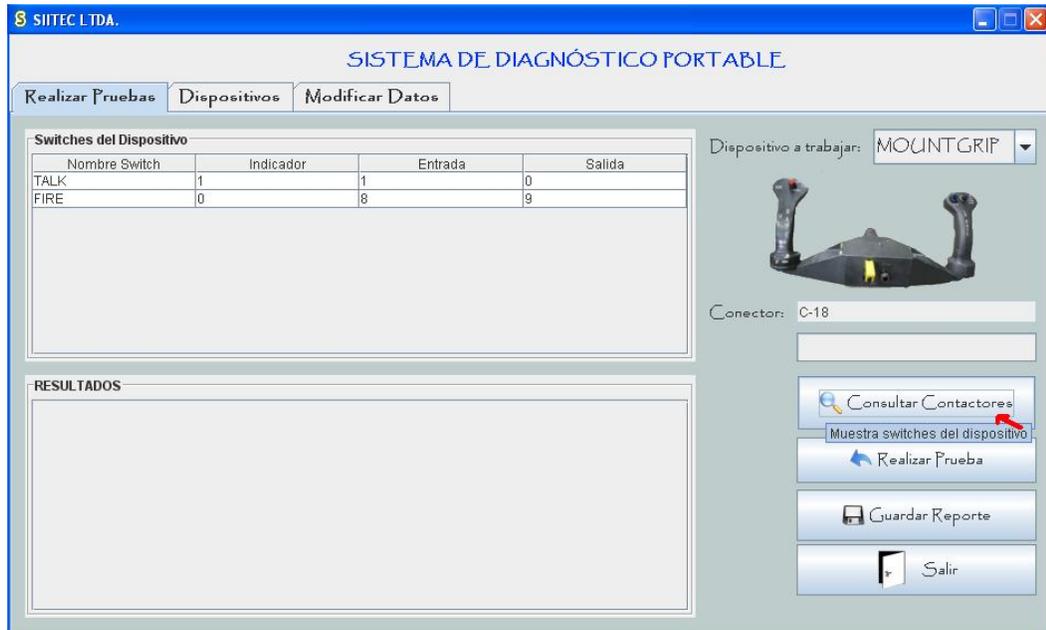
La aplicación creada ejecuta tres (3) procesos como son: realizar prueba, agregar dispositivos y modificar datos. La idea para crear la aplicación, era mantener similitud con el menú que se tiene de forma independiente con LCD y teclado, con la diferencia que al digitar un nuevo dispositivo la presentación es más fácil e interactivo. En la Figura 39 se puede ver el panel principal, en el se observa la imagen del dispositivo a diagnosticar.

Figura 39. Panel principal



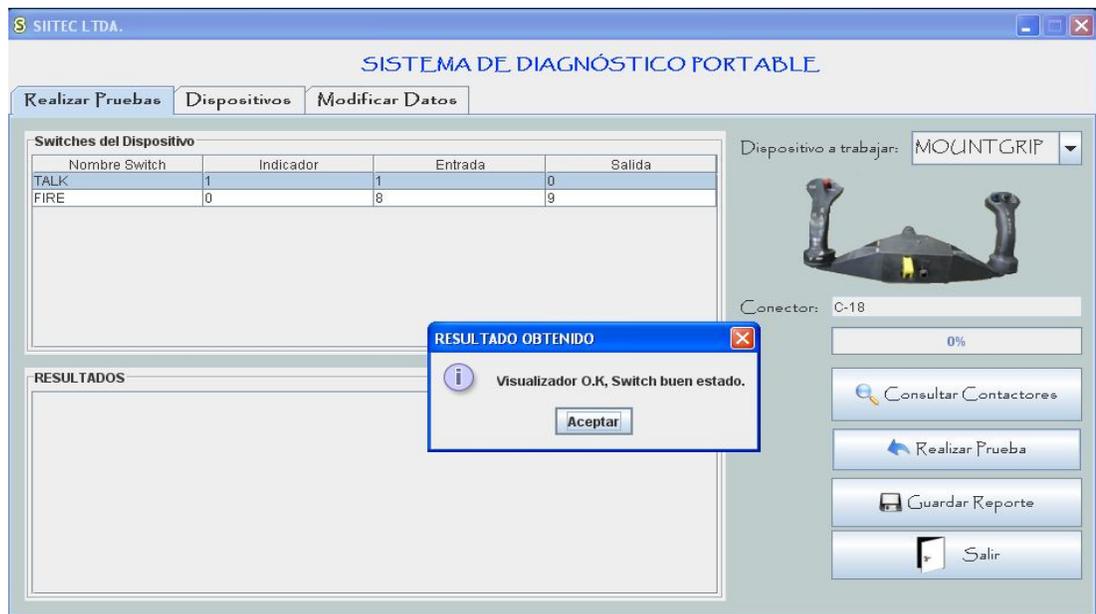
El programa funciona al dar click en el botón de consultar contactores, este botón permite visualizar la tabla con la cantidad de contactores que pertenecen a este dispositivo, como se ve en la Figura 40 al elegir el contactor que se desea diagnosticar, se presiona el contactor seleccionado y se da click en el botón realizar prueba, obteniendo la respuesta del estado del contactor.

