

xx(86645.1)



**CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL IPN
UNIDAD GUADALAJARA**

**Herramienta para el desarrollo y pruebas de
algoritmos computacionales, con aplicación a
manejadores externos de papel Hewlett Packard**

Tesis que presenta
Nahúm Vladimir Castillo Félix

Para obtener el grado de
Maestro en Ciencias

En la especialidad de
Ingeniería Eléctrica



Guadalajara, Jal. Octubre de 2000

CLASIF.	
ADQUIS.	TESIS-2001
FEC	29-III-01
PRO	Depo. Serid. 101

Bibliograficos

Herramienta para el desarrollo y pruebas de algoritmos computacionales, con aplicación a manejadores externos de papel Hewlett Packard

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Nahúm Vladimir Castillo Félix

Ingeniero Electrónico
Instituto Tecnológico de Sonora 1994-1998

Becario de CONACYT, expediente no. **129176**

Director de Tesis:
Dr. Deni Librado Torres Román

CINVESTAV del IPN Unidad Guadalajara, Octubre de 2000

Agradecimientos

A mis padres, Flora y Mauro, quiénes me han apoyado siempre en cada decisión de mi vida y a quiénes, con su ejemplo y dedicación, debo cada parte de lo que soy. Espero nunca defraudarlos y estar junto a ustedes siempre. Los amo.

A mi familia, por su apoyo y comprensión en cada momento de mi vida; a mis abuelos, Ramona, Cecilio, Aurora y Francisco; a todos mis tíos y tías; a mis suegros, Rosa y Jesús; a mis cuñadas y sobrinas, Guadalupe, Adriana, Karla, Ana Fernanda y Jimena; a mis primos, en especial a Lucero, Víctor, Orlando, René, Ari, Paty, Karla, Alma Cecilia, Ricardo, Juan y Eduardo.

A CINVESTAV, por darme la oportunidad de cumplir con mi objetivo profesional, y más importante aún, por haberme permitido conocer a excepcionales profesores y excelentes compañeros.

A México y su gente, por que gracias a su apoyo a través del instituto CONACyT, ha hecho posible que yo y mis compañeros sigamos preparándonos para luchar por un mejor país.

A mis profesores, quienes han sabido inculcar en mí el gusto por la investigación y la excelencia profesional; en especial, al Dr. Deni Torres Román por su paciencia y perseverancia; al Dr. Arturo Veloz, por su inigualable calidad y dedicación; al Dr. Manuel Guzmán, por sus consejos y oportunidad de formar parte de su proyecto; a Ramón Rodríguez, Jorge Hermsillo y Ramón Parra, por ser parte importante de mi formación.

Al departamento de investigación y desarrollo de Hewlett Packard de México, por conducto de Héctor Rodríguez, por haber gestado la oportunidad y creer en la posibilidad de involucrar a un estudiante en un proyecto de investigación y diseño.

A cada una de las personas que hicieron posible el desarrollo de mi tesis y que además de compartir conmigo toda su experiencia y calidad profesional, me han regalado su amistad, en especial a Jorge de la Torre, Jaime de la Torre, Raúl Ocampo y Héctor Rodríguez.

A Hernán Gutiérrez, ya que gracias a su apoyo, paciencia y constante ejemplo durante el desarrollo de las etapas de mi trabajo, fue posible alcanzar cada uno de los objetivos del proyecto.

A mis compañeros de generación, porque en ellos encontré una amistad y fraternidad muy especial. Gracias Liz, Héctor, Eduardo, Pablo, Benítez, Jacobo, Enrique, Raúl, Temoc y Jorge Fausto.

A la vida, por darme la dicha de seguir contando con mis hermanos Anhué, Anhuar, Tadeo, Fabián, Ricardo, Mario y por haber puesto los medios para que en mi camino aparecieran Alfredo, Sergio, Alberto, Vicente y Joaquín.

A Dios, por permitirme despertar y convivir cada día de mi vida con la mujer que amo. Gracias Irene por tu dedicación, valor y motivación para emprender juntos nuestra aventura. Te amo.

Para las personas más importantes en mi vida:

Flora, Mauro, Anhué, Anhuar e Irene

Resumen

La elaboración del esquema de control para cada nuevo dispositivo en el área de la electrónica (hardware y software), generalmente se lleva a cabo mediante un proceso iterativo que consta de las siguientes etapas: concepción, análisis y diseño, implementación y pruebas.

Una familia de productos o dispositivos está constituida por elementos que dadas sus similitudes en estructura, operación y funciones, pueden ser diseñados bajo una misma arquitectura. De acuerdo a este concepto surge lo que se conoce como herramienta de desarrollo, que representa una base para el diseño de cada nuevo producto de la familia.

Hoy en día, existen impresoras diseñadas para satisfacer altas necesidades de impresión, y que mediante el uso de manejadores de papel, proporcionan además la capacidad para acomodar, separar, distribuir y engrapar documentos. Un dispositivo manejador de papel tiene como función principal mover, con un alto grado de precisión y, en coordinación con la impresora, hojas de papel. Dado que estos dispositivos están basados en la operación y control de un sistema electromecánico (motores, sensores), con características similares, pueden ser construidos sobre una misma arquitectura.

Con el presente trabajo se implementó una herramienta que simplifica de manera eficiente, el diseño del esquema de control de los dispositivos manejadores de papel que Hewlett Packard desarrolla. Permite que ingenieros con diferentes grados de expertez ejerciten a los motores con diversos modos de operación.

El sistema está conformado por tres bloques funcionales básicos: hardware, firmware y software de aplicación.

Se diseñaron y construyeron dos tarjetas de circuitos impresos que constituyen la arquitectura de hardware del sistema. Se componen de varios elementos básicos: microprocesador, circuitos periféricos al procesador (memorias, interfaces de comunicación, lógica digital), manejadores de motores, electrónica para lectura de sensores.

Se desarrolló el firmware de control en lenguaje C, formado por dos componentes básicos: el primero proporciona un conjunto $A = \{A_0, A_1, \dots, A_n\}$ de algoritmos para ejercitar los motores; el segundo componente consiste en un código que accede al conjunto A para mover motores y, a su vez, se comunica al exterior a través de un puerto serial estándar a una velocidad de 38400 bps. La información que este módulo maneja es capturada e interpretada por el tercer bloque funcional, el software de aplicación de usuario.

El software de aplicación de usuario implementado reside en una PC y, además de comunicarse por el puerto serial con el firmware, provee una interfaz gráfica, desarrollado en objetos, empleando como herramienta C++ Builder 3.0, para que un usuario interactúe con el sistema de acuerdo a sus propias necesidades.

El uso de esta herramienta representa una plataforma para el desarrollo del esquema de control de los manejadores de papel Hewlett Packard, debido a que cada nuevo producto se origina a partir de una arquitectura ya probada, que sólo debe ser ajustada a las necesidades de cada dispositivo en particular, lo que se traduce en una considerable reducción del tiempo requerido para el diseño y prueba de los prototipos, que conducen a un producto final. Además, representa una base para la investigación de los perfiles de movimiento de las trayectorias del papel a través de los manejadores, utilizados en los diversos productos de calidad que esta empresa desarrolla.

Tabla de Contenido

RESUMEN	1
1 INTRODUCCIÓN	4
2 MANEJADORES DE PAPEL PARA IMPRESORAS HEWLETT PACKARD	5
2.1 Manejo de papel en una impresora	6
2.2 Manejadores externos de papel	8
2.3 Manejadores de papel Hewlett Packard	9
2.3.1 Tipo de papel	11
2.3.2 Sistema electromecánico	11
2.3.3 Movimiento de papel en un manejador	13
2.3.4 Comunicación impresora-manejador de papel	16
3 ARQUITECTURA	18
3.1 Especificación	18
3.1.1 Perspectiva del producto	18
3.1.2 Características del usuario	19
3.1.3 Estudio de viabilidad del sistema	19
3.1.4 Requerimientos específicos	19
3.1.5 Restricciones generales	21
3.2 Análisis y diseño	21
3.2.1 Hardware	22
3.2.2 Firmware de control	23
3.2.3 Software de aplicación	24
4 HARDWARE	26
4.1 Hardware del sistema	26
4.1.1 Microprocesador y circuitos externos	27
4.1.2 Protocolo de comunicación CAN 2.0B	32
4.1.3 Comunicación serial	40
4.1.4 Conexión con el hardware de aplicación	41
4.1.5 Fuente de alimentación	41
4.2 Hardware de aplicación	41
4.2.1 Mapeo de memoria externa	43
4.2.2 Electrónica para lectura de sensores	44
4.2.3 Lógica de interrupción de sensores	44
4.2.4 Manejadores de motor	46
4.3 Diseño de PCB's	50
Herramienta para el desarrollo y pruebas de algoritmos computacionales, con aplicación a manejadores externos de papel Hewlett Packard	2

4.4	Pruebas de hardware	51
4.4.1	Pruebas eléctricas	51
4.4.2	Prueba funcional del hardware del sistema	52
4.4.3	Prueba funcional del hardware de aplicación	52
5	SOFTWARE Y PROTOCOLO DE COMUNICACIÓN	54
5.1	Software de aplicación de usuario	54
5.1.1	Arquitectura del software	55
5.1.2	Interfaz de usuario	58
5.1.3	Comunicación con el firmware de control	63
5.1.4	Pruebas del software	66
5.2	Protocolo de comunicación RS232C	67
5.2.1	Principio de operación	67
5.2.2	Implementación	69
6	CONCLUSIONES Y TRABAJO FUTURO	70
7	REFERENCIAS BIBLIOGRÁFICAS	72

1 Introducción

La concepción e implementación de un esquema de control, para cada nuevo dispositivo en el área de la electrónica, se lleva a cabo mediante un procedimiento que consta de varias etapas: concepción, análisis y diseño, implementación y pruebas. Este proceso se repite constantemente en el departamento de desarrollo de una empresa, aún cuando cada nuevo producto lanzado al mercado presente características similares, en estructura y operación, a su predecesor.

Una línea o familia de dispositivos la constituyen todos los elementos que cumplen con esta condición de generalidad, y que además pueden ser controlados a partir de una misma arquitectura. Aprovechando este concepto, y a fin de reducir los tiempos de desarrollo de cada producto, es posible crear una herramienta que sirva como base o plataforma para el diseño e implementación de los nuevos dispositivos de la familia.

Hoy en día, existen impresoras diseñadas para satisfacer altas necesidades de impresión, y que mediante el uso de manejadores de papel, proporcionan además la capacidad para acomodar, separar, distribuir y engrapar documentos. Un dispositivo manejador de papel tiene como función principal mover, con un alto grado de precisión y en coordinación con la impresora, hojas de papel.

Dado que estos dispositivos están basados en la operación y control de un sistema electromecánico (motores, sensores), con características similares, pueden ser controlados de acuerdo a un mismo esquema.

A fin de proporcionar una plataforma que simplifique el desarrollo de los sistemas de control de la nueva familia de productos manejadores de papel Hewlett Packard, y dado que su función principal es mover con alto grado de precisión, en coordinación con una impresora, y a través de trayectorias definidas, las hojas de papel, el sistema debe contar con los elementos para:

1. Proveer las señales requeridas para generar los movimientos de los motores de corriente directa y de pasos que hacen posible el transporte de papel.
2. Detectar por medio de sensores el estado de una hoja de papel durante su transporte.
3. Almacenar y ejecutar códigos programables por medio de un sistema de control basado en un microprocesador.
4. Permitir la comunicación del manejador con otros dispositivos físicos, empleando un protocolo estándar.
5. Comunicarse con una PC.
6. Interactuar con el usuario a través de un programa que se ejecute en una PC.

El objetivo es diseñar e implementar una herramienta que proporcione a ingenieros los elementos de configuración y control necesarios para implementar en períodos cortos de tiempo, la funcionalidad de los prototipos que conducen a un producto final de alta calidad.

2 Manejadores de papel para impresoras Hewlett Packard

Una impresora es un dispositivo que acepta texto y gráficos de una computadora y transfiere la información, usualmente, sobre hojas de papel de tamaño estándar [1].

Actualmente existen diferentes modelos de impresoras que varían en tamaño, velocidad, precio, capacidad; sin embargo, se pueden clasificar como dispositivos de impacto o no impacto. Las primeras impresoras de impacto trabajan como una máquina de escribir automática, con teclas que imprimen uno o varios caracteres al mismo tiempo; sin embargo, la impresora de matriz de puntos, que imprime una línea completa a la vez, ha sido una de las más populares debido a su bajo costo.

La impresora de no impacto más conocida es la de inyección de tinta, que permite el uso, a bajo costo, del color. Su trabajo consiste en esparcir tinta, que toma de un cartucho, sobre un área pequeña de papel. La impresora láser utiliza un rayo de luz reflejado en un espejo, para inyectar tinta (tóner) sobre el papel, de manera muy precisa.

La interfaz más común para la impresora y la computadora ha sido el puerto paralelo con su conector de 36 pines. En un futuro, sin embargo, se prevé el uso de una interfaz serial, principalmente a través del puerto USB (Universal Serial Bus), que no requiere de un adaptador adicional y que proporciona un conector mucho más pequeño y eficiente. Para ambientes de oficina, donde las impresoras representan un recurso compartido, el medio de comunicación es diferente, siendo implementada por medio de una tarjeta que interconecta las computadoras y la impresora a una red local.

El lenguaje de una impresora está constituido por comandos provenientes de una computadora hacia el dispositivo, que indican el formato del documento que será impreso. Estos comandos describen el tamaño de letra, las gráficas, la compresión de los datos, el color, etc. Los lenguajes de impresión más populares son Postscript y PCL (Printer Command Language).

Postscript es un lenguaje que utiliza frases del idioma inglés para describir la apariencia de una página impresa. Es desarrollado por Adobe en 1985, e introduce nuevas opciones de impresión, como letras fuera de línea y gráficas vectoriales.

PCL es un código utilizado para enviar mensajes a la impresora que coordinan la impresión de documentos. Se conoce también como "código de escape" debido a que la tecla 'Escape' determina el inicio de un comando. Hewlett Packard desarrolló originalmente PCL para sus impresoras de matriz de puntos y de inyección de tinta.

Para la mayor parte de los usuarios la calidad de una impresora está determinada por cinco parámetros: *resolución γ* , *velocidad v* , *color κ* , *memoria M* y *manejo de papel ρ* [1].

Resolución γ

La resolución (calidad de texto e imágenes en el papel) se mide en términos de puntos por pulgadas (dpi). Una $\gamma > 600$ dpi, provee al usuario de una calidad aceptable para sus trabajos de impresión.

Velocidad v

Para altos volúmenes de impresión, la velocidad es un factor importante. Se mide en páginas/unidad de tiempo. En la actualidad esta unidad es el minuto. La velocidad de las impresoras varía desde 3 a 6 p/min. a 32 p/min.

Color κ

Es importante para la impresión de presentaciones, mapas y todo tipo de trabajo donde forma parte esencial de la información; sin embargo, la mayor parte de los usuarios encuentran en una impresora a blanco y negro un modo más económico de trabajar. Se mide de acuerdo al número de colores y tonalidades que maneja.

Memoria M

La mayor parte de las impresoras vienen con una cierta cantidad de memoria que puede ser expandida por el usuario. La velocidad del dispositivo se incrementa proporcionalmente a M ya que el dispositivo es capaz de contener más información en un mismo proceso de comunicación con la PC.

Manejo de papel ρ

Para oficinas o departamentos de empresas con altas necesidades de impresión, el manejo de papel de una impresora representa un parámetro muy importante en su selección. Una impresora, de acuerdo a su diseño, puede soportar el uso adicional de algunos elementos conocidos como “manejadores de papel”, colocados a la entrada y salida del dispositivo de impresión principal, que permiten al sistema trabajar durante ciertos períodos de tiempo sin supervisión alguna, dado que son capaces de almacenar más de 500 hojas y alimentar constantemente a la impresora con el papel que necesita para trabajar. De igual forma, estos dispositivos son capaces de recibir los documentos impresos y distribuirlos sobre diferentes contenedores de salida, de acuerdo a una previa configuración, lo que se traduce en una forma más eficiente de atender las necesidades de impresión de varios usuarios.

2.1 Manejo de papel en una impresora

El control del proceso de impresión es implementado con un procesador dedicado, que a partir de los comandos que recibe de una computadora, crea una imagen en papel del documento transmitido. Para cumplir con su objetivo, el controlador requiere llevar a cabo varias tareas:

1. Establecer su comunicación con la PC.
2. Interpretar los comandos que arriban y que generalmente se encuentran inmersos junto a la información que será impresa.
3. Crear la imagen de acuerdo a ciertos parámetros ya definidos: tamaño de hoja de papel, márgenes, tipos de letras, etc.
4. Plasmar la imagen sobre el papel.

Este proceso es posible, gracias al mecanismo físico que implementa el movimiento de papel en la impresora; alimenta de papel al dispositivo, lo transporta a través del sistema de impresión y lo entrega al usuario por medio de un contenedor de salida.

El movimiento de papel en una impresora se describe por medio de ciertas trayectorias que determinan la posición exacta de una hoja durante su paso por el dispositivo. La Figura 2-1 ilustra este proceso, donde por medio de rodillos motorizados el papel es desplazado desde los contenedores de entrada o el alimentador manual hacia el sistema de impresión para finalmente ser depositado en un contenedor de salida.