**Figura 40.** Selección de contactor



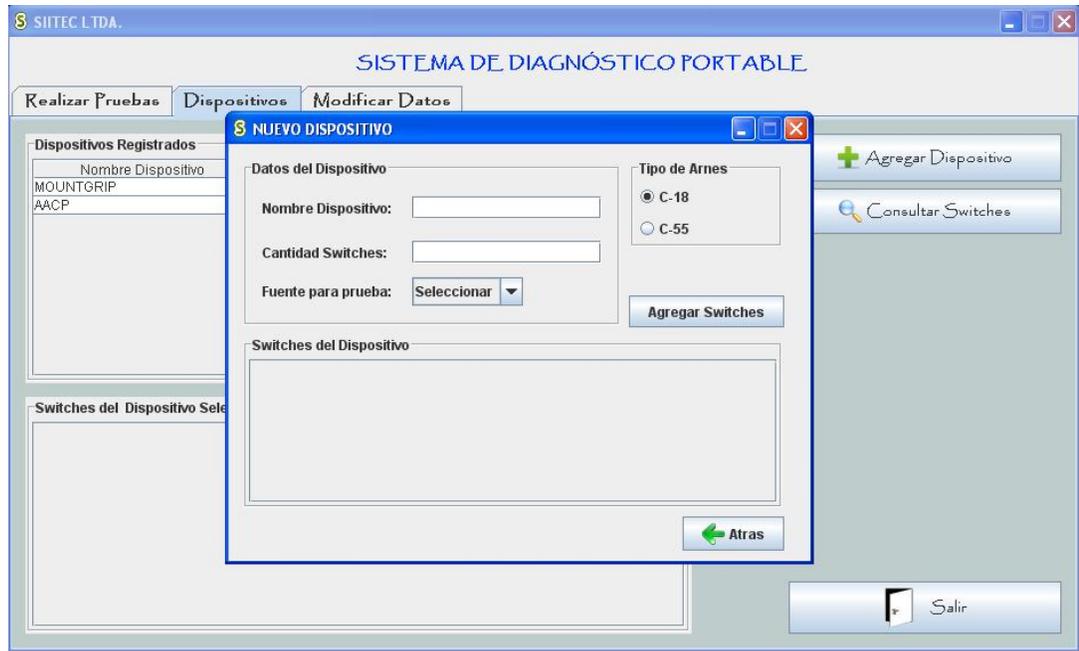
En la Figura 41 se puede ver el resultado del diagnóstico de un visualizador

**Figura 41.** Resultado de un contactor



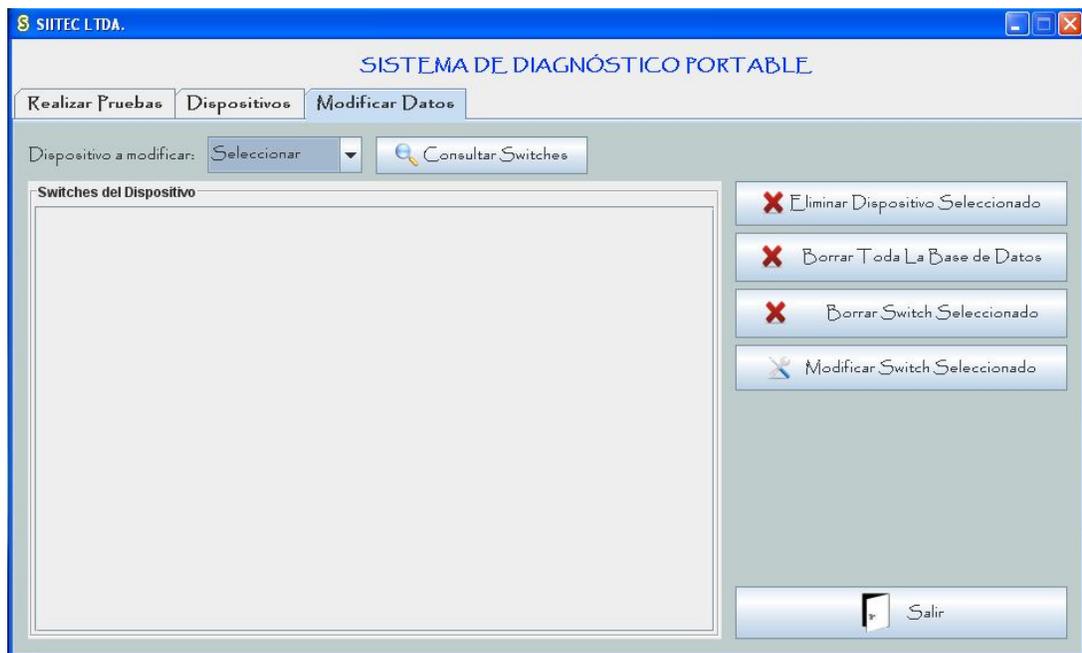
La Figura 42 muestra el panel para agregar dispositivos

**Figura 42.** Agregar dispositivo



La Figura 43 muestra el panel para borrar datos, con esta opción de menú se puede editar el nombre de un contactor, eliminar un dispositivo completamente.

**Figura 43.** Borrar tabla de datos



## CONCLUSIONES Y RECOMENDACIONES

Las pruebas de protocolos de seguridad son de gran utilidad y se debe minimizar el error de lectura en la realización de estas pruebas, para lo cual se desarrolló el Sistema de Diagnóstico Portable.

La memoria serial EEPROM 24LC512 en la cual se almacena la Tabla de Datos de los dispositivos, según su fabricante (Microchip) tiene una retención de datos mayor a 200 años; sin embargo, debido a factores externos y ambientales, se recomienda cambiar la memoria serial cada 20 años.

Para lograr una óptima comunicación entre el microcontrolador y el computador es necesario un correcto ajuste entre la frecuencia interna de reloj y la frecuencia a trabajar en el módulo USB del microcontrolador, bien sea a baja velocidad o a alta velocidad. Para el proyecto se escogió velocidad en módulo USB 2.0 a 48MHZ y con el uso de un cristal de 20MHZ y el ajuste de los prescaladores se configura la frecuencia interna de reloj a 48MHZ.

El desarrollo del proyecto brindó conocimiento en elementos que se utilizan en aeronáutica, como son los bombillos visualizadores, el cableado eléctrico que se utiliza para las conexiones internas de los dispositivos que manejan una electrónica avanzada, conectores Amphenol entre otros, los cuales suplen las necesidades de la Fuerza Aérea Colombiana dando una visión más amplia en el campo laboral, porque no solo existen empleos para el ingeniero electrónico en el área del petróleo o telecomunicaciones, sino también en el desarrollo de proyectos que suplan las necesidades de los colombianos.

Hay muchas ideas y proyectos a desarrollar, solo basta con sembrar y motivar al estudiantado para la creación de nuevos dispositivos, sin necesidad de recurrir a las creaciones extranjeras, que acarrear altos costos. Aplicando reingeniería a muchos de los aparatos que se cree no entender, se comprende su funcionamiento, para con ellos realizar uno nuevo y mejorado.

La teoría permite anticipar a unos resultados; pero, la práctica enseña a ser pacientes para la obtención de esos resultados. Como novatos en un proyecto de tal magnitud, el realizar las primeras pruebas trajo mucho desánimo, nada funcionaba; pero, con perseverancia y dedicación se lograron los objetivos planteados, trayendo consigo mucha experiencia, conocimiento y frutos para un futuro.

La experiencia de programar con Java sobre la plataforma NetBeans IDE fue satisfactoria, este software gratuito orientado a objetos facilita la programación; su sintaxis es similar al lenguaje C. Además de las ventajas de una presentación y entorno amigable.

## BIBLIOGRAFIA

### Libros

BREIJO GARCIA Eduardo. Compilador C CCS y simulador PROTEUS para Microcontroladores PIC. Alfaomega Grupo Editor, Mexico, junio de 2008.

DEITEL M. Harvey y DEITEL J. Paul. Como programar en JAVA. Pearson Prentice Hall. Quinta edición

### Enlaces electrónicos

Compilador C CCS, [www.ccsinfo.com](http://www.ccsinfo.com)

Conectores cilíndricos Serie III, <http://www.btcelectronics.com/documents/sealtron38999.pdf>, Consultado el 3 de Febrero de 2010.

Department Of Defense Test Method Standard Electronic and Electrical Component Parts. Norma MIL-SRD-202, <http://www.dscc.dla.mil/Downloads/MilSpec/Docs/Mil-Std-202/std202.pdf>, Consultado el 29 de julio de 2010.

Microchip products and Devices. Microcontrolador PIC18F4550, <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010300>. Consultado el 20 Febrero de 2010.