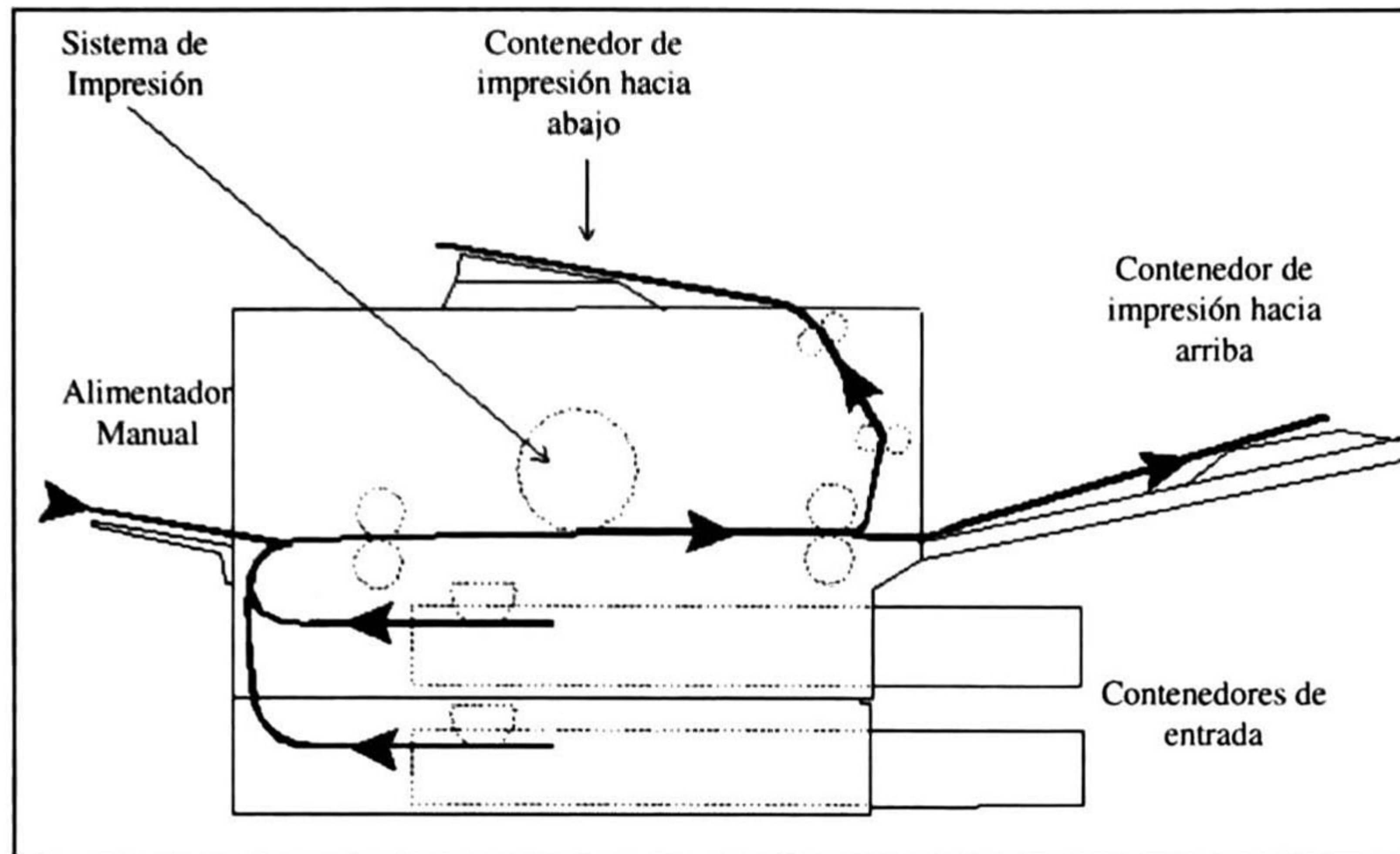


Figura 2-1 Trayectoria de papel en una impresora

A continuación se describe más a detalle la función de cada una de los elementos que toman parte del movimiento de papel en una impresora.

Contenedor de papel de entrada

Impresoras de gran capacidad utilizan contenedores de entrada para almacenar más de 200 hojas y asegurar que siempre exista una hoja de papel lista para ser tomada por el transportador de papel. Este mecanismo se compone de un elevador motorizado que mueve el papel del contenedor hacia la posición requerida por el sistema de impresión. Generalmente, una impresora tiene desde uno hasta cinco contenedores que pueden ser llenados con diferentes tipos de papel; así, el controlador de impresión obedece los comandos de la computadora que le especifican el contenedor o el tipo de papel a utilizar.

Alimentador manual de papel

El alimentador manual le permite al usuario utilizar un tipo de papel especial para su proceso de impresión como son tarjetas, etiquetas o transparencias. Este dispositivo tiene el objetivo de evitar su alimentación por medio de un contenedor, ya que dadas las características del papel, puede llegar a atascarse con mayor facilidad. Su uso, elimina la trayectoria de papel descrita por la relación entre el contenedor de entrada y el sistema de impresión. Además de soportar el uso de papel especial, puede ser alimentado con hojas de tamaño no estándar, por lo que no requiere de un contenedor con una dimensión específica. Su única desventaja es que dado que el dispositivo tiene que detectar las dimensiones del papel, el proceso de impresión se vuelve mas lento.

Contenedor de papel de salida

El contenedor de salida recibe el papel impreso. Algunas impresoras contienen varios contenedores, que pueden ser seleccionados a través de un comando, que permiten la separación de diferentes trabajos, sin embargo, la mayoría cuenta con dos, el contenedor de impresión hacia arriba y el contenedor de impresión hacia abajo.

Contenedor de impresión hacia arriba

Recibe el papel con la impresión hacia arriba, ocasionando que un documento de varias páginas sea ordenado de manera inversa, ya que la página dos cae sobre la página uno.

Debido a esto, la mayoría de las impresoras son diseñadas de manera tal que este accesorio sólo sea utilizado como un contenedor auxiliar para tarjetas, transparencias y otros tipos de documentos alimentados manualmente. Como el elemento auxiliar que es, la mayoría contienen sólo una pequeña parte de los trabajos impresos. Sin embargo, en impresoras de oficinas de gran capacidad, se pueden agregar mecanismos externos manejadores de papel que permiten un uso más eficiente para la salida.

Contenedor de impresión hacia abajo

Recibe el papel con la impresión hacia abajo, por lo que los documentos son entregados en el orden correcto, con la página dos detrás de la página uno. Como esta es la forma en la que los documentos impresos son requeridos, es utilizado como el principal dispositivo de salida para una impresión normal. La mayoría de los productores de impresoras diseñan sus contenedores de impresión hacia abajo, con la capacidad de manejar al menos la misma cantidad de papel que sus contenedores de entrada, aumentando así la cantidad de tiempo que la impresora puede trabajar sin supervisión.

2.2 Manejadores externos de papel

El número y capacidad de los contenedores de papel de una impresora, varían de manera considerable, dependiendo del tamaño y el tipo de la impresora. La mayor parte de las impresoras personales tienen un contenedor de hojas con capacidad para unas 100 hojas de papel estándar y la opción para la alimentación manual. En las impresoras estacionarias de gran tamaño, los contenedores tienden a ser mayores. Las impresoras de oficina regularmente soportan un par de alrededor de 250 hojas cada uno, pero pueden llegar a contar con contenedores con capacidad para 2500 hojas.

En algunas impresoras, generalmente estacionarias de gran capacidad y velocidad, el tamaño de página utilizado está determinado por la dimensión del contenedor del papel de entrada (dispositivo que inserta papel en el mecanismo de alimentación de la impresora). La mayor parte de estas impresoras, vienen con contenedores de dimensión única estándar, como el tamaño carta (8.5" x 11") en Norte América, o el A4 (210mm x 297mm) en el resto del mundo. Otros tamaños de papel pueden ser manejados utilizando diferentes contenedores, generalmente opcionales, o por medio de la alimentación manual, que permite la utilización de hojas de tamaño variable, con la desventaja de tener que alimentar a mano, cada una de las hojas a imprimir. Pocos modelos de impresoras de este tipo proveen contenedores de papel ajustables capaces de trabajar sobre un cierto rango de tamaño del papel [2].

Estos dispositivos para oficinas y trabajo de grupos, ofrecen la posibilidad de conectarles a su salida de impresión hacia arriba, un apilador de papel o un mecanismo de distribución final.

Un apilador de salida es de mucha utilidad para procesos de impresión extensos, ya que unido a un contenedor de alta capacidad, reducen la frecuencia de intervención del operador, permitiendo el trabajo continuo de la impresora por largos períodos tiempo y con un mínimo de supervisión. Ofrece además la opción de insertar pausas entre trabajos, lo que extiende su utilidad hacia ambientes de recursos compartidos.

El mecanismo de distribución final, es un sofisticado separador de trabajos, que permite la salida del papel por medio de diferentes contenedores. Un trabajo impreso puede ser directamente dirigido hacia uno de estos separadores, evitando el desorden de las impresiones (cada uno puede ser utilizado por un usuario en particular), o alternativamente, cada una de las partes de un trabajo puede ser colocada en un diferente contenedor.

Cuando la generación de copias de un mismo documento es requerida, el uso del mecanismo de distribución permite que cada una sea entregada en su respectivo separador. De esta manera la impresión directa y eficiente de muchas copias es posible, ya que se pueden imprimir el número que se requieran de la página 1, cada una hacia un separador diferente, seguido de la necesidad de copias de la página 2, y así sucesivamente.

El mecanismo de distribución final es complejo, por lo que es aceptable un mínimo de fallas durante su operación. Son necesariamente grandes físicamente, y su uso está enfocado para impresoras de oficina, convirtiendo una impresora de escritorio en una gran impresora estacionaria de gran tamaño y capacidad.

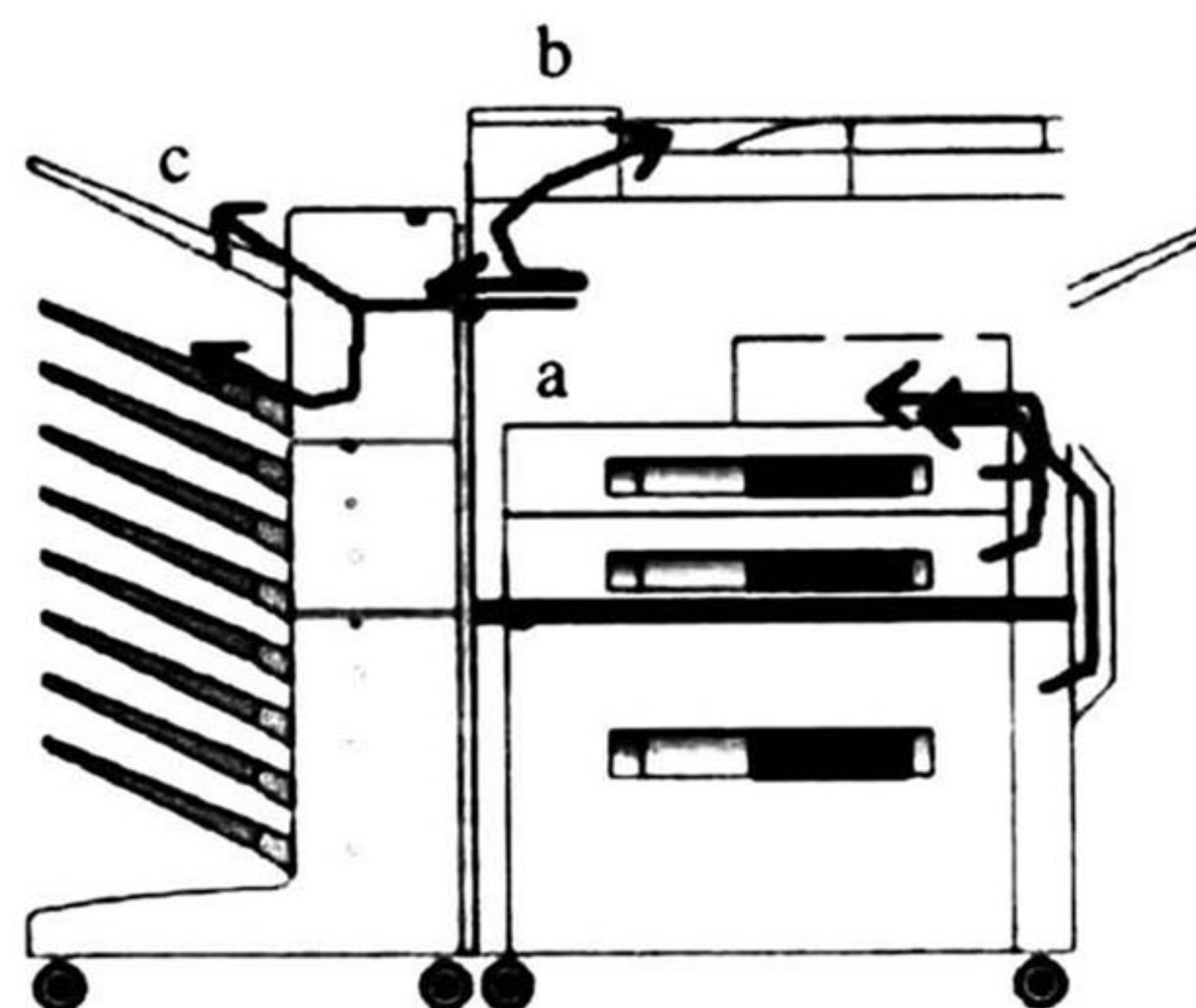


Figura 2-2 Manejadores externos de papel en una impresora de gran capacidad.

2.3 Manejadores de papel Hewlett Packard

El contenedor de salida de papel en una impresora determina el formato de entrega de los documentos que el dispositivo ofrece al usuario. La Figura 2-3 muestra las posibles rutas que ofrece una impresora Hewlett Packard como destino final de las hojas de papel impresas.

Los contenedores de papel con impresión hacia arriba e impresión hacia abajo, constituyen el formato de salida normal del dispositivo. Sin embargo, existe la posibilidad de agregar un elemento "manejador de papel" en lugar del contenedor de impresión hacia arriba, para obtener otro resultado.



- a. Trayectoria del contenedor de impresión hacia arriba
- b. Trayectoria del contenedor de impresión hacia abajo
- c. Trayectoria del manejador de papel

Figura 2-3 Salida hacia los contenedores de papel, en una impresora Hewlett Packard.

Un manejador de papel Hewlett Packard ofrece la oportunidad de apilar, separar, distribuir e incluso engrapar trabajos con el propósito de hacer más eficiente el uso de impresoras de gran capacidad [2].

El objetivo principal del manejador es mover, con alto grado de control y precisión, hojas de papel desde una impresora, hacia un contenedor de salida propio, con la idea de proporcionar un acabado y distribución final diferente al ofrecido por el dispositivo de impresión.

Los manejadores que HP desarrolla para sus impresoras son el *apilador de papel* y el *buzón de 8 separadores*.

Apilador de papel

El apilador de papel (Figura 2-4) provee la funcionalidad requerida para satisfacer las necesidades de manejo de documentos impresos de oficinas o departamentos, ya que además de soportar grandes volúmenes de impresión, ofrecen la posibilidad de separar, insertar retardos y engrapar trabajos terminados, todo en un solo paso [3].



Figura 2-4 Apilador de papel Hewlett Packard

Entre sus principales características se encuentran:

- Capacidad para contener hojas tamaño estándar, separar y engrapar documentos en un solo paso.
- Capacidad para engrapar documentos, con multiposicionamiento de grapas (capacidad para configurar la posición de una, dos, tres o seis grapas).
- Capacidad para insertar retardos de tiempo para un mejor manejo de los trabajos terminados.
- Facilidad para seleccionar desde el manejador de la impresora, el contenedor de salida deseado entre el apilador y el contenedor de impresión hacia arriba.
- Facilidad para configurar y controlar su operación desde el panel de control de la impresora.

Buzón de 8 separadores

El manejador de papel conocido como buzón de 8 separadores (Figura 2-5) enfoca su uso a grupos de trabajo y departamentos donde 8 o más usuarios comparten una misma impresora, o donde ésta se encuentra a una considerable distancia del lugar de trabajo. Su función principal es separar trabajos impresos con el fin de evitar su desorden [3].

Las principales características de este dispositivo son:

- Capacidad para contener hojas de papel tamaño estándar.
- Facilidad para seleccionar un separador a través del manejador de la impresora.

- Facilidad para configurar y controlar su operación desde el panel de control de la impresora.
- Capacidad para trabajar de acuerdo a cuatro modos de operación diferentes:
 - Buzón* – 8 separadores, que pueden ser asignados a un usuario, grupo o departamento, con un nombre identificador configurable en el manejador de la impresora.
 - Apilador* – acomoda las hojas empezando con el separador de abajo y hacia arriba.
 - Separador de trabajos* – Cada trabajo es enviado al siguiente separador de arriba hacia abajo.
 - Separador de copias* – Cada copia de la página de un trabajo es enviada a un separador diferente.



Figura 2-5 Buzón de 8 separadores Hewlett Packard

2.3.1 Tipo de papel

El tipo de papel utilizado por los manejadores Hewlett Packard, y en general por las impresoras, debe ser de buena calidad y especial para trabajos de fotocopiado, a fin de que su tamaño (variación +/- 2 mm) y estado (libre de polvo e imperfecciones) sea aceptable.

De igual forma, el papel demasiado delgado no es recomendable, ya que su uso ocasiona frecuentes problemas como hojas atoradas o atascadas, por lo que la mayoría de estos dispositivos son diseñados para trabajar de manera óptima con papel de 80 a 90 gmc (gramos por metro cuadrado).

Algunas veces es deseable la impresión de documentos sobre diferentes tipos de papel. El uso de hojas plastificadas, membretadas o recicladas, es permitido por la mayor parte de impresoras y manejadores, con la aclaración de que los errores de operación pueden incrementarse debido a la dificultad de su manejo. Otros tipos de materiales que un manejador soporta son tarjetas, sobres o etiquetas y en menor escala transparencias, cuyo uso se restringe a dispositivos especialmente diseñados para ello.

2.3.2 Sistema electromecánico

Un dispositivo manejador de papel está constituido por uno o más motores que dadas sus características de operación y la forma en que se encuentran organizados, permiten llevar a cabo el movimiento de papel deseado.

Si bien los motores constituyen la parte central de estos dispositivos, existen también otros elementos indispensables que regulan su funcionamiento debido a que generalmente se encuentran condicionados por una serie de variables como el tiempo, la

posición de una hoja, su misma posición o el cumplimiento de cierta condición en el sistema. Estas variables son descritas por medio de los componentes electrónicos conocidos como sensores.

Existe una gran variedad de procesos que requieren de movimientos para cumplir con su objetivo, sin embargo, el diseño e implementación de su sistema electromecánico depende, en gran medida, de la precisión, la carga o la velocidad requerida del trabajo a realizar. No es lo mismo concebir un sistema capaz de mover una carga considerable de peso, a mover solo una hoja de papel, o desplazar a gran velocidad una banda de transporte, que controlar con alto grado de precisión la punta de ensamble de una tarjeta electrónica. Un sistema por lo tanto, requiere de uno o más tipos de movimiento identificados para cumplir con su objetivo, lo que al final determina su implementación y los tipos de motor utilizados para ello.

El sistema electromecánico de un manejador de papel Hewlett Packard está diseñado en base a tres elementos fundamentales: motores de pasos, motores de corriente directa y sensores (ópticos y mecánicos) detectores de posición. A continuación se describen sus principales características y aplicación en los manejadores de papel.

2.3.2.1 Motor de corriente directa

El motor de corriente directa es la primera máquina inventada por el hombre capaz de convertir energía eléctrica en mecánica. Las características de estos dispositivos, que permiten todavía su uso en muchas aplicaciones son:

1. Capacidad de carga (torque) inicial.
2. Capacidad para operar con baterías y fuentes de corriente directa simples.
3. Costo.

Debido a que la electrónica que se requiere para implementar el control de posición de este tipo de motor es más costosa, su actuación se reduce al movimiento de partes mecánicas como la base de un apilador y la operación de la engrapadora, procesos que requieren cierta fuerza y no mucha precisión.

2.3.2.2 Motor de pasos

Uno de los elementos fundamentales del sistema electromecánico de un manejador de papel Hewlett Packard es el motor de pasos. El uso de este tipo de dispositivo se debe principalmente a las características de operación que presenta:

1. Es una máquina que convierte pulsos de energía eléctrica en movimientos mecánicos discretos.
2. Su control de posición es muy preciso.
3. Cuando es energizado y los pulsos aplicados no cambian, mantiene firmemente su posición.
4. No requieren de retroalimentación para el control de su velocidad o posición.