Microchip Products and Devices. Datasheet de memoria serial 24LC512, <http://ww1.microchip.com/downloads/en/devicedoc/21754g.pdf>. Consultado el 15 de Abril del 2010

Militar Specification. Cable, Radio Frequency and semirigid. Norma MIL-C-17-G. <http://www.wbdg.org/ccb/FEDMIL/c17g.pdf> . Consultado el 15 de Julio de 2010

NetBeans IDE 6.8, <http://netbeans.org/kb/>

Labcenter Electronics, Junio 2009. Simulador Proteus ISIS. Consultado el 10 de Agosto de 2009. <http://www.labcenter.com>

OÑATIVIA, Gerónimo. Universidad Nacional de Tucumán – Facultad de Ciencias Exactas – Ingeniería en Computación. [geronimox@gmail.com](mailto:geronimox@gmail.com) - [desarrollos@divideandconquer.com.ar](mailto:desarrollos@divideandconquer.com.ar) - [www.divideandconquer.com.ar](http://www.divideandconquer.com.ar). Consultado el 4 Julio de 2010.

## ANEXOS

**SISTENA DE DIAGNÓSTICO PORTABLE**

**MANUAL DEL USUARIO**

## CONTENIDO

### Presentación del Sistema de Diagnóstico Portable (S.D.P.).

#### Inicio rápido

1. Requisitos para utilizar el S.D.P.
2. Como usar el Sistema de Diagnóstico Portable (S.D.P.)
  - 2.1 Realizar pruebas con el S.D.P.
  - 2.2 Agregar nuevo dispositivo al S.D.P.
  - 2.3 Borrar Tabla de datos en memoria del S.D.P.
3. Consejos y soluciones de problemas.
4. Soporte técnico.

## PRESENTACION DEL SISTEMA DE DIAGNÓSTICO PORTABLE

El sistema de diagnóstico portable para dispositivos modulares de aeronaves colombianas es el siguiente:

Figura 1. Sistema de diagnóstico portable



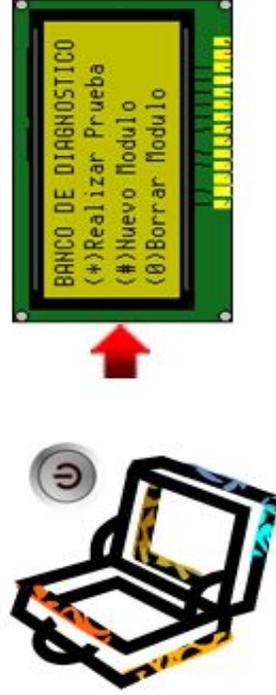
## INICIO RÁPIDO

En esta sección se enseña a utilizar el S.D.P. de forma rápida.

El S.D.P. se gestiona mediante un software instalado previamente en el, para iniciar a usarlo es necesario

oprimir el botón de encendido. Al encender el sistema encontraremos en la pantalla de cristal líquido LCD el menú principal, en el cual podremos elegir uno de las opciones (Figura 2) para empezar a utilizarlo.

Figura 2. Menú principal

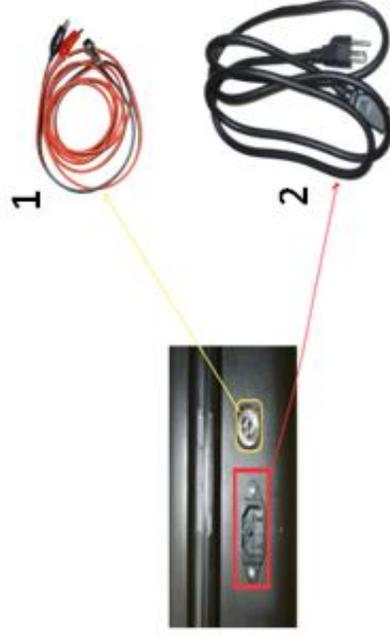


los elementos de un dispositivo, bien sea contactores y contactores indicadores.

Es indispensable cumplir con las siguientes condiciones previo a un diagnóstico.

1. Debe estar conectado el cable de poder (Figura 3), hay dos cables útiles para hacer la conexión: Uno para la conexión a la red alterna (1) y el otro para conectarla a las baterías del helicóptero (2), se debe usar solamente uno de los cables de poder.

Figura 3. Cables de poder



## 1. REQUISITOS PARA UTILIZAR EL SISTEMA DE DIAGNÓSTICO PORTABLE

 **Nota Aclaratoria:** En el desarrollo de este manual se usa el término Switch y su abreviatura sw para referirse a

2. Al iniciar una prueba es importante elegir el cable de conexión necesario para comunicar el dispositivo que se desea diagnosticar con el S.D.P. Se proporcionan dos (2) cables de conexión; cada uno en un extremo cuenta con un conector universal etiquetado como **J1** y es el que debe conectarse al S.D.P, el otro extremo del cable esta etiquetado según el dispositivo a diagnosticar, bien sea **AACP** o **MOUNTGRIP** (Figura 4).

Figura 4. Cables de conexión para cada dispositivo



## 2. COMO USAR EL SISTEMA DE DIAGNÓSTICO PORTABLE (S.D.P)

El menú principal (Figura 5) presenta tres (3) opciones a las cuales podemos acceder mediante la pulsación de las teclas funcionales del teclado del sistema.

Figura 5. Menú principal



En el teclado del sistema:

- Oprimir \* (asterisco), se accede a realizar pruebas.
- Oprimir # (numeral), se accede a Agregar un nuevo dispositivo.
- Oprimir 0 (cero), se accede al menú de borrar las tablas de datos.

### 2.1 REALIZAR PRUEBAS CON EL S.D.P

En pantalla (Figura 6), se presenta un menú en el cual con las teclas numéricas **1, 2, 0** se puede elegir uno de las siguientes opciones:

Figura 6. Modo de prueba



La opción **1, Independiente**, trabaja con el maletín sin hacer uso de un computador portátil; el sistema está diseñado para comodidad del usuario.

La opción **2, Interfaz PC (USB)**, trabaja exactamente igual que el equipo portable, mas interactiva y fácil de usar.

La opción **0, Salir**, envía al menú principal del maletín.

### 2.1.1 REALIZAR PRUEBAS EN MODO INDEPENDIENTE

Al presionar la tecla **1**, se elige el modo independiente y en pantalla (Figura 7) se presenta el menú con los dispositivos disponibles para realizar la prueba.

Figura 7. Menú dispositivos



Por defecto el S.D.P tiene la posibilidad de realizar la prueba tanto con el dispositivo MOUNTGRIP como con el AACP, sin embargo es expansible hasta 30 dispositivos. Al elegir presionar la tecla **1** o **2**, en pantalla (Figura 8) se presentara una precaución.

Figura 8. Advertencia inicial



Desactivar los C. indicadores visuales (estos no deben estar alumbrando) y los contadores (no debe estar ninguno presionado o en posición de ON) presionar la tecla \* (asterisco) para continuar con la prueba, y en pantalla se mostrará la Figura 9.

Figura 9. Dispositivo elegido



9

Según el dispositivo elegido se indica el arnés o cable de conexión que debe estar conectado, en caso de no estarlo este es el momento para asegurarse de la correcta conexión del dispositivo con el sistema de diagnóstico. Seguidamente el Sistema de Diagnóstico indicara según el dispositivo elegido que fuente de voltaje será utilizada (Figura 10) para probar los indicadores visuales.

Figura 10. Fuente de Voltaje Visualizadores



En los dispositivos por defecto, se deben cumplir los siguientes requisitos:

MOUNTGRIP = Fuente conmutada 28v.

AACP= Fuente conmutada 5v.

Estos valores de voltaje son acorde al funcionamiento interno de cada dispositivo.

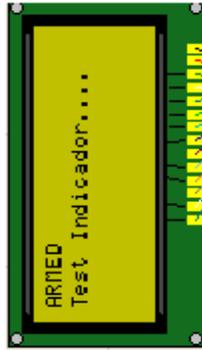
Seguido a esto se da inicio a las pruebas para los switches de un determinado dispositivo.

#### **Probar indicador visual:**

En pantalla (Figura 11) se presenta el nombre del indicador que se está probando y aparecerán al lado de la frase Test Indicador un punto cada segundo

aproximadamente, marcando de esta forma el tiempo de prueba para el indicador visual.

Figura 11. Prueba indicador visual



Luego del último punto en pantalla se presentara la Figura 12.

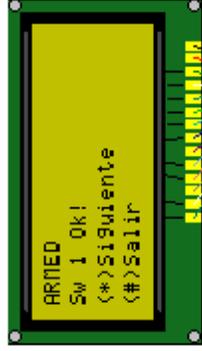
Figura 12. Retorno visual



Si el indicador se encuentra en correcto estado, este debió estar iluminado durante el tiempo en que se realizo el Test indicador. Presionando la tecla '\*' o '#' (Figura 12) se informa al Sistema de Diagnóstico la condición del

indicador. Si el indicador encendió y es presionada '\*', obtendremos la Figura 13.

Figura 13. Resultado OK! para la prueba del indicador visual.



Si por el contrario el indicador no encendió y se presiona '#', tenemos la Figura 14.