Dado que el manejo de papel implementado por HP está basado en un sistema de rodillos en serie que guían la hoja de papel desde la impresora hasta su destino final, el control de la velocidad y posición del mecanismo, se implementa por medio de este tipo de motores, lo que lo convierten en el principal elemento actuador.

2.3.2.3 *Sensor detector de posición*

Actualmente es difícil encontrar un dispositivo electromecánico que trabaje sin la ayuda de un sensor. La automatización de los procesos es posible gracias a estos elementos que representan la manera en que un sistema percibe su entorno y se vuelve capaz de tomar decisiones; a pesar de esto, los sensores por su propia naturaleza, no son elementos complejos. El sensor electrónico detector de posición on/off es probablemente el más utilizado y uno de los más simples, su función es enviar señales eléctricas de encendido/apagado al controlador del sistema, para que interprete la información y actúe de acuerdo a su programación. Estos circuitos indican la posición de partes en movimiento, como su entrada y salida al sistema, o puntos intermedios.

Los sensores electrónicos detectores de posición forman parte importante de los manejadores de papel, ya que regulan el funcionamiento de sus motores, estos elementos determinan la posición actual de la hoja de papel que es transportada a través del dispositivo. Un sensor, por ejemplo, detecta la salida de papel de la impresora, que acciona el primer rodillo motorizado encargado de tomar la hoja. Enseguida, sensores colocados estratégicamente sobre la trayectoria del papel, permiten el movimiento coordinado y preciso de los motores para que la hoja sea colocada en el contenedor correspondiente. Un sensor permite además detectar los niveles de papel de los contenedores y separadores del manejador [2].

2.3.3 Movimiento de papel en un manejador

El movimiento de papel en un manejador Hewlett Packard, se describe por un conjunto de trayectorias $T = \{T_1, T_2, \dots, T_n\}$ que determinan las posibles rutas que las hojas de papel pueden seguir desde su salida de la impresora, hasta el contenedor seleccionado del dispositivo.

A cada una de estas trayectorias de papel está asociado un perfil de movimiento específico M_i , necesario para transportar la hoja a través del manejador. Cada M_i es definido por un conjunto de movimientos simples m_k que pueden formar parte de cualquier trayectoria, $M_i = \{m_1, m_2, \dots, m_k\}$ y donde cada m_k a su vez, es producto de la acción asociada f_k de un motor con uno o varios sensores, $m_k = f_k \{e_1, e_2, \dots, e_n\}$.

Dados los m_k requeridos por estos dispositivos, movimientos con carga mínima, de alta precisión y control, los elementos e_n representan en su mayoría sensores detectores de posición que regulan el paso de las hojas a través del dispositivo; motores de pasos, encargados de girar rodillos en base a diferentes parámetros como velocidad y posición; y en menor proporción, motores de corriente directa que cumplen con la funcionalidad de mover los elementos mecánicos del sistema (apiladores de papel, engrapadora).

Motores de corriente directa.

Uno de los usos de un motor de corriente directa en un manejador de papel se muestra en la Figura 2-6. El motor está acoplado generalmente a un sistema de engranes o bandas que cambian la relación de velocidad, para trabajar en un punto de operación deseado y bajo los parámetros de operación óptimos del motor, de acuerdo a los datos que el fabricante proporciona. Estos engranes a su vez, transmiten el movimiento a una flecha, en donde se encuentran los rodillos transportadores.

Los rodillos proveen una fuerza proporcional a la atracción entre ambos elementos y el coeficiente de fricción del material con el que están elaborados. Bajo condiciones óptimas de operación, el movimiento angular de la flecha es transmitido en su totalidad al papel,

para que éste experimente un movimiento lineal. Así, la velocidad y aceleración a los que el papel está sometido son directamente controlados por el motor.

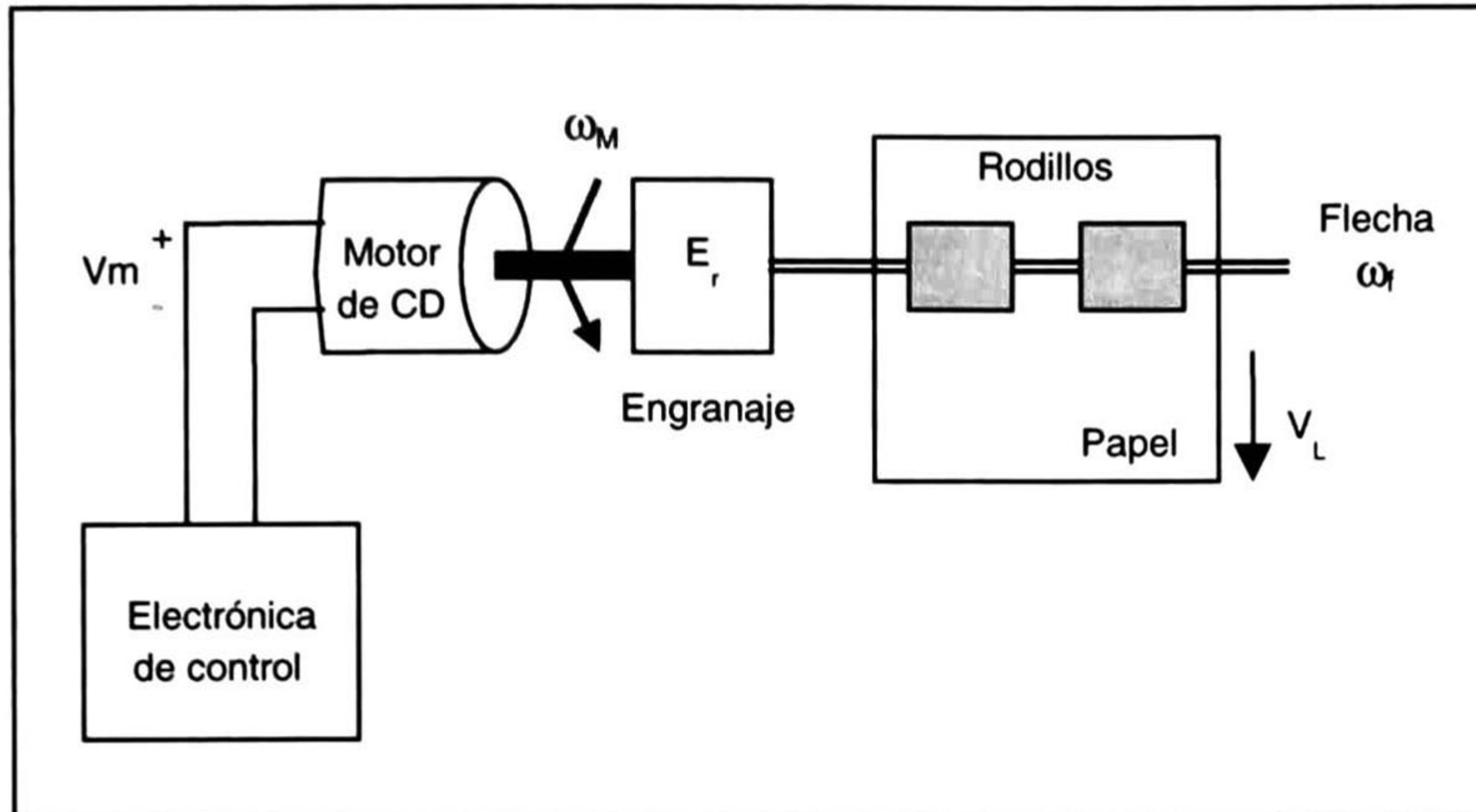


Figura 2-6 Movimiento de papel producido por un motor de corriente directa.

De acuerdo al esquema de la figura, la velocidad lineal (V_L) está representada por la ecuación $V_L = P\omega_f$, donde $P = 2\pi R_r$ constituye un factor de atracción ejercido por los rodillos de radio R_r sobre las hojas de papel, cuyo valor es multiplicado por la velocidad angular final de la flecha ω_f . Este parámetro depende de la relación de engranaje E_r y la velocidad angular del motor ω_M ($\omega_f = E_r\omega_M$) [4].

Dado que ω_M a su vez, es el producto del voltaje aplicado al motor y su constante de fabricación ($\omega_M = V_M K_M$), la velocidad lineal puede ser descrita de acuerdo a la siguiente ecuación:

$$V_L = 2\pi R_r E_r V_M K_M$$

El análisis anterior muestra la relación entre el voltaje aplicado al motor y la velocidad lineal final proporcionada. Esta velocidad es proporcional al movimiento lineal (M_L) del papel ejercido por el motor, al introducir un factor de tiempo, representado por la función escalón $\mu(t-t_0)$; así, el papel se mueve finalmente de acuerdo a la relación:

$$M_L = V_L \mu(t-t_0)$$

En este esquema, la electrónica de control aplica y regula el voltaje para obtener una velocidad lineal en el papel según se requiera.

La dirección del movimiento proporcionado por un motor de corriente directa está dada por la polaridad del voltaje aplicado, ya sea en uno u otro sentido.

Motor de pasos.

La utilización de un motor de pasos en manejadores de papel se muestra en la Figura 2-7. Con esta tecnología, la velocidad angular del rotor es proporcional a la frecuencia con que se le introducen pulsos eléctricos a las bobinas del motor. Al avance en radianes del rotor que es provocado por un pulso en las bobinas del estator se le denomina 'paso'. De acuerdo a la técnica de construcción del motor, el número de pasos por revolución del rotor varía. El resto del análisis de la velocidad y movimiento es similar al mostrado con los motores de corriente directa [5].

Sin embargo, dado que el voltaje V_M aplicado al motor, tal y como se menciona anteriormente, es ahora sustituido por una fuente de frecuencia f , la velocidad lineal del papel utilizando un motor a pasos esta dada por:

$$V_L = 2\pi R_r E_r f K_M$$

Dado que su factor de fabricación puede desglosarse de acuerdo a sus características de construcción por medio de la relación $K_M = 2\pi/n$, donde n representa el número de pasos por revolución en el rotor, la velocidad lineal del papel proporcionado por este dispositivo es:

$$V_L = 4\pi^2 R_r E_r f / n$$

La gran ventaja que presenta un motor a pasos frente a un motor de corriente directa es un control de la velocidad más preciso, sin la necesidad de un sistema de retroalimentación. Además, este tipo de dispositivo permite mantener un torque o fuerza constante aún cuando no se esté moviendo, con el propósito de fijar la posición de una hoja de papel en un punto dado, sin posibilidades de que se mueva o desajuste de su posición fácilmente.

En un esquema controlado por un motor a pasos, la dirección del movimiento está dada por la secuencia con la que los pulsos son introducidos en las bobinas del motor.

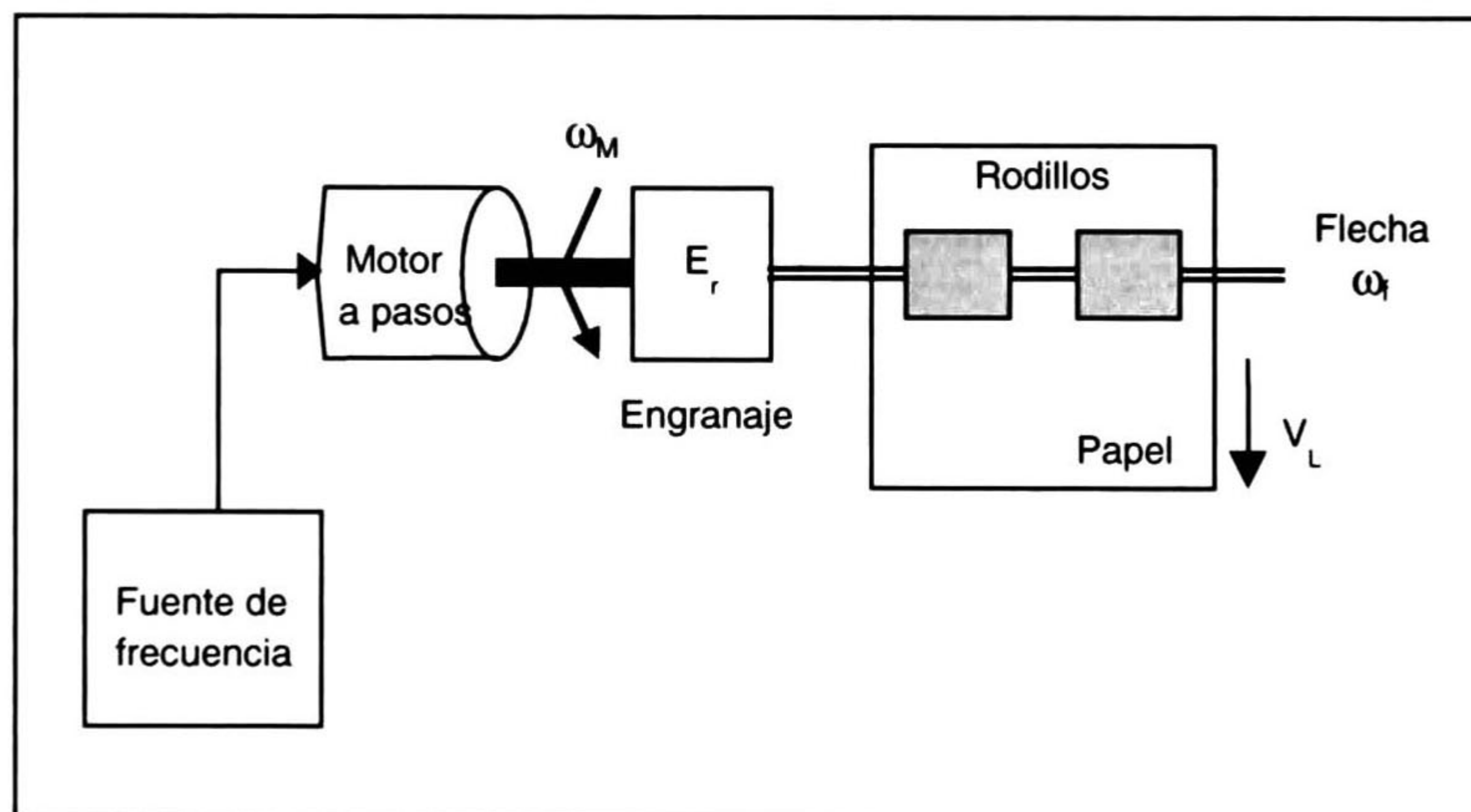


Figura 2-7 Movimiento de papel producido por un motor de pasos.

Perfiles de movimiento, velocidad y movimiento.

La Figura 2-8 muestra una gráfica de la relación entre aceleración, velocidad y movimiento que generalmente se transmiten a una hoja de papel en un manejador. De acuerdo a la figura, los perfiles de aceleración y desaceleración son constantes, pero en la realidad pueden tener diferentes formas. Estos parámetros proporcionan los niveles de velocidad lineal en el sistema encargados de generar el movimiento necesario para transportar el papel a través del dispositivo. El estudio y prueba de estos factores es de gran importancia durante el desarrollo de un manejador.

Cuando deseamos analizar los tiempos y posiciones del papel durante su transporte dentro del manejador, los relativos a la aceleración y desaceleración se desprecian debido a que son mucho menores que los de transporte, $[T_1, T_3, T_5] \ll [T_2, T_4]$.

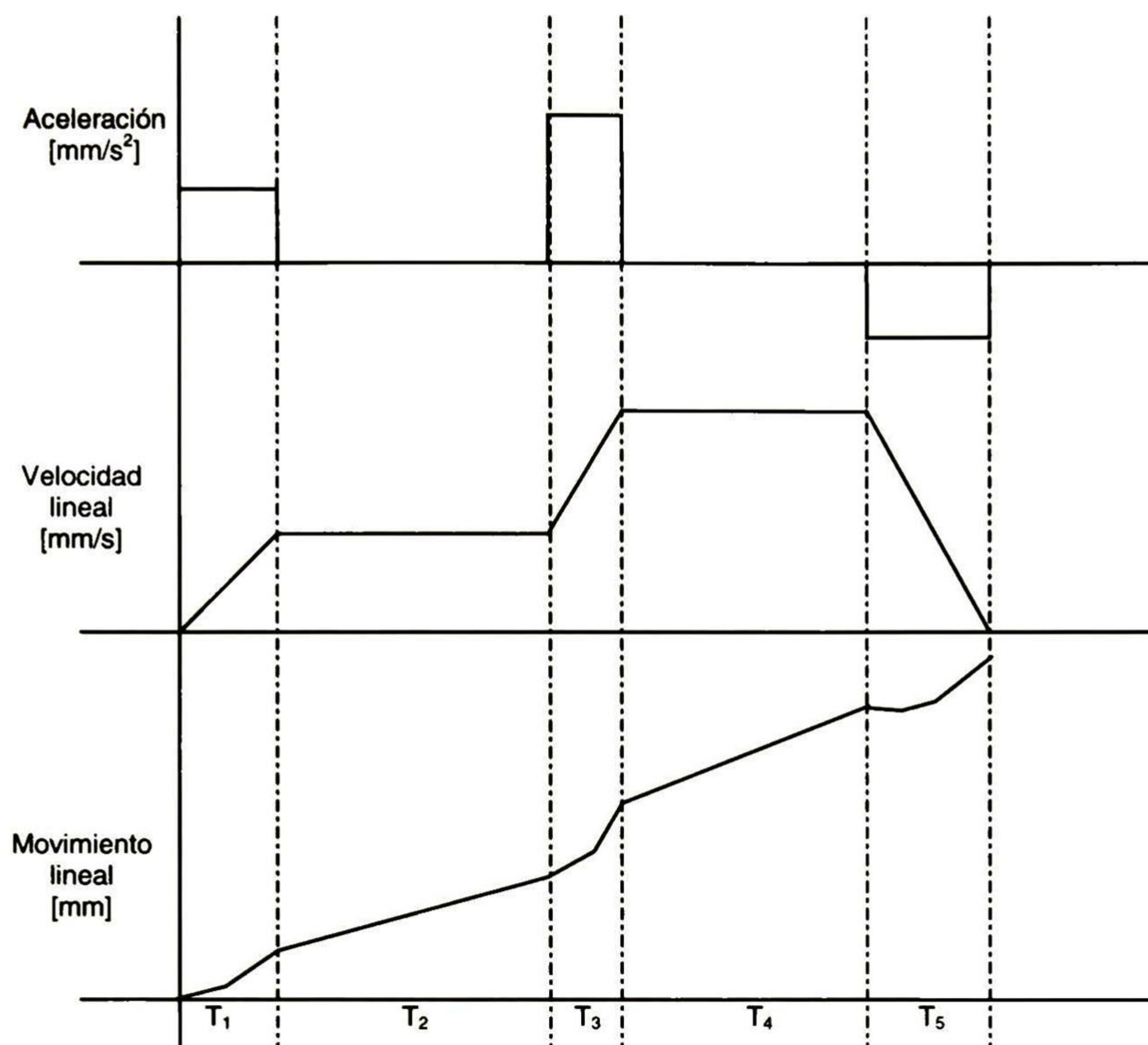


Figura 2-8 Relación entre aceleración, velocidad y movimiento de papel en un manejador.

2.3.4 Comunicación impresora-manejador de papel

Un manejador de papel requiere, entre otras cosas, detectar la posición de hojas, mover rodillos y seleccionar un contenedor de salida, para trabajar adecuadamente. Sin embargo, este dispositivo no actúa solo a partir de sus propios parámetros, sino que depende del controlador del mecanismo principal de impresión, para conocer las condiciones generales sobre las que trabaja.

El intercambio de información entre la impresora y el manejador de papel se implementa por medio de mensajes codificados, a través del protocolo de comunicación CAN 2.0B, cuyo principio de operación e implementación se abordará en capítulos posteriores.

Los principales mensajes de operación que complementan la interfaz del manejador con la impresora son:

Señal de encendido

En la Figura 2-9 se muestra el sistema de encendido de un manejador de papel. Al encenderse la impresora, su controlador envía la señal de encendido al manejador de papel que tiene conectado, éste, se energiza e inicializa. Este procedimiento permite que no se demande energía del sistema central por los dos dispositivos al mismo tiempo, sino que su encendido sea gradual.

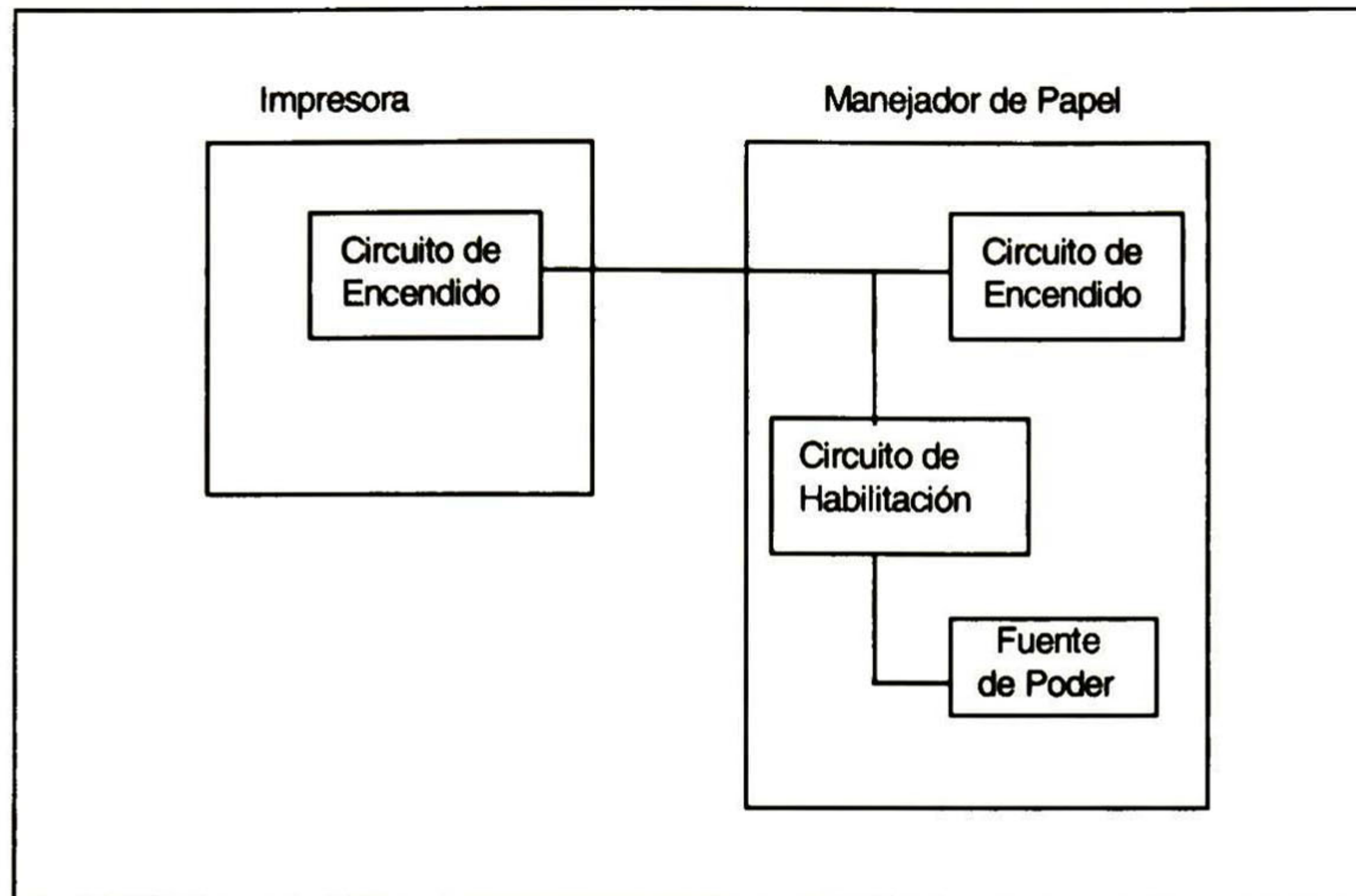


Figura 2-9 Sistema de encendido de un manejador de papel Hewlett Packard.

Secuencia de arranque

Describe la configuración y capacidad del dispositivo. Identificador, tamaños de papel, capacidad de hojas, número de contenedores, número de separadores, soporte para terminado de trabajo (engrapado).

Estado general

Describe el estado actual del manejador. Estos son: *activo* (en espera del próximo trabajo de impresión), *inactivo* (en bajo consumo de energía), *en operación*, *inactivo por error durante su operación*.

Recepción de una página

Activa el manejador de papel, y lo prepara para recibir la primera hoja del siguiente trabajo de impresión.

Manejo de errores

Mensajes básicos para el manejo de errores y reportes de mal funcionamiento. Errores de operación, puertas abiertas, atascamiento de hojas, contenedor lleno, entre otros.

Coordinación de las operaciones

Señalización que coordina el movimiento de papel en el manejador y describe sus parámetros de operación más importantes: inicio y fin de trabajo, tamaño de papel, contenedor o separador destino, opción de engrapado (número y posición de grapas).

3 Arquitectura

Una familia de productos o dispositivos está constituida por todos los elementos que dadas sus similitudes de operación y funciones, pueden ser diseñados de acuerdo a una misma arquitectura. Aprovechando este concepto surge una herramienta de desarrollo, que representa una plataforma para el diseño de cada nuevo producto miembro de una familia, sin olvidar sus características individuales.

Para el caso de los manejadores de papel, dispositivos basados en un sistema electromecánico, cuya operación depende de motores y ciertas señales (sensores), la reutilización de su estructura de control y sus manejadores de hardware es posible, ya que son diseñados y operados bajo un mismo esquema y sobre subsistemas electrónicos bien definidos.

Actualmente, la mayor parte de estos productos están implementados sobre un sistema de control programable, constituido por un microprocesador, componentes periféricos al procesador, y ciertos módulos de aplicación, que proporcionan la flexibilidad necesaria para desarrollar muy diversas tareas de propósito general, como manejo de componentes electromecánicos, monitoreo de señales, o la comunicación con otros dispositivos.

Una vez definida la plataforma común para el desarrollo de los manejadores, se complementa con una aplicación de software que permita la interacción del usuario con el sistema, a través de una interfaz gráfica. La función del programa es que el diseñador o ingeniero mecánico pueda configurar la herramienta, para probar la funcionalidad de los dispositivos, de acuerdo a su propia necesidad.

3.1 Especificación

El objetivo principal es realizar una herramienta que sea utilizada para interactuar con motores de pasos, motores de corriente directa y sensores, que permita el desarrollo y pruebas de los manejadores de papel Hewlett Packard en períodos de tiempo menores a los que llevaría su proceso común de diseño e implementación.

Con la finalidad de cumplir con el objetivo de desarrollar y probar el esquema de control de los dispositivos manejadores de papel, la arquitectura de la herramienta deberá contener los elementos necesarios para:

1. Mover motores de corriente directa y de pasos, de acuerdo a determinadas funciones.
2. Detectar por medio de sensores el estado de una hoja de papel durante su transporte en cualquier tiempo t .
3. Almacenar y ejecutar códigos programables, conformados por diferentes algoritmos.
4. Permitir la comunicación entre diferentes dispositivos de la plataforma de hardware, empleando un protocolo estándar.
5. Comunicarse con una PC, vía puerto serial, a una velocidad no menor que 9600 bps.
6. Proporcionar al usuario la capacidad para interactuar con la herramienta por medio de un software residente de una PC.

3.1.1 Perspectiva del producto

Representa el primer producto relacionado con una plataforma de desarrollo para el esquema de control de los manejadores de papel Hewlett Packard.

Debido a que la herramienta implementa las necesidades actuales de operación sobre estos dispositivos, a continuación pueden desarrollarse, en base a su arquitectura, nuevas versiones, que cumplan con futuros requerimientos.

3.1.2 Características del usuario

Los usuarios de este sistema serán ingenieros de diversas disciplinas encargados del diseño de manejadores, con conocimientos básicos de motores, pero no necesariamente expertos en los diversos algoritmos o métodos para controlarlos.

3.1.3 Estudio de viabilidad del sistema

Para el desarrollo de este producto, Hewlett Packard proporcionará todas las facilidades y recursos necesarios, como son: instalaciones para realizar el desarrollo y las herramientas de trabajo: computadora personal, recursos de búsqueda de información (internet, líneas telefónicas), compra de componentes, sistemas de cómputo para el diseño a detalle del hardware, fabricación y ensamble de prototipos, lenguajes de programación, compiladores, sistemas operativos.

3.1.4 Requerimientos específicos

La herramienta deberá construirse sobre una arquitectura general que pueda ser utilizada para el diseño, implementación y prueba de cada nuevo dispositivo manejador de papel.

A fin de representar una plataforma real de desarrollo para este tipo de producto, el sistema debe ser capaz de:

1. Debido a que el movimiento de papel en un manejador Hewlett Packard se describe por un conjunto de trayectorias T , el hardware debe ser capaz de proporcionar las señales necesarias para implementar los m_k (movimientos simples) de motores de corriente directa y de pasos, definidos por las ecuaciones ya descritas en el capítulo anterior, que hacen posible el transporte de las hojas a través del dispositivo.

- *Motores de corriente directa.*

Arrancar en una dirección, a un nivel de velocidad único.

Arrancar en la otra dirección, a un nivel de velocidad único.

Parar, independientemente de la dirección en que se mueva.

- *Motores de pasos.*

Arrancar en una dirección, de acuerdo a una secuencia de pasos (rampa) de aceleración inicial que determine su velocidad final.

Arrancar en la otra dirección, de acuerdo a una secuencia de pasos (rampa) de aceleración inicial que determine su velocidad final.

Moverse sobre otros dos diferentes niveles de velocidad, de acuerdo a rampas configurables de aceleración y desaceleración.

Parar, de acuerdo a una rampa de desaceleración final e independientemente de la dirección en que se mueva.

Dar 'n' número de pasos en una dirección.

Moverse repetitivamente 'n' número de pasos, en una y otra dirección.

2. Detectar por medio de sensores el estado de una hoja de papel durante su transporte.
3. Almacenar y ejecutar códigos programables a través de un sistema de control basado en un microprocesador y elementos periféricos. El procesador, elemento esencial que determinará en gran medida la capacidad de la herramienta, deberá cumplir con los siguientes criterios para su selección:
 - a) Soporte de un lenguaje de alto nivel.
 - b) Soporte de un sistema operativo de tiempo real.
 - c) Familia escalable, código compatible con micros de 8, 16 y 32 bits.
 - d) Arquitectura probada y con disponibilidad inmediata.
 - e) Hardware integrado:
 - Temporizadores.
 - Puertos seriales.
 - Máximo número de entradas y salidas de alta velocidad (retardo de 100ns).
 - f) Herramientas para su desarrollo:
 - Compiladores.
 - Emuladores.
 - Sistemas operativos de tiempo real.
 - g) Costo.
4. Debido a que los dispositivos manejadores de papel tienen una interacción directa con impresoras y otros dispositivos físicos, que hacen posible su operación, deberá implementarse el protocolo de comunicación CAN 2.0B, el cual ha sido adoptado por Hewlett Packard para cumplir con esta funcionalidad.
5. Comunicarse con una PC por medio de un puerto serial y mediante el protocolo RS232C, a una velocidad no menor a los 9600 bps.
6. Proporcionar la capacidad para interactuar con la herramienta, por medio de un software residente de una PC. Mediante esta aplicación, el usuario debe ser capaz de:
 - a) Conocer el estado actual del sistema, a través de mensajes explícitos.
 - b) Mostrar la información necesaria para configurar la plataforma física.
 - c) Configurar la operación de motores de acuerdo a los siguientes parámetros:
 - Tipo de motor (de pasos o de corriente directa).
 - Tipo de control, dirección, corriente, secuencias de pasos (rampas) de aceleración y desaceleración.
 - d) Ejercitar motores de corriente directa y de pasos, de acuerdo a los tipos de movimiento soportados por la herramienta.
 - e) Mostrar el estado de los sensores.

3.1.5 Restricciones generales

El hardware del sistema debe ser modular y debe cumplir con los requerimientos de control y operación de la nueva familia de manejadores de papel. Durante el desarrollo y diseño del hardware se deberán considerar varias opciones para la implementación de su funcionalidad básica. Toda la información a detalle del sistema debe manejarse por medio de un paquete de diseño de CAD (Mentor Graphics).

El firmware debe ser desarrollado en lenguaje C con un compilador específico para el procesador que forme parte del hardware, además debe cumplir con los lineamientos de codificación que la empresa maneja.

El software y el desarrollo de sus interfaces son a libre formato, el paquete a utilizar para su desarrollo debe correr en una PC bajo el sistema operativo de Windows. Para el desarrollo se puede utilizar Visual Basic o C++ Builder. La interfaz de usuario debe ser lo más intuitiva posible para que éste acceda a toda la funcionalidad de la herramienta.

La capa de protocolo de comunicación a utilizar sobre el canal serial es a libre elección; sin embargo, debe ser perfectamente documentada para así poder brindarle un mantenimiento futuro.

3.2 Análisis y diseño

De acuerdo a la especificación del sistema, la arquitectura de la herramienta de desarrollo para dispositivos manejadores de papel Hewlett Packard (Figura 3-1), se compone de tres bloques funcionales básicos: *hardware*, *firmware de control* y *software de aplicación*.

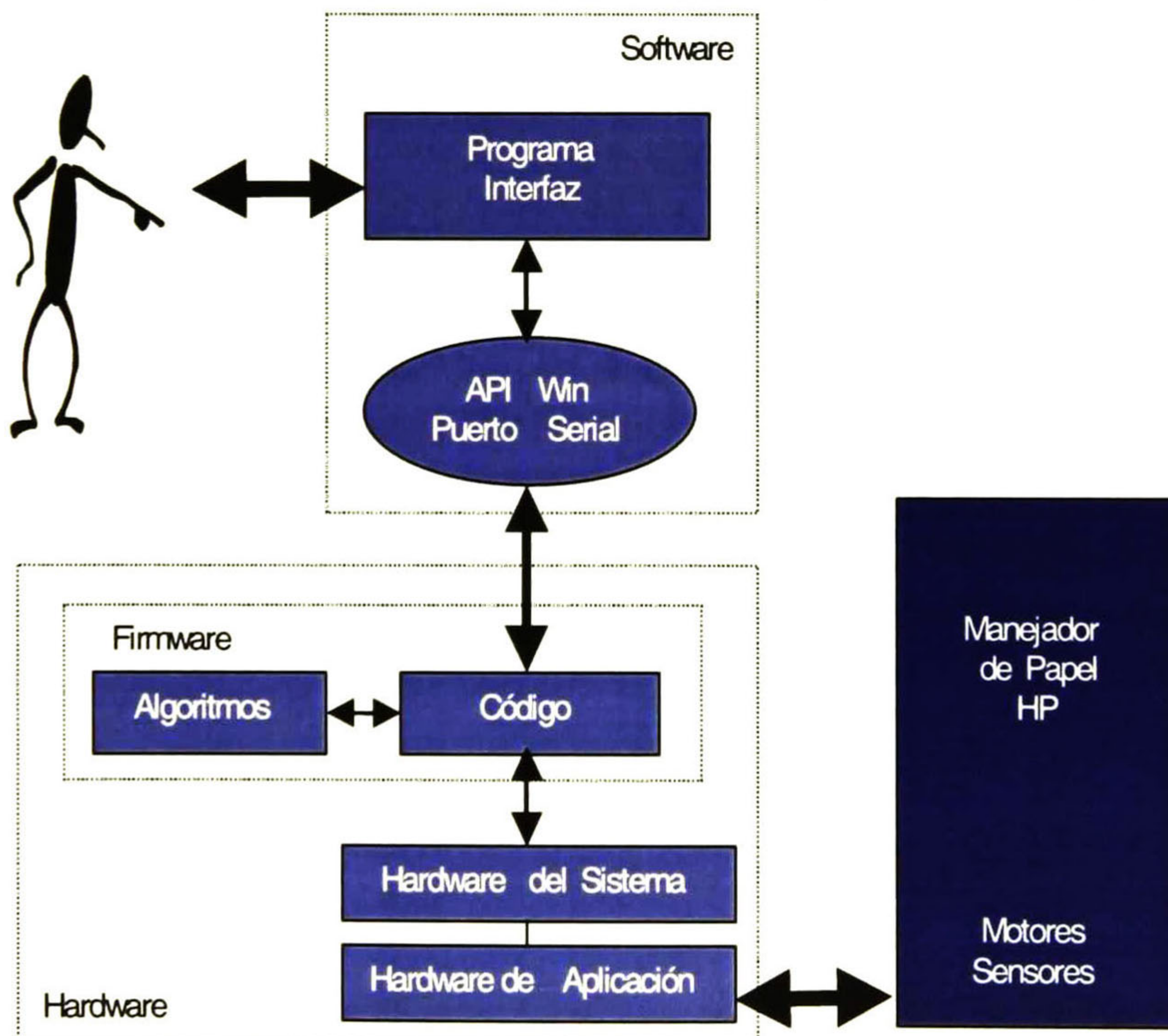


Figura 3-1 Arquitectura general de la herramienta para el desarrollo y pruebas de los manejadores de papel Hewlett Packard.

El bloque de *hardware*, está representado por varios elementos básicos como son: microprocesador, circuitos periféricos al procesador (memorias, interfaces de comunicación), manejadores de motores, electrónica para lectura de sensores.

El *firmware de control* se divide en dos componentes básicos. El primero proporciona los algoritmos para ejercitar los motores; este bloque de algoritmos será desarrollado y controlado por Hewlett Packard. El segundo componente consiste en un código que accede a los algoritmos para mover motores y, a su vez, se comunica al exterior a través de un puerto serial estándar a una velocidad no menor a 9600 bps. La información que este módulo maneja será capturada e interpretada por el tercer módulo funcional, el software.