Figura 14. Resultado Error! Para la prueba del indicador visual

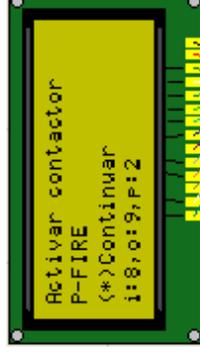


### Probar switch clase 0:

El switch clase 0 es aquel que debe ser activado para probar su funcionamiento (Figura 15).

Se debe activar el contactor y seguidamente con el contactor activo presionar la tecla \* (asterisco).

Figura 15. Activar contactor.



Dependiendo del estado del contactor, se obtendrá una de las siguientes respuestas en pantalla:

- Si el sw se encuentra en correcto funcionamiento se observa la Figura 16.

Figura 16. Resultado "OK!", prueba activa



- Si por el contrario el sw no está apto para ser usado y debe pasar a servicio técnico, en pantalla se observa la Figura 17.

Figura 17. Resultado "ERROR!", prueba activa



### Probar switch clase 2:

El switch clase 2 es aquel que no debe ser activado para probar su funcionamiento, al probar un switch clase 2, en pantalla se presenta la Figura 18.

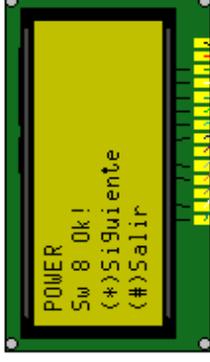
Figura 18. No activar contactor



Si se activa el contactor, seguidamente se presiona la tecla \* (asterisco). Dependiendo del estado del contactor obtendremos una de las siguientes posibles respuestas en pantalla.

- Si el sw se encuentra en correcto funcionamiento, observaremos la Figura 19.

Figura 19. Resultado "OK!", prueba pasiva.



- Si por el contrario el sw no está apto para ser usado y debe pasar a servicio técnico, en pantalla se presenta la Figura 20.

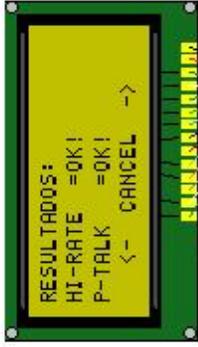
Figura 20. Resultado "Error!", prueba pasiva.



### Resultados de la prueba:

Terminados de probar todos los switches del dispositivo se obtiene la lista de resultados de la Figura 21.

Figura 21. Resultados



Con las teclas de navegación (multifuncionalidad de \*, 0, #), si presionamos la tecla 0 correspondiente a la función CANCEL regresaremos al menú de inicio del S.D.P, con \* y # nos podemos desplazar y ver el reporte de todos los componentes del dispositivo que fueron diagnosticados, si todos los componentes tienen como resultado OK!, el dispositivo está listo para su uso, de lo contrario deberá pasar a revisión por parte de servicio técnico.

### 2.1.2 REALIZAR PRUEBAS EN MODO INTERFAZ PC (USB)

Al presionar la tecla 2 en el menú de realizar la prueba y elegir el modo interfaz PC (USB), se presenta la Figura 22, y entrara en modo de espera de las acciones que se

lleven a cabo en El computador portátil, aplicación Interfaz Grafica de Usuario S.D.P

Figura 22. Conexión USB.



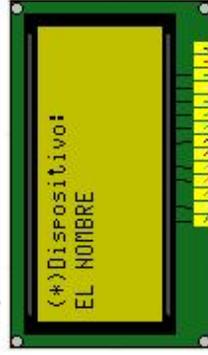
### 2.2 AGREGAR UN DISPOSITIVO NUEVO

Para agregar un nuevo dispositivo se elige presionando la tecla # (numeral) la opción Nuevo en el menú principal que se muestra en la Figura 1.

Al ingresar a esta opción, se debe introducir una clave de 5 números suministrada por el fabricante, se tendrá acceso a modificar la Tabla de Datos del S.D.P por lo tanto se recomienda un alto cuidado y no ingresar a esta a menos que sea estrictamente necesario.

Una vez ingresada y validada la clave se observa en pantalla la Figura 23.

Figura 23. Nombre Dispositivo



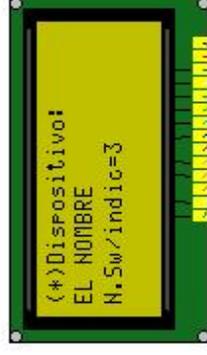
Como primer paso para agregar un nuevo dispositivo se debe ingresar el nombre de este y validarlo con la tecla \*, para este caso de ejemplo el nuevo dispositivo tiene por nombre "EL NOMBRE". Para ingresar la información necesaria del dispositivo se usa el teclado alfanumérico del Sistema de Diagnóstico, su distribución se puede apreciar en la figura 24.