El *software de aplicación* reside en una PC y, además de comunicarse por el puerto serial con el firmware, provee una interfaz gráfica para que un usuario interactúe con el sistema de acuerdo a sus propias necesidades.

3.2.1 Hardware

Proporciona los componentes físicos que hacen posible el movimiento de los motores y la lectura de los sensores que conforman un manejador de papel. Se compone de varios elementos como un microprocesador, circuitos periféricos al procesador (memorias, controladores), manejadores de motores (también conocidos como “drivers”), electrónica para lectura de sensores, circuitos de comunicación, entre otros [6].

Dada la concepción original de la herramienta, de conformar una plataforma para el desarrollo del esquema de control de diferentes productos, el hardware puede dividirse física y lógicamente en dos partes: hardware del sistema y hardware de aplicación.

3.2.1.1 Hardware del sistema

Elemento central de la plataforma, que constituye el sistema mínimo de control de la herramienta. Proporciona, con un microprocesador programable, la capacidad para implementar muy diversas tareas de propósito general. Se compone de varios elementos:

1. Microprocesador, parte esencial del esquema de control y que determina en gran medida los alcances y limitaciones de la herramienta.
 - a) Memoria FLASH para almacenamiento y ejecución de código.
 - b) Memoria RAM para almacenar datos.
 - c) Componentes y conectores que implementan la unión del hardware de sistema con la interfaz física y lógica del protocolo CAN 2.0B utilizado para la comunicación del manejador de papel con la impresora.
 - d) Controlador y conector que habilita la comunicación serial del microprocesador con la PC.
 - e) Lógica digital que implementa la interfaz entre los componentes del hardware del sistema.
 - f) Conector paralelo que constituye la interfaz física entre el hardware del sistema y el hardware de aplicación.
 - g) Componentes de señalización.
 - h) Conector para fuente de voltaje.

3.2.1.2 Hardware de aplicación

Implementa una función específica, en este caso, la generación de las señales físicas que hacen posible el movimiento de los motores y la lectura de los sensores del manejador de papel. Su diseño está determinado por la arquitectura del hardware del sistema, con quién se comunica a través de un conector. Los elementos que constituyen el hardware de aplicación son:

- a) Electrónica que acople y filtre señales provenientes de los sensores de un manejador de papel, para su lectura lógica y digital.
- b) Manejadores o 'drivers' de motores que transforman las señales digitales del hardware de aplicación a señales analógicas que alimentan a motores del manejador de papel.
 - 'Drivers' de propósito general.
 - 'Drivers' de propósito general con capacidad para manejo de alto nivel de corriente.
 - 'Driver' para motor de CD controlado por una señal modulada por ancho de pulso (PWM).
 - 'Driver' discreto para motor de CD de alta corriente.
 - 'Driver' controlado por un convertidor digital-analógico (D/A) externo.
 - 'Driver' controlado por un convertidor digital-analógico (D/A) interno
- d) Componentes de señalización.
- e) Conector para unirse al hardware del sistema.

3.2.2 Firmware de control

Representa la interfaz lógica entre el hardware y el software de aplicación de la herramienta. Se compone de un conjunto de instrucciones programables dirigidas al microprocesador, que hacen posible que éste genere las señales requeridas por los circuitos electrónicos periféricos, para llevar a cabo las operaciones deseadas en el sistema [7].

Se divide en dos componentes básicos: el primero proporciona un conjunto $A = \{A_0, A_1, \dots, A_n\}$ de algoritmos para ejercitar motores de corriente directa y de pasos, que implementan la funcionalidad necesaria para generar los movimientos simples m_k que definen cada uno de los perfiles de movimiento, encargados de transportar las hojas de papel sobre el conjunto T de trayectorias en un manejador. El segundo componente consiste en un código que accede al conjunto A para mover motores y, a su vez, se comunica al exterior a través de un puerto serial estándar a una velocidad mayor a 9600 bps.

La funcionalidad deseada del sistema se implementa en base a comandos de operación provenientes del software de aplicación. Estos comandos pasan a través del manejador del puerto serial del procesador, y una vez identificados son decodificados a fin de determinar las acciones que se deben ejecutar para cumplir con su objetivo. Después de recibir un comando, el bloque de firmware debe responder, indicando el éxito o fallo de su ejecución.

El firmware de aplicación provee, en respuesta a los comandos transmitidos por el software y recibidos por el puerto serial, la funcionalidad necesaria para ejecutar, sobre cada uno de los componentes electromecánicos, las siguientes acciones:

- Sensor.
 - Informar del nivel lógico del sensor.
- Motor de corriente directa.
 - Informar el estado del motor.
 - Arrancar motor.
 - Parar motor.
- Motor de pasos.
 - Informar el estado del motor.
 - Arrancar motor.
 - Parar motor.
 - Dar un determinado número de pasos.
 - Cambiar velocidad de motor.
 - Configurar rampas de aceleración.
 - Configurar rampas de desaceleración.

La Figura 3-2 muestra un diagrama que especifica la estructura de operación del firmware de control.

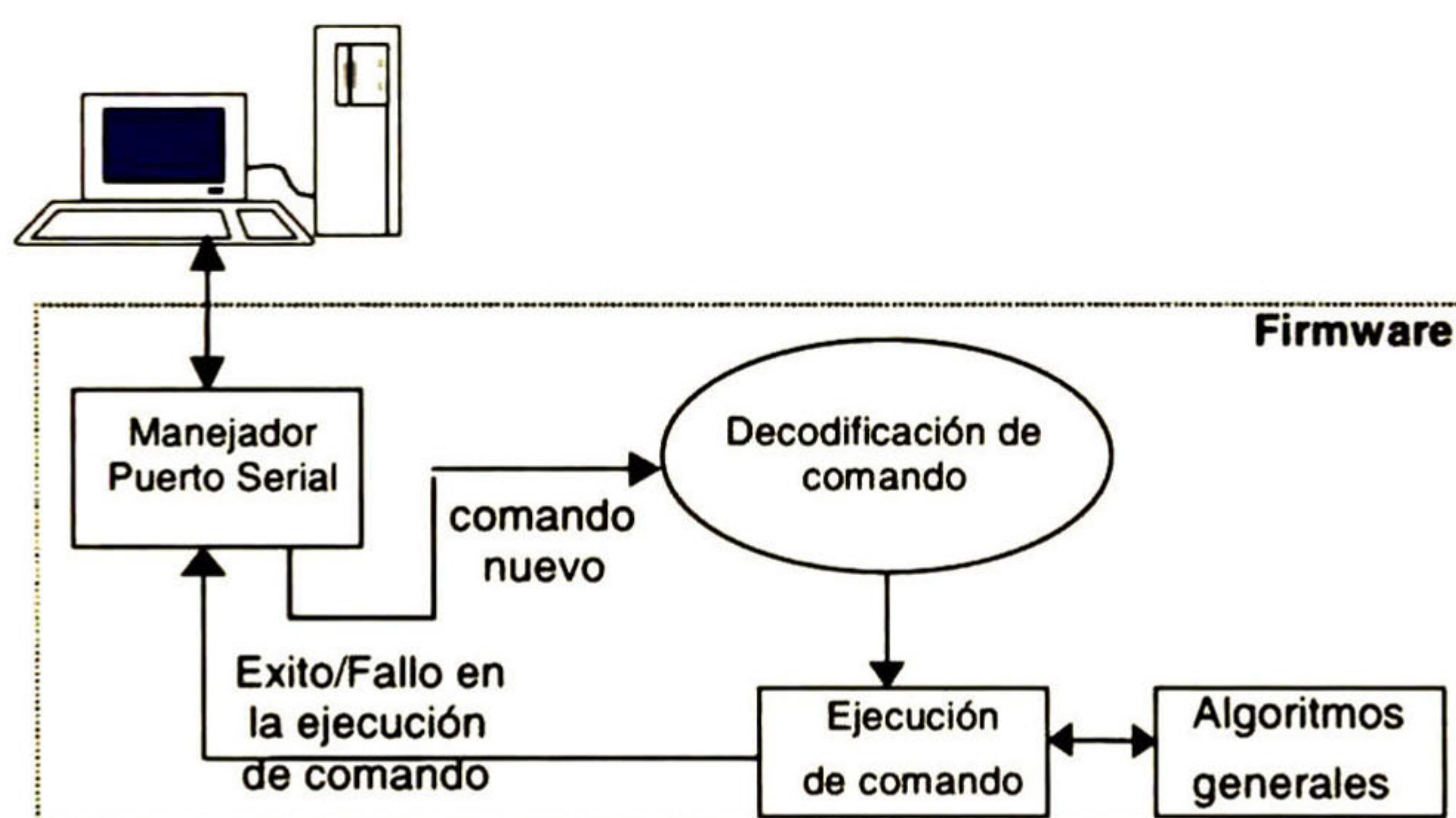


Figura 3-2 Estructura de operación del firmware de control.

3.2.3 Software de aplicación

Debe ofrecer al usuario una interfaz con la plataforma física de la herramienta, para mover motores de pasos y motores de corriente directa o conocer el estado de sensores.

Por medio de un programa residente en una PC, que se ejecuta sobre el sistema operativo Windows 95/98/NT, el usuario configura, de acuerdo a su necesidad, los parámetros necesarios para controlar el funcionamiento de los componentes electromecánicos de un manejador.

La información obtenida por el programa es codificada y transmitida hacia el firmware de control para que ejecute la operación requerida. Así, el programa ofrece al usuario la capacidad para manejar:

- Sensores, con detección constante del nivel lógico de cada señal.
- Motores de corriente directa, donde cada motor puede ser operado independientemente de acuerdo a los siguientes parámetros:
 - Tipo de control.
 - Dirección.
 - Nivel de corriente.
- Motores de pasos operados de acuerdo a los siguientes parámetros configurables:
 - Dirección.
 - Nivel de corriente.
 - Niveles de velocidad diferentes.
 - Capacidad para dar un cierto número de pasos especificado por el usuario.
 - Rampas de aceleración.
 - Rampas de desaceleración.

La interfaz con el firmware se crea a través de un protocolo de comunicación basado en comandos que indican las acciones que el hardware debe ejecutar con relación a los motores y sensores utilizados por el usuario. El programa cuenta con su propio manejador para el puerto serial de la PC, implementado a partir del API de Windows 95.

4 Hardware

En el presente capítulo se describe el hardware que forma parte de la herramienta para el desarrollo y pruebas del esquema de control de los manejadores de papel Hewlett Packard. Este módulo del sistema representa la plataforma física, conformada por un conjunto de elementos, que constituyen este tipo de dispositivos y que proporcionan la funcionalidad deseada por el usuario.

El hardware proporciona los componentes que hacen posible el movimiento de los motores y la lectura de los sensores que forman parte de un manejador de papel. Para esto se vale de un grupo de elementos como un microprocesador, circuitos periféricos al procesador (memorias, controladores), manejadores de motores (también conocidos como "drivers"), electrónica para lectura de sensores, circuitos de comunicación.

Esta integrado por dos tarjetas de circuitos impresos denominadas: hardware del sistema y hardware de aplicación.

4.1 Hardware del sistema

Elemento central de la plataforma de desarrollo, que constituye el sistema mínimo de control de la herramienta. Proporciona, sobre la base de un microprocesador programable, la capacidad para implementar muy diversas tareas de propósito general.

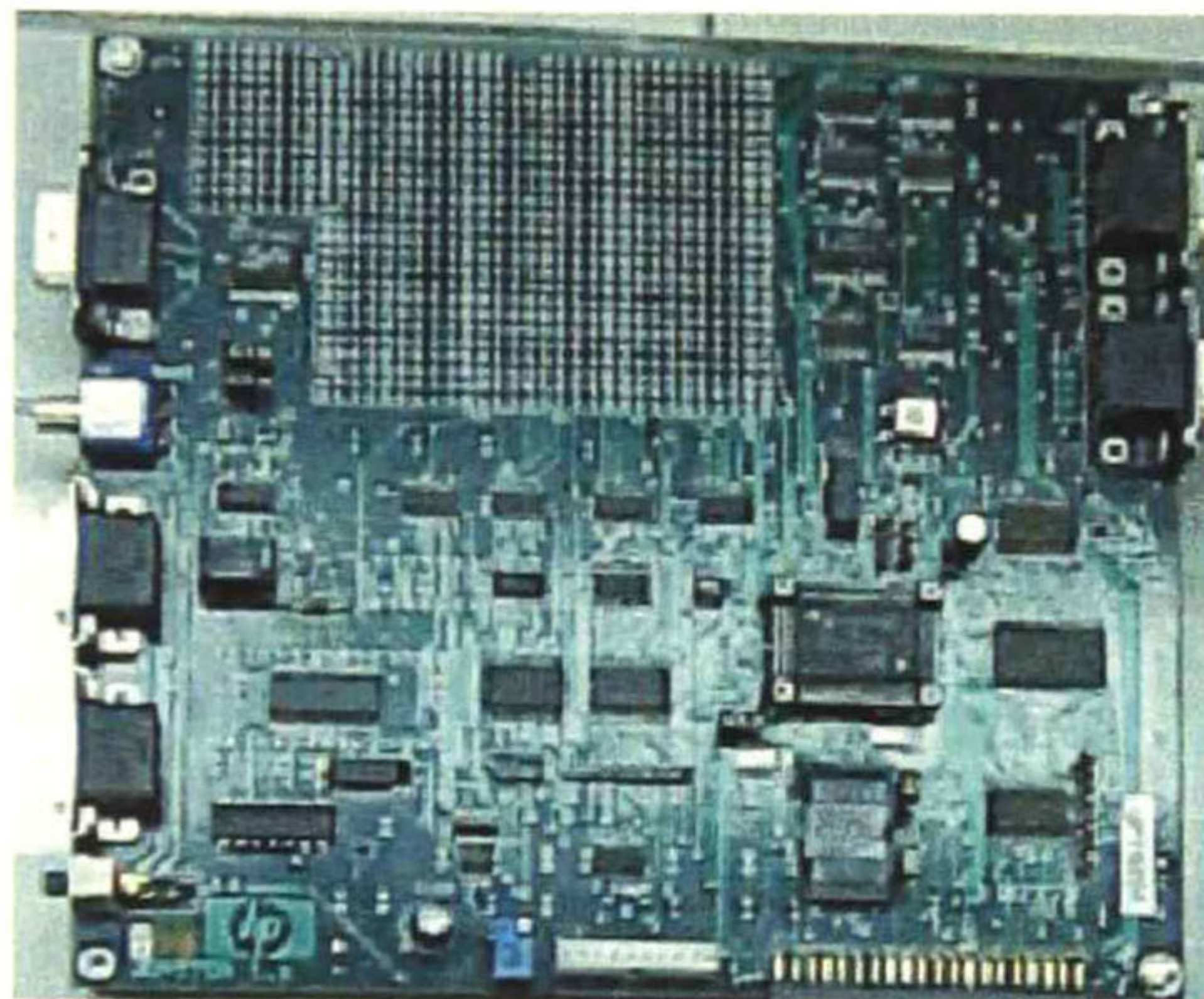


Figura 4-1 Tarjeta del hardware del sistema.

El diseño del hardware del sistema, está definido por la selección del microprocesador y componentes periféricos, su requerimiento de funcionalidad e interfaz con otros sistemas. Se conforma de los siguientes elementos:

- Microprocesador de la familia H8S/23xx de Hitachi, de alta eficiencia, basados en una arquitectura de 16 bits.
- Memoria FLASH de 2 Mbits, organizada en 128 Kbytes x 16 bits.
- Memoria RAM estática de 512 Kbits, organizada en 32 Kbytes x 16 bits.
- Memoria no volátil (EEPROM serial) de 512 bytes.
- Controlador del protocolo de comunicación CAN 2.0B.
- Canal de comunicación serial (RS232C).
- Interfaz con el hardware de aplicación.

El diagrama a bloques de la Figura 4-2 describe la arquitectura del sistema. Muestra cada uno de los bloques funcionales que lo componen y la forma en que se relacionan entre ellos.

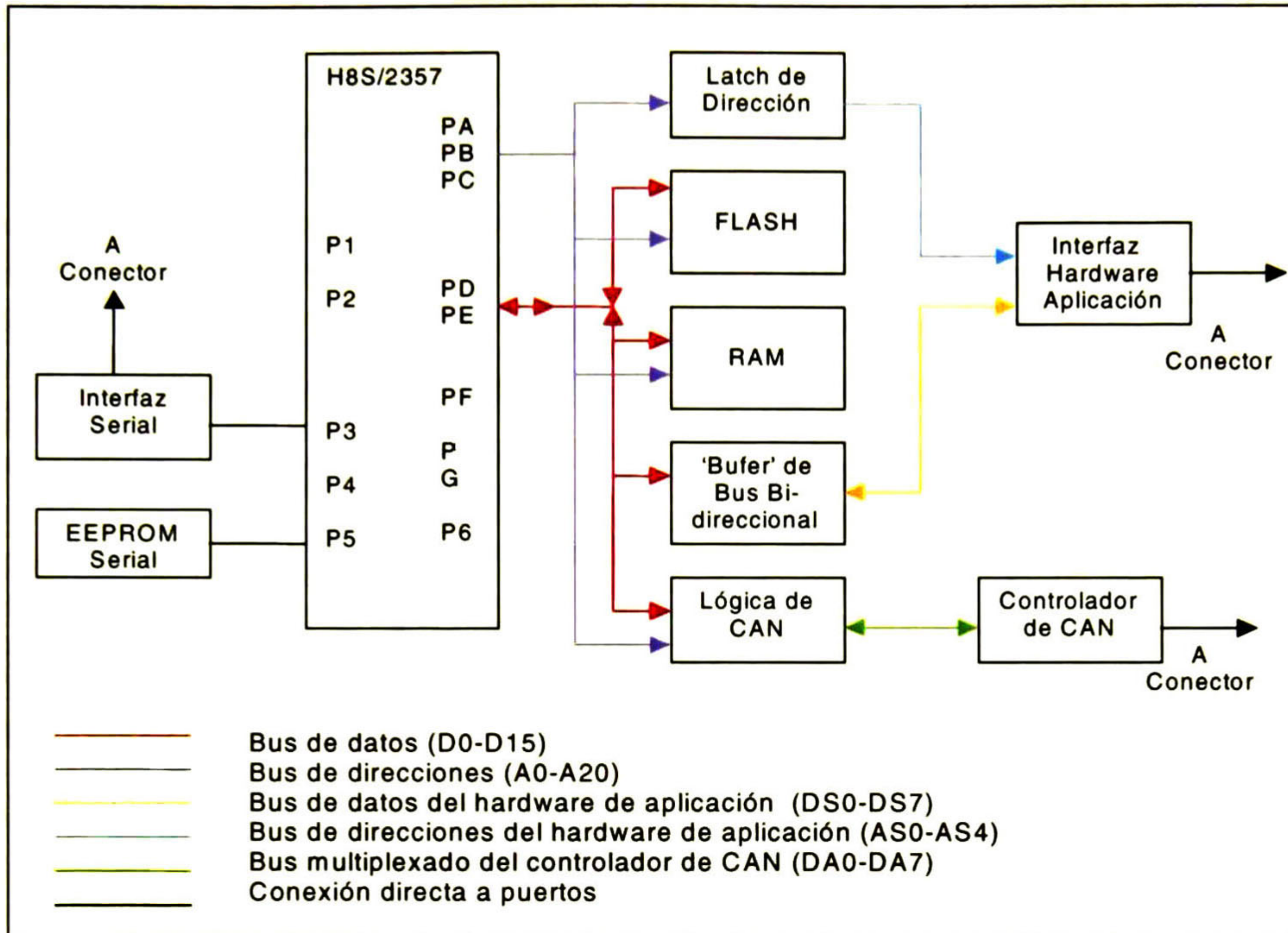


Figura 4-2 Arquitectura del hardware del sistema.

La descripción de cada bloque se basa en el desarrollo de los siguientes puntos:

1. Microprocesador y circuitos externos.
2. Protocolo de comunicación CAN 2.0B.
3. Comunicación serial.
4. Conexión con el hardware de aplicación.
5. Fuente de alimentación.

4.1.1 Microprocesador y circuitos externos

El hardware del sistema basa su operación en el microprocesador de Hitachi H8S/2357. La selección del procesador está basada en los siguientes parámetros:

- a) Escalabilidad (8, 16 bits).
- b) Eficiencia y frecuencia.
- c) Buen soporte de herramientas (compiladores, sistemas operativos, emuladores).
- d) Hardware integrado.
- e) Arquitectura de programación.
- f) Precio y disponibilidad.

El H8S/2357 tiene memoria ROM integrada, sin embargo es compatible con la versión más económica H8S/2352 sin ROM. Sus circuitos externos son diseñados de acuerdo a las especificaciones del microprocesador y las recomendaciones de Hitachi [8]. A continuación se describen sus principales características y los elementos periféricos que lo conforman.

4.1.1.1 Microprocesador H8S/2357

El microprocesador H8S/2357 pertenece a la familia H8S/23xx de Hitachi, basada en una arquitectura de 16 bits. Los recursos internos con los que cuenta son:

- a) 10 puertos de entrada y salida configurables a múltiples funciones.
- b) 8 interrupciones externas enmascarables, con manejo de prioridad. Una interrupción no enmascarable.
- c) 3 canales de comunicación serial.
- d) 6 canales de temporizadores configurables para varias opciones.
- e) 2 temporizadores de comparación.
- f) Soporte de acceso dinámico a memoria (DMA, por sus siglas en inglés).
- g) Temporizador Watchdog.
- h) 8 canales de convertidor analógico-digital.
- i) 2 canales de convertidor digital-analógico [8].

4.1.1.2 Generación de reloj

El reloj es generado utilizando el oscilador interno del H8S/2357. La frecuencia de operación la determina un cristal estándar de 16 MHz, que satisface la eficiencia requerida por el firmware de control. La Figura 4-3 muestra la implementación del reloj del sistema.

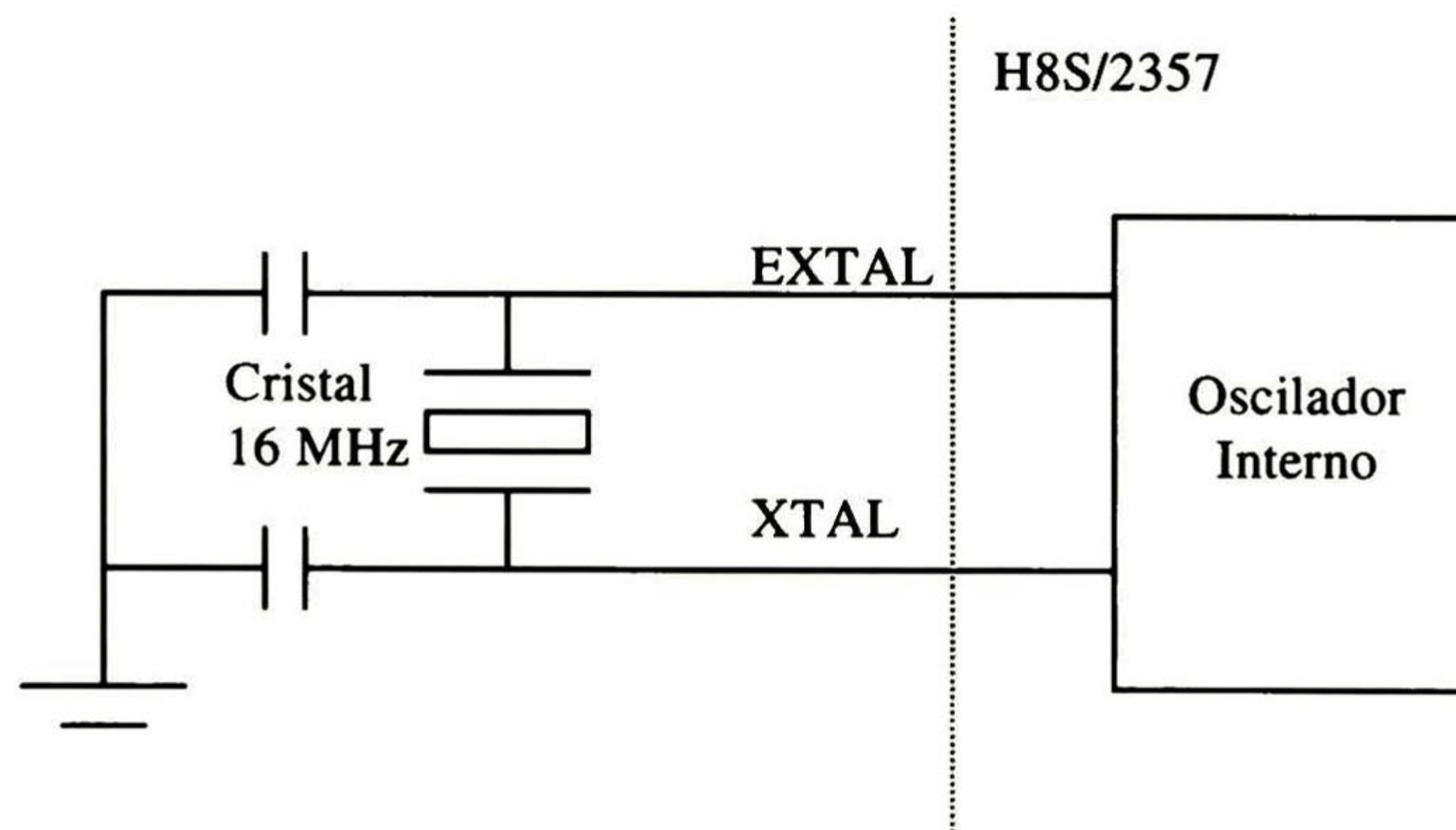


Figura 4-3 Fuente de frecuencia del H8S/2357.

4.1.1.3 Circuito de reinicio (reset)

Un circuito monitor (DS1813) es conectado al pin de reinicio del microprocesador H8S/2357 con dos propósitos:

- a) Mantener el pin en nivel bajo, al menos 20 ms después del encendido del sistema, para asegurar el correcto reinicio del H8S/2357.
- b) Asegurar el correcto funcionamiento del procesador después de accionar el botón de reinicio, de acuerdo a las especificaciones del H8S/2357.

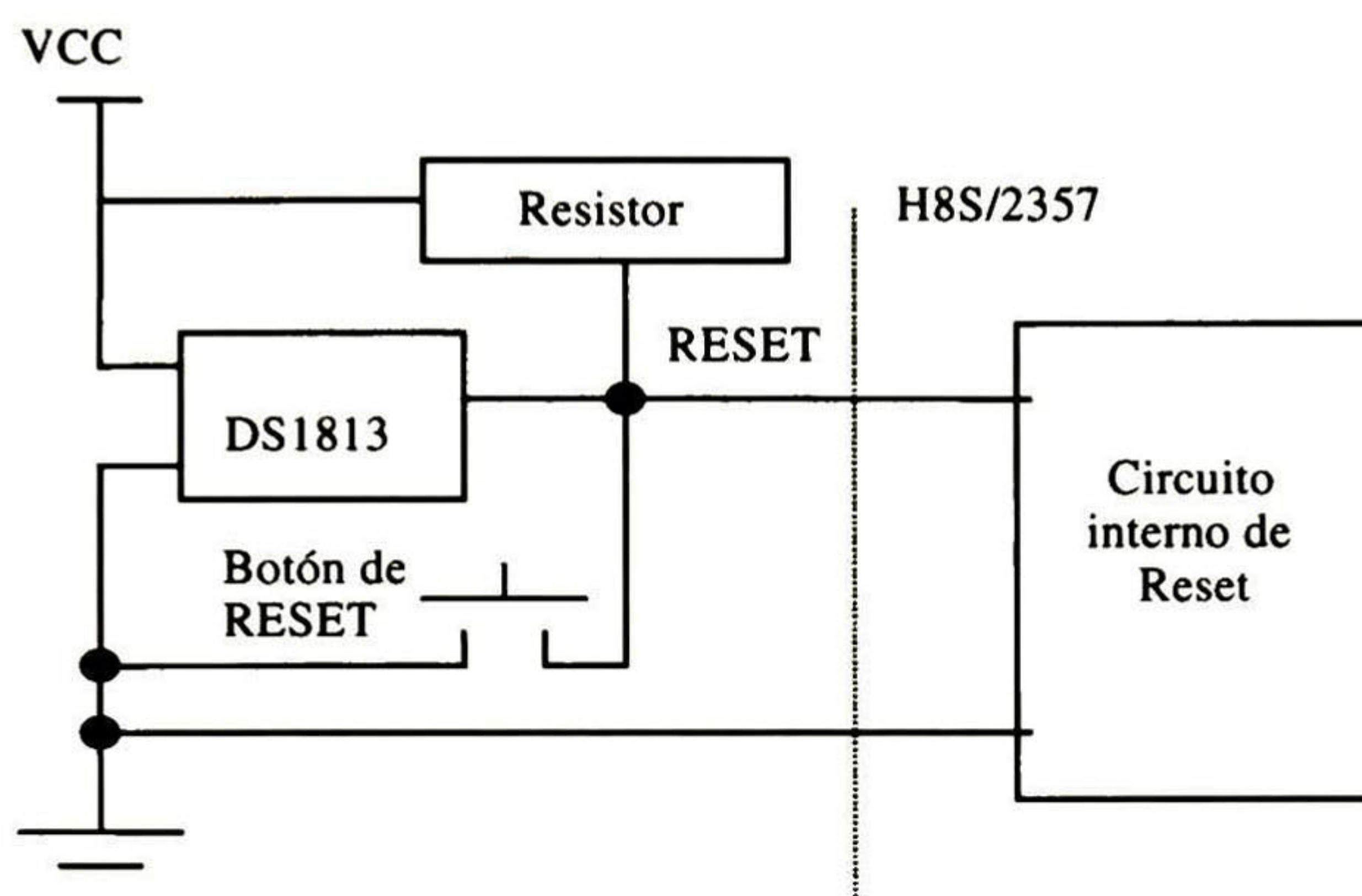


Figura 4-4 Circuito de reinicio del H8S/2357.

4.1.1.4 Puertos

El diseño del hardware del sistema basa su operación en la utilización de los puertos configurables del H8S/2357. La Tabla 4-1 muestra la función implementada por cada puerto del procesador como parte de la plataforma de hardware del sistema:

Puerto	Tipo	Aplicación
PA.0-PA.4	Bus	Señales A16-A20 del bus de direcciones
PA.5-PA.7	Entrada	Interrupciones: IRQ5, IRQ6 e IRQ7
PB.0-PB.7	Bus	Señales A8-A15 del bus de direcciones
PC.0-PC.7	Bus	Señales A0-A7 del bus de direcciones
PD.0-PD.7	Bus	Parte baja del bus de datos de 16 bits
PE.0-PE.7	Bus	Parte alta del bus de datos de 16 bits y bus de datos de 8 bits
PF.0	Ent/Sal	Ninguna
PF.1	Ent/Sal	Ninguna
PF.2	Entrada	Señal -Self Test- proveniente de switch de encendido manual
PF.3	Control	Señal -LWR- para escritura de memoria externa
PF.4	Control	Señal -HWR- para escritura de memoria externa
PF.5	Control	Señal -RD- para lectura de memoria externa
PF.6	Control	Señal -AS- para control de accesos a memoria externa

Puerto	Tipo	Aplicación
PG.0	Salida	Señal –MCRST- para reinicio manual del controlador de CAN
PG.1	Control	Señal –CS3- mapeo de área 3 de memoria externa
PG.2	Control	Señal –CS2- mapeo de área 2 de memoria externa
PG.3	Control	Señal –CS1- mapeo de área 1 de memoria externa
PG.4	Control	Señal –CS0- mapeo de área 0 de memoria externa
P1.0	Salida	Encendido/apagado de LED
P1.1-P1.3	Ent/Sal	Ninguna
P1.4	Entrada	Entrada de alta velocidad
P1.5	Entrada	Entrada de alta velocidad
P1.6	Entrada	Entrada de alta velocidad
P1.7	Entrada	Entrada de alta velocidad
P2.0-P2.3	Ent/Sal	Ninguna
P2.4	Salida	Señal de PWM
P2.5	Ent/Sal	Ninguna
P2.6	Salida	Señal de PWM
P2.7	Ent/Sal	Ninguna
P3.0	Salida	Transmisor del canal 0 para comunicación serial
P3.1	Salida	Transmisor del canal 1 para comunicación serial
P3.2	Salida	Receptor del canal 0 para comunicación serial
P3.3	Salida	Receptor del canal 1 para comunicación serial
P3.4	Salida	Señal –MotEnable- que controla activación de motores
P3.5	Entrada	Señal –IntEnable- que controla habilitación del PLD
P4.0-P4.5	Entrada	Canales del convertidor analógico-digital
P4.6-P4.7	Entrada	Canales del convertidor digital-analógico
P5.0	Salida	Reloj serial (SCL) para la EEPROM
P5.1	Ent/Sal	Direcciones y datos del serial (SDA) para la EEPROM
P5.2	Salida	Señal –PowerOut- para encendido de otro dispositivo
P5.3	Entrada	Señal –ADTRG- que activa el convertidor A/D interno
P6.0	Control	Señal –CS4- mapeo de área 4 de memoria externa
P6.1	Control	Señal –CS5- mapeo de área 5 de memoria externa
P6.2	Ent/Sal	Ninguna
P6.3	Ent/Sal	Ninguna
P6.4	Entrada	Señal –CAN_INT- interrupción del controlador de CAN
P6.5	Ent/Sal	Ninguna
P6.6	Control	Señal –CS6- mapeo de área 6 de memoria externa
P6.7	Entrada	Señal –PLD_INT- interrupción del PLD

Tabla 4-1 Utilización de puertos del H8S/2357.

4.1.1.5 Bus de datos y de direcciones

La función de bus de datos es implementada por los puertos D y E. En accesos a memoria externa configurada como 16 bits, el puerto E representa la parte baja del bus y el puerto D su parte alta; en accesos a 8 bits, sólo el puerto D es utilizado.

El bus de direcciones se compone por una parte del puerto A (parte más significativa del bus) y por los puertos B y C (direcciones menos significativas). Cuando el acceso a memoria externa es configurado a 16 bits, la dirección menos significativa del bus no es utilizada.

4.1.1.6 Dispositivos de entrada/salida mapeados a memoria

El espacio de memoria externa (16 Mbytes) es utilizado para el acceso a memorias y componentes externos. La opción de decodificación interna del microprocesador es utilizada para dividir el espacio de memoria en 8 áreas de 2 Mbytes cada una y habilitar las señales de selección correspondientes a cada área (CS0-CS7). La memoria externa es utilizada para código y datos. El código es almacenado y ejecutado desde una memoria Flash de 2 Mbits configurada como 128 Kbytes x 16 bits (Intel 28F200B5) y dividida en 5 bloques de memoria independientes. Está conectada a los buses de datos y direcciones del microprocesador y es controlada por las señales de lectura (RD) y escritura (HWR). Esta memoria soporta ciclos de lectura y bajo cierto estado, de borrado y escritura. Requiere de la señal de reinicio proporcionada por el microprocesador para que al energizarse, se asegure siempre su estado en modo de lectura.

Los datos son almacenados en 512 Kbits de memoria RAM estática (SRAM), organizados como 32 Kbytes x 16 bits y conectada a los buses de datos y direcciones del procesador. Es controlada por las señales de lectura (RD), y escritura (LWR, HWR).

La Tabla 4-2 describe los dispositivos de entrada/salida mapeados a memoria del H8S/2357; únicamente las áreas 0 y 1 son accesadas con un bus de datos de 16 bits, el resto es configurada para accesos a 8 bits.

Señal de decodificación	Rango de direcciones.	Elemento externo.
CS0	0x000 000-0x1FF FFF	128 Kbytes x 16 bits de memoria Flash.
CS1	0x200 000-0x3FF FFF	32 Kbytes x 16 bits de memoria RAM.
CS2	0x400 000-0x5FF FFF	Controlador de CAN.
CS3	0x600 000-0x7FF FFF	Señales de sensores, hardware de aplicación.
CS4	0x800 000-0x9FF FFF	Manejadores de motores, hardware de aplicación.
CS5	0xA00 000-0xBFF FFF	No es utilizado.
CS6	0xC00 000-0xDFF FFF	No es utilizado.
CS7	0xE00 000-0xFFF FFF	No utilizado, dirección de registros internos.

Tabla 4-2 Dispositivos de entrada/salida mapeados a memoria.

4.1.2 Protocolo de comunicación CAN 2.0B

Los dispositivos manejadores de papel Hewlett Packard, requieren llevar a cabo el control de un determinado número de procesos: sensado de variables, configuración de elementos, activación y desactivación de motores, relacionados con el movimiento de papel en el producto. Sin embargo, una de las principales características de este tipo de dispositivos es que no es encuentran aislados, sino que tiene una interacción directa con impresoras y otros elementos auxiliares para el manejo de papel.

Debido a esto, cada manejador requiere establecer un tipo de comunicación que le permita conocer las condiciones externas a él mismo, sobre las cuales trabaja. El protocolo de comunicación CAN 2.0B ha sido adoptado por Hewlett Packard para definir la interfaz entre los manejadores de papel y las impresoras que desarrolla.

La herramienta de desarrollo ha integrado a su esquema los elementos necesarios para cumplir con las especificaciones de este protocolo estándar, a fin de cubrir con cada aspecto relacionado a los dispositivos que maneja.

4.1.2.1 CAN (Controller Area Network)

La red para control de área (CAN por sus siglas en inglés), es un protocolo de comunicación serial que fué originalmente diseñado para la industria automotriz, pero que actualmente se ha convertido en un protocolo muy popular en procesos de automatización y otras aplicaciones; es ampliamente utilizado en un gran número de sistemas de control, pero principalmente en aquéllos cuya red de comunicación se establece a través de microprocesadores. Físicamente se constituye por un bus de dos cables que tienen la característica de establecer a través de ellos una comunicación bidireccional (half duplex) a muy alta velocidad. Entre sus principales beneficios se encuentran su robustez, su confiabilidad y el respaldo que ha obtenido, en los últimos años, por parte de la industria de semiconductores.