Figura 24. Teclado del Sistema



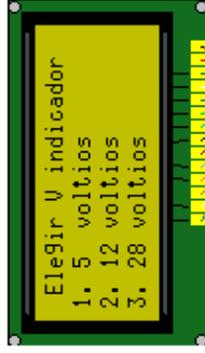
De manera tal que mediante la pulsación repetitiva de una misma tecla obtienen los caracteres numéricos y del alfabeto, si deseamos borrar el carácter escrito en pantalla lo podemos hacer con la tecla multifunción #. Seguidamente debemos especificar el total de switches que tiene el dispositivo, y validar la información con la tecla \* tal como lo indica la Figura 25.

Figura. 25 Información general del dispositivo.



Una vez ingresada y validada la información anterior se presenta la Figura 26, Seguidamente se elige el valor del voltaje necesario para activar los indicadores visuales del dispositivo en cuestión.

Figura. 26 Voltajes de Indicadores.



Al presionando las teclas 1, 2, o 3, se elige el valor necesario, esto es de suma importancia, si no está seguro del valor de voltaje para los indicadores visuales le recomendamos abortar y borrar el dispositivo que se estaba agregando como se explica en la sección "2.3 Borrar Tabla de datos en memoria del S.D.P.". Para este caso de ejemplo se elige la tecla 1 el valor de 5 voltios.

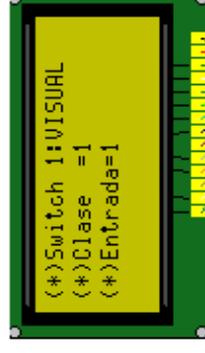
Terminada de agregar la información general del dispositivo, proseguir con ingresar la información específica para cada uno de los switches, para este caso de ejemplo serán tres (3) ya que así lo especificamos previamente (Figura 25).

#### **Agregar un Switch clase 1 (indicador visual):**

De manera similar a como lo hicimos con la información general del dispositivo (Figura 27), se ingresa el nombre

del switch, su clase y el parámetro entrada. Se valida cada campo presionando la tecla \* (asterisco).

Figura 27. Información del Indicador.



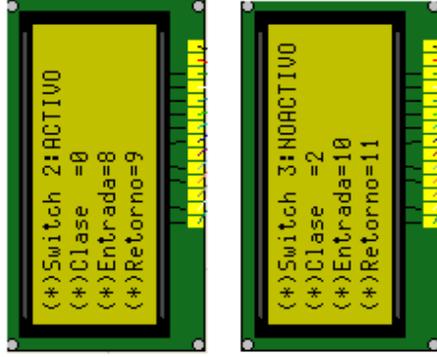
El parámetro clase informa que tipo de prueba se debe realizar, de esta forma si es un Indicador visual debe ponerse siempre clase =1.

#### **Agregar un Switch clase 0 o clase 2:**

De forma similar a como se hizo previamente, se ingresa y valida el nombre del switch, su número de clase, y los parámetros entrada y retorno. En pantalla se verá la figura 28.

Clase =0, si la prueba de diagnóstico que se realice corresponde a un switch que se debe activar.

Figura 28. Información switch clase 0 y 2.



Clase =2, si la prueba de diagnóstico que se realice corresponde a un switch que no se debe activar.

Los parámetros Entrada, Retorno son específicos para cada switch e indican que terminal del conector se utilizara para enviar la señal y que terminal para leer la señal de retorno correspondiente. Para los visualizadores el retorno se lleva a cabo de manera visual por el usuario como se explico en la sección "2.1 REALIZAR PRUEBAS CON EL S.D.P". Es recomendable obtener más información respecto a estos parámetros en el libro DISEÑO E IMPLEMENTACION DE SISTEMA DE

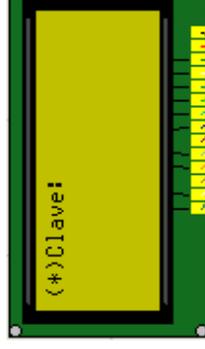
DIAGNÓSTICO PORTABLE PARA AERONAVES, capítulos 2 y 3.

### 2.3 BORRAR DISPOSITIVOS DE LA BASE DE DATOS

Para borrar un dispositivo debe elegir presionar la tecla 0 (cero) la cual es la opción Borrar T.D en el menú principal de la Figura 4.

Al ingresar a esta opción se debe introducir una clave de 5 números suministrada por el fabricante, tendremos acceso a modificar la Tabla de Datos del Sistema de Diagnóstico por lo tanto se recomienda un alto cuidado y no ingresar a esta a menos que sea estrictamente necesario.

Figura 30. Ingresar Clave.