Teóricamente, puede enlazar hasta 2032 dispositivos en una misma red, sin embargo, dadas las limitaciones prácticas del hardware, solo puede soportar hasta 110 nodos. Este bus ofrece enlaces de comunicación hasta de 1Mbit/s, lo que permite el control de sistemas en tiempo real. Además, dada su probabilidad de error, y la capacidad para la detección de los mismos, CAN es un sistema confiable para trabajar en ambientes con altos niveles de ruido [9].

El protocolo está definido por el estándar ISO 11898 y sus principales características son:

- La capa física utiliza la transmisión diferencial en un par de cables.
- Utiliza una secuencia de bits arbitraria no destructiva para controlar el acceso al bus.
- Los mensajes son pequeños (ocho bytes de datos como máximo) y son protegidos por el método de verificación CRC (Cyclic Redundancy Check).
- No existe una dirección específica en sus mensajes, en su lugar, cada mensaje contiene un valor numérico que controla su prioridad en el bus, además de servir como una identificación del contenido del mensaje.
- Manejo de un esquema elaborado para el manejo de errores que da como resultado en la retransmisión de mensajes que no son apropiadamente recibidos.
- Métodos efectivos para detectar las fallas de conexión, que determinan o no la deshabilitación de un nodo en particular.

4.1.2.2 Principio de operación

Los mensajes de datos transmitidos por cualquiera de los nodos conectados al bus CAN no contienen direcciones y tampoco la identidad del nodo que lo transmite o el nodo destino. En su lugar, el contenido del mensaje es etiquetado por un identificador que es único a través de la red. Una vez liberado el mensaje, todos los nodos en la red reciben el mensaje y cada uno de ellos lleva al cabo un proceso de prueba con el identificador, para determinar si el mensaje, y su contenido, es relevante para cada uno en particular. Si el mensaje es relevante, es procesado, de lo contrario se ignora [9].

El identificador único, también determina la prioridad del mensaje. Mientras más bajo sea el valor numérico del identificador, mayor es su prioridad. Esto permite el control de tráfico del bus, en caso de que dos o más nodos compitan por el acceso al mismo tiempo.

Este control de acceso al bus garantiza que el mensaje con la mayor prioridad, aparezca como el único mensaje que está siendo transmitido. Los mensajes de menor prioridad, mientras tanto, son automáticamente retransmitidos en el siguiente ciclo del bus, o en un ciclo subsecuente si existen todavía otros mensajes de mayor prioridad esperando a ser transmitidos.

A continuación se describe, en base al sistema de comunicación de una impresora y un manejador de papel Hewlett Packard de la Figura 4-5, el principio de funcionamiento básico del protocolo CAN.

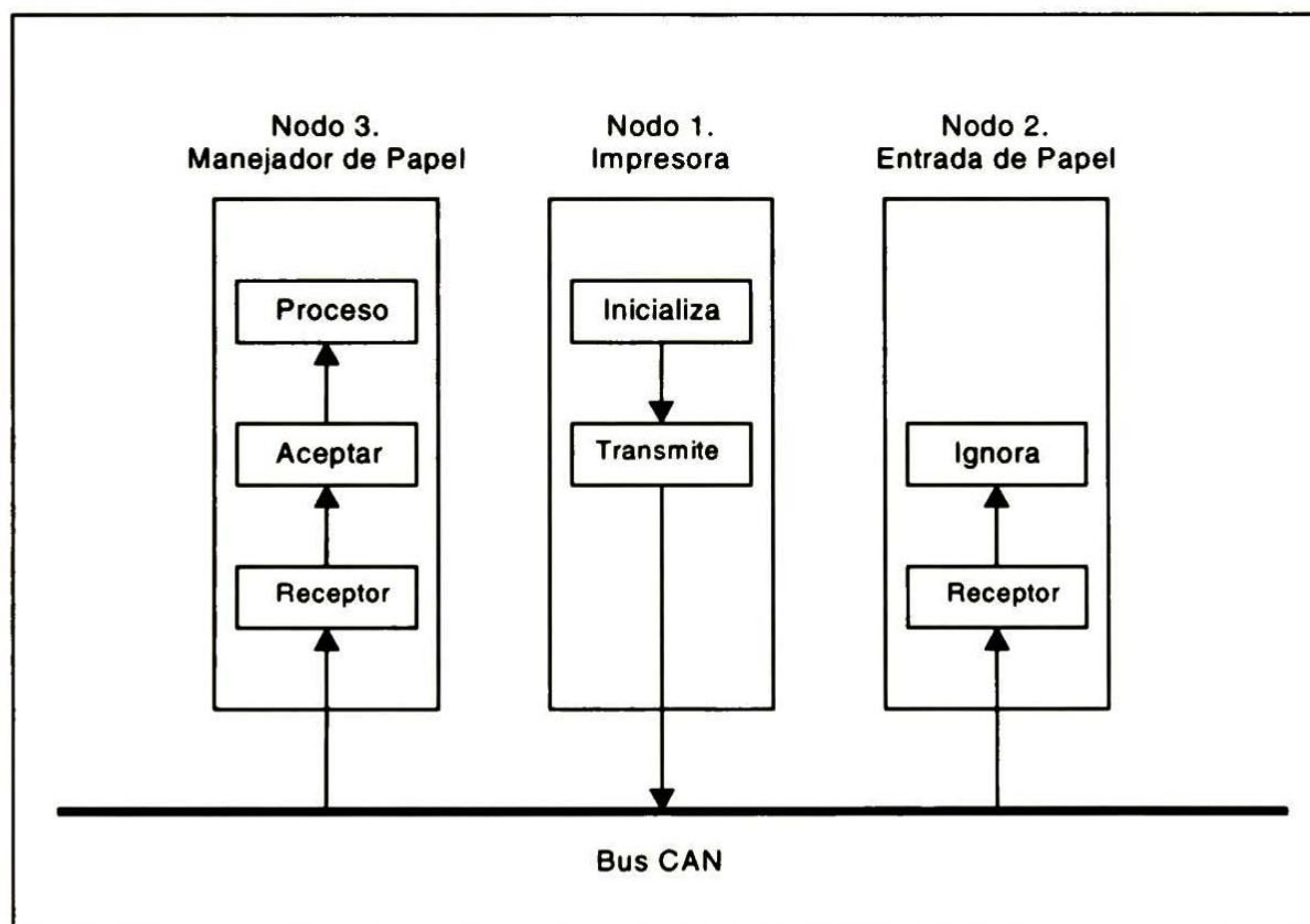


Figura 4-5 Transmisión de mensajes entre una impresora y un manejador por medio del protocolo CAN.

1. El bus CAN contiene tres nodos, el nodo 1 lo representa el controlador principal de la impresora, el nodo 2 lo representa un componente auxiliar, encargado de la entrada de papel a la impresora, mientras que el nodo 3 es conectado a un manejador de papel conectado a la salida del dispositivo de impresión.
2. El controlador en el nodo 1 transmite un mensaje, que indica que una hoja ya impresa, se dispone a salir de la impresora hacia el manejador de papel.

3. El mensaje y su identificador, son pasados hacia el circuito con el controlador de CAN integrado del nodo 1. Este circuito construye el mensaje, de acuerdo con el protocolo y lo transmite sobre el bus.
4. El mensaje transmitido por el nodo 1 convierte automáticamente a los nodos 2 y 3 en receptores.
5. Enseguida, cada uno de estos nodos lleva a cabo una evaluación de la información contenida en el mensaje, para determinar si es relevante para su elemento de control o no.
6. Si el mensaje no es relevante para un nodo, éste lo ignora (por ejemplo, para el nodo 2, ya que este solo controla la entrada de papel).
7. Por el contrario, si la información si tiene relevancia, el mensaje es aceptado y procesado, en este caso por el nodo 2 que inmediatamente activa el dispositivo para recibir la hoja de papel.

4.1.2.3 Formato de mensajes

En un sistema CAN, los mensajes de datos son transmitidos utilizando tramas. Estas tramas llevan la información desde el nodo transmisor hacia uno o mas nodos receptores.

El protocolo CAN está compuesto por dos tipos de formatos para sus tramas:

- CAN estándar (versión 2.0A).
- CAN extendido (versión 2.0B).

La mayoría de los controladores transmiten y reciben solamente mensajes con el formato estándar, sin embargo pueden recibir también el formato extendido (conocido como 2.0B pasivo), aunque esos mensajes son ignorados. Los controladores 2.0B pueden mandar y recibir mensajes en ambos formatos [10].

CAN estándar (2.0A)

Un mensaje basado en tramas estándar se muestra en la Figura 4-6. Este tipo de trama se compone de siete diferentes campos de bits:

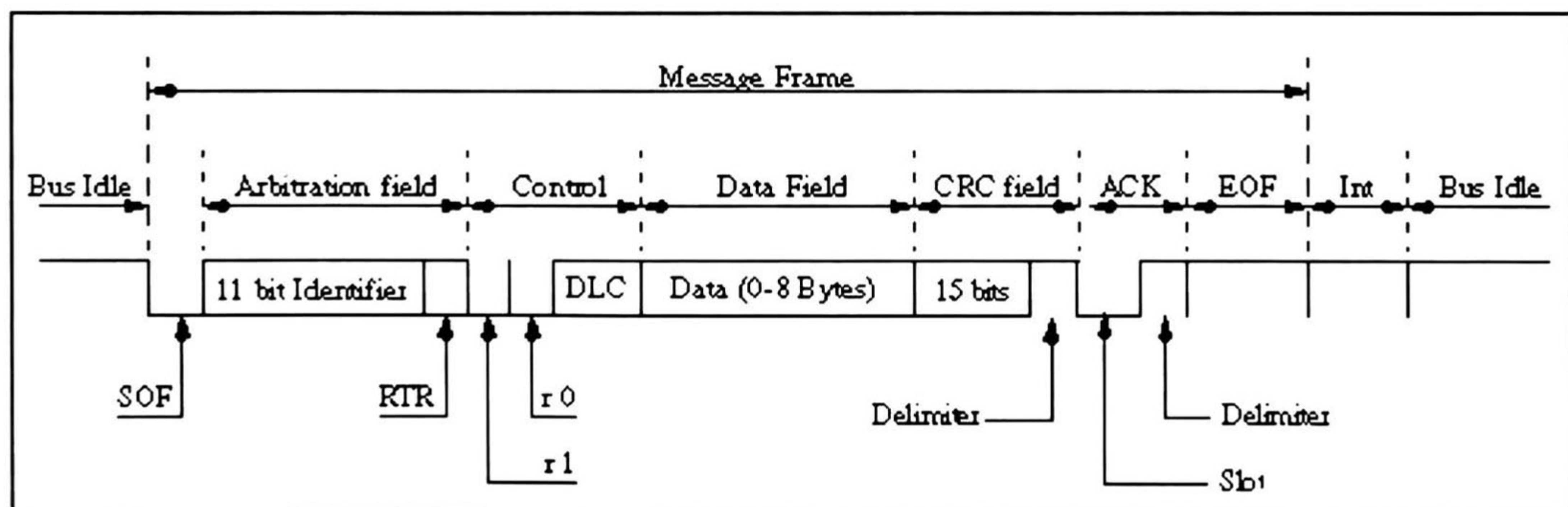


Figura 4-6 Mensaje con el formato CAN 2.0A

- Campo de inicio de trama (Start Of Frame, SOF). Bit dominante (0 lógico) que indica el inicio de una trama. La detección de un nivel de bit dominante a cualquier tiempo durante el estado desocupado del bus es interpretado como un inicio de trama.

- Un campo arbitrario, que contiene un identificador de mensaje de 11 bits y el bit de petición para transmisión remota (Remote Transmission Request, RTR). Un bit dominante (0 lógico), RTR indica que el mensaje es una trama de datos. Un valor recesivo (1 lógico) indica que el mensaje es una petición para una transmisión remota. Una trama remota es una petición de datos por parte de un nodo, para otro nodo en el bus. Las tramas remotas no contienen el campo de datos. El código de longitud de datos especifica el número de datos, en bytes, de la trama requerida.
- Un campo de control compuesto de 6 bits:
 - Dos bits dominantes (r0, r1), son reservados para un uso futuro.
 - Código de longitud de datos (Data Length Code, DLC), que indica el número de bytes en el campo de datos de la trama.
- Campo de datos, que contiene desde cero hasta un máximo de 8 bytes. Como en todos los bytes de las tramas de CAN, el bit más significativo de el primer dato es transmitido primero (el de más a la izquierda).
- El campo CRC, contiene 15 bit en codificación cíclica y redundante y un bit recesivo que limita al campo.
- El campo de ACK, consiste de dos bits. El primero de ellos es un bit de ranura o separación (slot) transmitido como un bit recesivo, pero que subsecuentemente es sobrescrito por bits dominantes transmitidos por los otros nodos que reciben el mensaje. El segundo es un bit recesivo que limita el campo.
- Campo de fin de trama, que consiste de siete bits recesivos (EOF).

CAN extendido (2.0B)

El formato de las tramas de mensajes CAN 2.0B, que se muestra en la siguiente figura, consta de veintinueve (29) bits para el identificador, a diferencia de los once (11) en 2.0A.

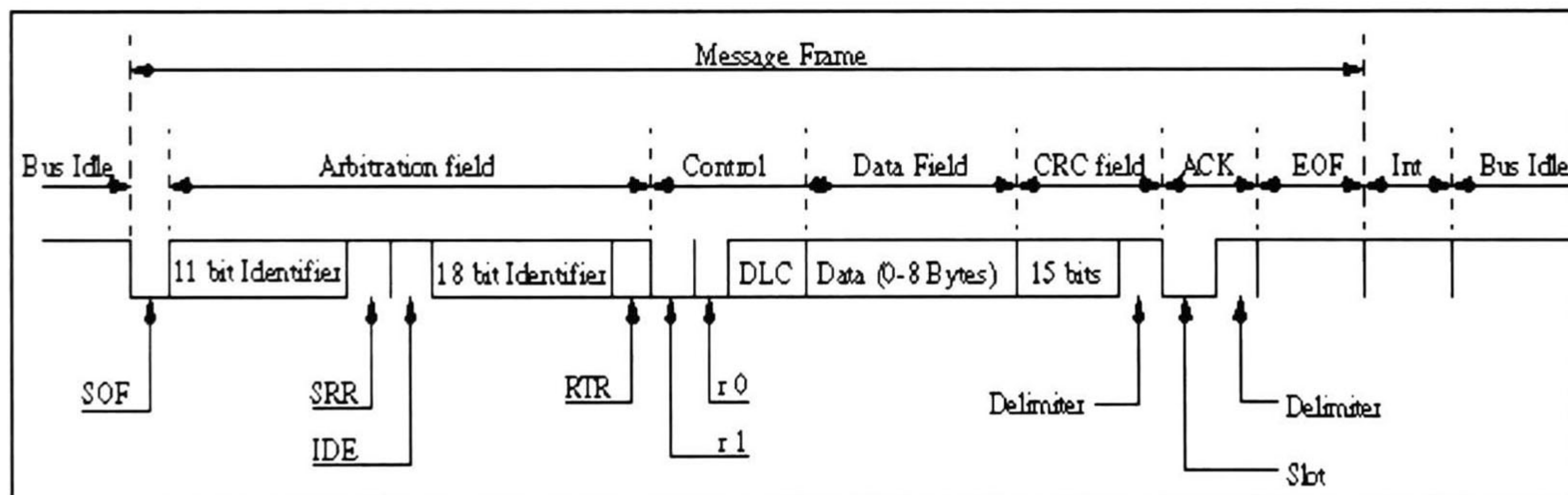


Figura 4-7 Mensaje con el formato CAN 2.0B.

La versión 2.0B está enfocada hacia la interacción con otros protocolos de comunicación seriales utilizados en algunas aplicaciones de la industria automotriz en Estados Unidos. Debido a esto este formato, maneja lo que se conoce como formato extendido, en donde las tramas además de contar con esta característica, son compatibles con las de la versión 2.0A. Las diferencias entre los dos formatos son:

- El campo arbitrario de la versión 2.0B contiene dos campos de bits para el identificador (ID). El primero (el ID básico) consta de 11 bits, para su compatibilidad

con la versión 2.0A. El segundo campo (el IDE extendido) es de 18 bits, para dar un total de 29 bits utilizados como identificador.

- El reconocimiento del formato de una trama se determina mediante un bit identificador de extensión. Este es transmitido como un bit recesivo cuando se mandan mensajes en el formato 2.0B y como un bit dominante cuando lo hace en el formato estándar (2.0A).
- El bit de petición para sustitución remota (Substitute Remote Request, SRR) es incluido en el campo arbitrario. Este bit es siempre transmitido como un bit recesivo para asegurar que, en caso de una arbitrariedad entre una trama de datos estándar y una trama de datos extendida, la trama estándar sea la que tenga una mayor prioridad si ambos mensajes tienen el mismo identificador básico de 11 bits. Note que, tal y como sucede en la versión 2.0A, el bit RTR es todavía el dominante para la transmisión de una trama y recesivo para la transmisión de una trama remota.
- Todos los demás campos en una trama de la versión 2.0B son idénticos a los del formato estándar.

Debido a que ambos formatos, generalmente coexisten en un mismo bus, los mensajes en formato estándar siempre tienen una mayor prioridad que los mensajes del formato extendido, cuando ambos contengan el mismo identificador básico (11 bits).

Para asegurar la mayor compatibilidad entre los controladores de diferentes fabricantes, el diseño de los sistemas CAN no debe utilizar campos de identificadores, tanto en 2.0A como en 2.0B, con los siete bits más significantes en 1 (1111111XXXX, identificador no válido). Esto significa que el número de identificadores únicos válidos para el usuario, en una red con el formato 2.0A se reduce a 2,032 ID diferentes. Sin embargo, para el caso de la utilización de un bus con el uso exclusivo del formato 2.0B, el número de identificadores válidos excede los 500 millones [10].

4.1.2.4 Implementación

El diseño e implementación de CAN 2.0B es parte de la especificación de la herramienta de desarrollo para la nueva familia de productos manejadores de papel. Este sistema tiene la finalidad de proveer la capacidad de establecer un medio eficiente de comunicación entre un manejador y el controlador de manejo de papel del sistema central de las impresoras.

Actualmente existen en el mercado, un gran número de controladores que implementan este protocolo, sin embargo el controlador SJA1000 de Philips representa la mejor opción, en cuanto a funcionalidad y operación. La única desventaja que presenta el SJA1000 es que el diseño de su interfaz con un microcontrolador se basa en las familias de procesadores Intel y Motorola, lo que obliga a emplear lógica adicional para adecuar la señalización del microprocesador de Hitachi utilizado.