Se valida la clave presionando la tecla \* (asterisco). Una vez ingresada la clave correcta, en pantalla se verá el menú de la Figura 31.

Figura 31. Menú para Borrar Dispositivos



- Oprimir \* (asterisco), se borra **TODA** la información de la Tabla de datos almacenada en memoria, es decir **no quedara registro de ningún dispositivo para realizar pruebas de diagnóstico**.
- Oprimir # (numeral), se da la opción de elegir un dispositivo específico que se desee eliminar entre los dispositivos previamente agregados, al eliminar un dispositivo **no se podrá volver a realizar la prueba de diagnóstico para el dispositivo eliminado**.
- Oprimir 0 (cero), salir de este menú y regresar al menú principal de inicio del Sistema de Diagnóstico Portable.

### 3. CONSEJOS Y SOLUCIONES

- Antes de encender el Sistema de Diagnóstico Portable, verificar que el cable de poder se encuentre en perfectas condiciones y la alimentación de la señal AC sea 115 - 120 V.
- Verificar periódicamente que el Cable de conexión para cada dispositivo se encuentre en perfectas condiciones.
- Si desea agregar un nuevo dispositivo para realizar su diagnóstico, necesitara el cable de conexión específico para el dispositivo, lo invitamos a que se comunique con el fabricante para el desarrollo de el nuevo cable de conexión.
- Para mayor información sobre el funcionamiento interno del dispositivo lo invitamos a consultar el libro **DISEÑO E IMPLEMENTACION DE SISTEMA DE DIAGNÓSTICO PORTABLE PARA MODULOS AACP Y MOUNTGRIP**

#### 4. SOPORTE TECNICO:

➤ Al realizar la prueba para el dispositivo MOUNTGRIP tenga en cuenta lo siguiente:

**POWER:** Se dejó de última en la tabla para evitar corto con las señales análogas, por esto se pide que todos los sw estén en estado OFF antes de empezar la prueba.

**ARMED:** El sw POWER debe mantenerse en off, el ojo de gato debe ser presionado para ver que esta funcionando.

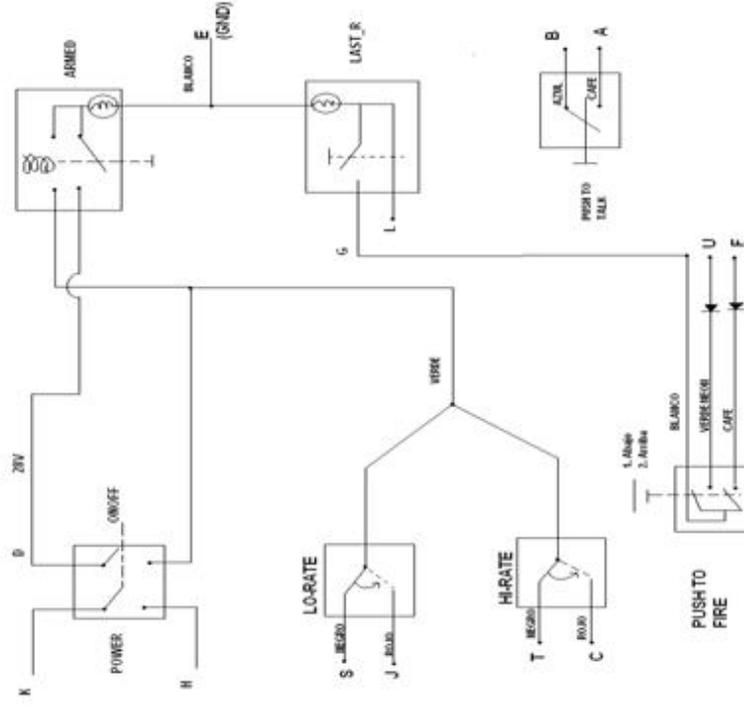
**LAST\_ROUND:** No se debe presionar el ojo de gato ya que se encuentra en contacto directo.

**Lo-RATE:** El sw POWER debe estar desactivado, este sw gatillo se puede probar tanto en dos estados ya que sin presionar esta en circuito cerrado con S y T (Hi-rate y Lo-rate conectados entre sí ver diagrama de conexión, figura 32). Como se dejó trabajando combinaciones J y T funciona presionando el gatillo.

**HI-RATE:** Power off, no presionar ARMED y presionar gatillo. Funciona igual q Lo-rate.

Diagrama de conexiones Empuñadura (MOUNTGRIP)

Figura 32. Diagrama de conexión MOUNTGRIP.



- Al realizar la prueba para el dispositivo AACP, tenga en cuenta que es indiferente para el sistema desactivar los switches después de su respectiva prueba.