A continuación se hace una breve descripción de los puntos más relevantes acerca del SJA1000, su principio de operación, la interfaz necesaria con el bus de CAN y su compatibilidad con el microcontrolador H8S/2357. El objetivo es presentar la manera en que el CAN 2.0B es implementado sobre la nueva plataforma de desarrollo, la forma en que opera y los requerimientos necesarios para su función.

El SJA1000 es un controlador diseñado para trabajar sobre toda clase de ambientes industriales, aunque su principal aplicación sigue siendo en la industria automotriz. Es el sucesor del controlador PCA82C200, ambos diseñados por la compañía Philips

Semiconductors. La principal diferencia entre estos dos, es que el SJA1000 soporta además del modo de operación básico (CAN 2.0A) el modo de operación PeliCAN, el cuál implementa las especificaciones necesarias para soportar el formato extendido del protocolo (CAN 2.0B).

El SJA1000 cuenta, como antes se menciona, con dos diferentes modos de operación:

- Modo BasicCAN (compatible con el PCA82C200).
- Modo PeliCAN.

Algunas de las principales características del SJA1000 son:

- Compatibilidad con el PCA82C200.
- Búfer de recepción extendido (64 bytes), que permite almacenar hasta 21 mensajes.
- Compatible con el protocolo CAN 2.0A y CAN 2.0B.
- Soporta identificadores de 11 y 29 bits.
- Bit rate de hasta 1 Mbit/s, capaz de soportar aplicaciones de alta velocidad.
- Características del modo PeliCAN:

Límites de advertencia de error, programables.

Interrupciones para cada error diferente en el bus de CAN.

Interrupción de pérdida arbitraria de comunicación que permite la optimización del sistema.

Modo de monitoreo, utilizado para analizar el tráfico en el bus o para la detección automática del bit rate.

Detección por software del bit rate en el bus.

Filtros de aceptación de mensajes.

Recepción de mensajes controlado.

- Frecuencia de operación igual a 24 MHz.
- Interfaz con una gran variedad de microprocesadores.
- Configuración de interfaz de salida programable, que permite su relación todo tipo de capa física.
- Rango de operación a temperaturas de -40 a +125 grados centígrados.

Generalmente cada módulo o nodo de CAN puede ser dividido en bloques que cumplen con una diferente funcionalidad. La conexión con las líneas del bus se implementa generalmente a través de un transmisor-receptor optimizado para cada aplicación. Este transmisor-receptor controla y regula los niveles lógicos de las señales que vienen del controlador y que van hacia la capa física, para que presenten fielmente sus respectivos niveles físicos de voltajes en el bus. De igual forma lleva a cabo la interfaz entre estos dos niveles de voltajes en la dirección inversa a la antes descrita [11].

El siguiente nivel es el controlador de CAN el cuál implementa completamente el protocolo definido en su especificación. La mayoría de las veces también ofrece la posibilidad de llevar a cabo el filtrado de mensajes.

Todas estas funciones, sin embargo son controladas por un módulo controlador, el cuál implementa la funcionalidad propia de la aplicación. Por ejemplo, controla actuadores, lee sensores y maneja la interfaz hombre-máquina.

La Figura 4-8 representa la forma en que se estructura cada uno de éstos módulos o nodos del bus CAN antes descrita, y los elementos de la plataforma de hardware que en este caso cumplen con cada función.

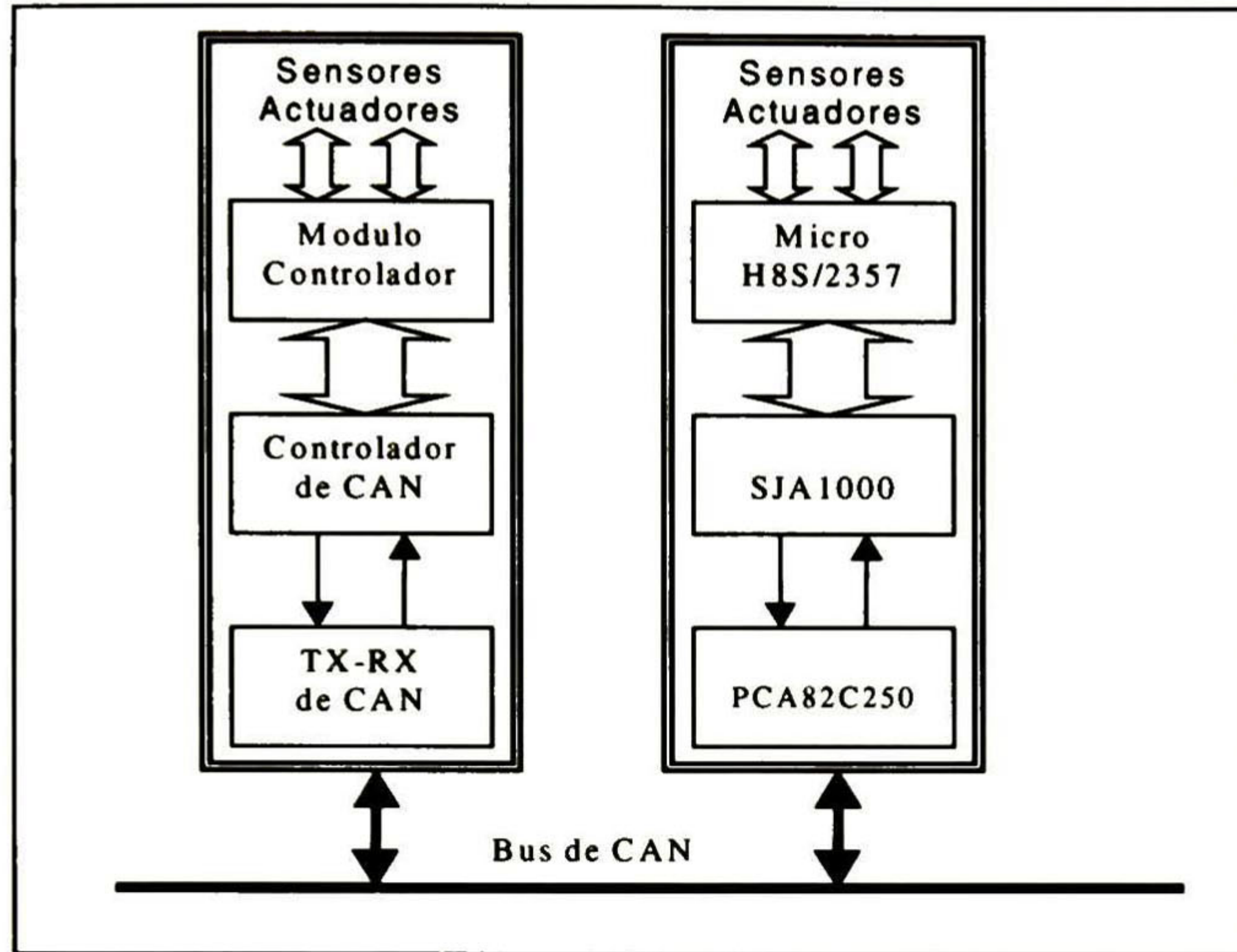


Figura 4-8 Arquitectura del nodo en el bus CAN.

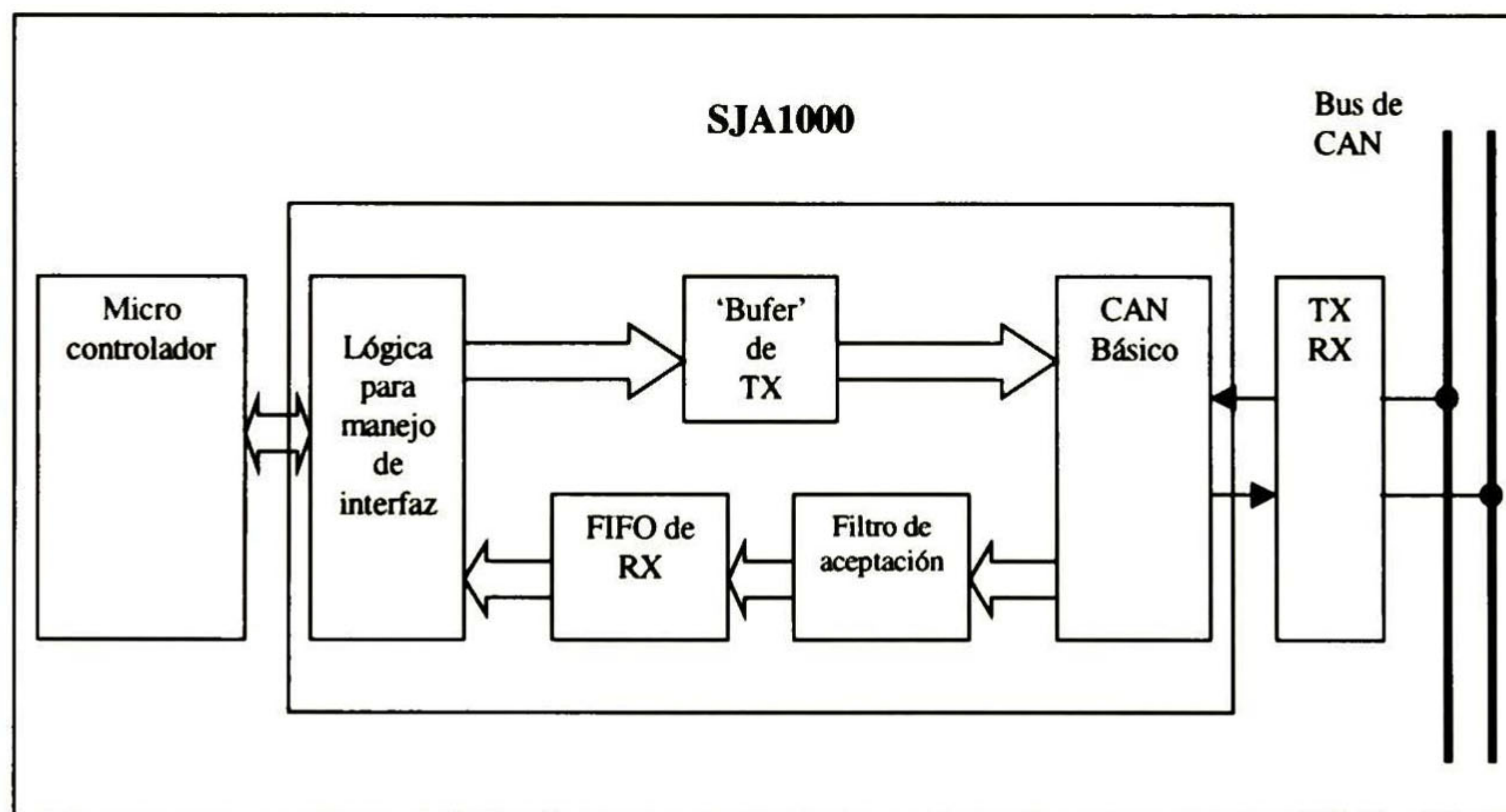


Figure 4-9 Diagrama a bloques interno del SJA1000.

El bloque CAN Básico controla la transmisión y la recepción de los mensajes de acuerdo con la especificación de CAN.

El bloque que contiene la lógica para manejo de interfaz implementa su interacción con las señales lógicas de un microcontrolador. Cada acceso a un registro del SJA1000, a través de un bus multiplexado de direcciones y datos es controlado por las señales de lectura y escritura (RD/WR) y manejado por este elemento. La mayor parte de las funciones adicionales del modo PeliCAN están implementadas en este bloque (creación de un numero mayor de registros y lógica interna).

El 'buffer' de transmisión del SJA1000 es capaz de almacenar un mensaje completo ya sea en formato estándar o formato extendido. Una vez colocado en este 'buffer', el mensaje será transmitido hacia el bus cuando el bloque de CAN básico lo requiera.

Cuando se recibe un mensaje, este bloque convierte los bits que arriban al sistema en forma serial, en un dato paralelo que es alimentado enseguida al filtro de aceptación. Con este filtro programable, el SJA1000 decide que mensajes van a ser recibidos y cuales son rechazados.

Todos los mensajes aceptados por el filtro de aceptación son almacenados en una cola tipo FIFO de recepción. Dependiendo del modo de operación y la longitud de los datos de cada mensaje, hasta un total de 32 mensajes pueden ser almacenados. Esto le permite al usuario, crear rutinas de servicio a interrupción mucho más flexibles con respecto a este controlador, ya que la probabilidad de que la cola FIFO se llene es muy baja.

Dadas las características de señalización y la diferencia de concepción de interfaz entre el SJA1000 y el H8S/2357 (principalmente por la multiplexación del bus de 8 bits en el controlador de CAN), es necesario agregar lógica entre estos dos elementos para permitir que se establezca entre ellos una correcta comunicación. La Figura 4-10 presenta el diagrama a bloques que establece la comunicación del H8S/2357 basado en el protocolo CAN 2.0B.

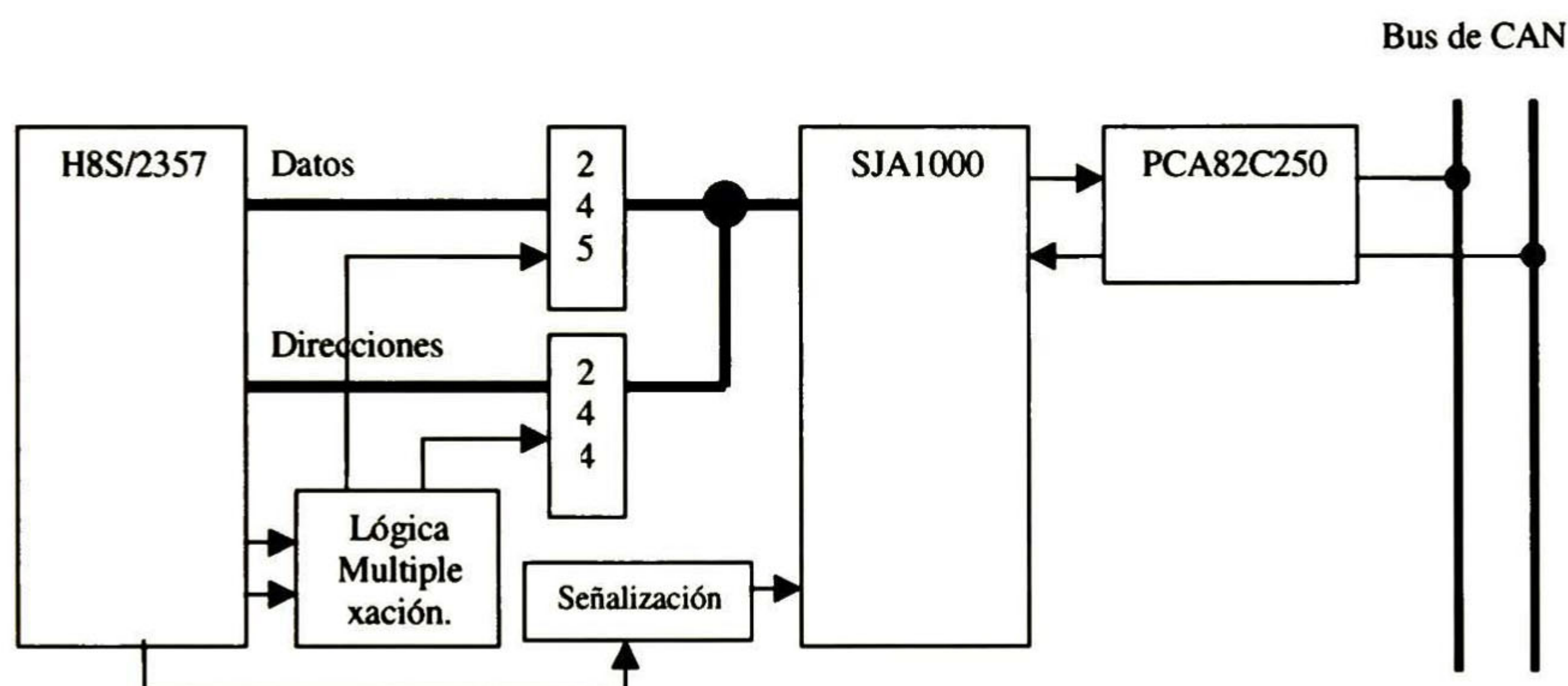


Figura 4-10 Implementación de CAN 2.0B.

El intercambio de datos entre el microprocesador y el SJA1000 se realiza por medio de registros (segmento de control) y la memoria RAM ('buffer' de mensajes) del controlador. Los registros y las localidades donde se encuentran almacenados los mensajes de entrada y salida del SJA1000 aparecen al micro como parte de su memoria externa.

Cada uno de los datos que serán transmitidos sobre el bus de CAN es cargado primeramente en el área de memoria del SJA1000, conocido como 'buffer' de transmisión. Los datos recibidos del bus son almacenados en el área de memoria del SJA1000 conocido como 'buffer' de recepción. Estos 'buffers' contienen 2, 3 o 5 bytes para el identificador y para la información de su trama (dependiendo del modo de operación y el tipo de mensaje) y hasta 8 bytes de datos.

- En el modo BasicCAN, cada mensaje requiere 10 bytes:
 - 2 bytes para el identificador.
 - 8 bytes de datos.

- En el modo PeliCAN, cada mensaje ocupa 13 bytes:
 - 1 byte para la información del mensaje.
 - 2 ó 4 bytes para el identificador (mensaje estándar o mensaje extendido).
 - 8 bytes de datos.

Este controlador cuenta también, con un versátil filtro de aceptación que le permite el detectar automáticamente los bytes del identificador y de los datos. Utilizando métodos efectivos de filtrado, mensajes o grupos de mensajes no válidos para ciertos nodos del sistema pueden ser ignorados y no almacenados en el 'buffer' de recepción.

El filtro es controlado por un código de aceptación y registros de enmascarado; el dato recibido es comparado bit por bit con los valores contenidos en los registros de códigos de aceptación, mientras que el registro de enmascaramiento define las posiciones de los bits que son relevantes para la comparación (0 = relevante, 1 = no relevante). Para aceptar un mensaje todos los bits recibidos relevantes deben de ser iguales a sus respectivos bits en el registro de código de aceptación [11].

Los métodos necesarios que debe implementar el control del hardware del sistema para establecer la comunicación CAN 2.0B por medio del SJA1000 son:

- Establecer la conexión física y lógica entre el microprocesador H8S/2357 y el controlador.
- Inicializar y configurar el SJA1000 en base a su modo de operación, manejo de errores, filtros de aceptación de mensajes.
- Transmitir mensajes.
- Recibir mensajes.
- Manejar los errores que ocurren durante la comunicación.

4.1.3 Comunicación serial

Los canales 0 (TX0, RX0) y 1 (TX1, RX1) de comunicación serial del microprocesador son utilizados para implementar su interfaz con una PC, por medio del protocolo de comunicación RS232C. A través de dos elementos de configuración, el usuario es capaz de seleccionar uno de los dos canales del microprocesador. La tarea del circuito integrado LT1180, es acoplar, en ambos sentidos, el nivel lógico de las señales del procesador, a los niveles analógicos de las señales físicas establecidas por el protocolo RS232C. La Figura 4-11 muestra la implementación de la comunicación serial.

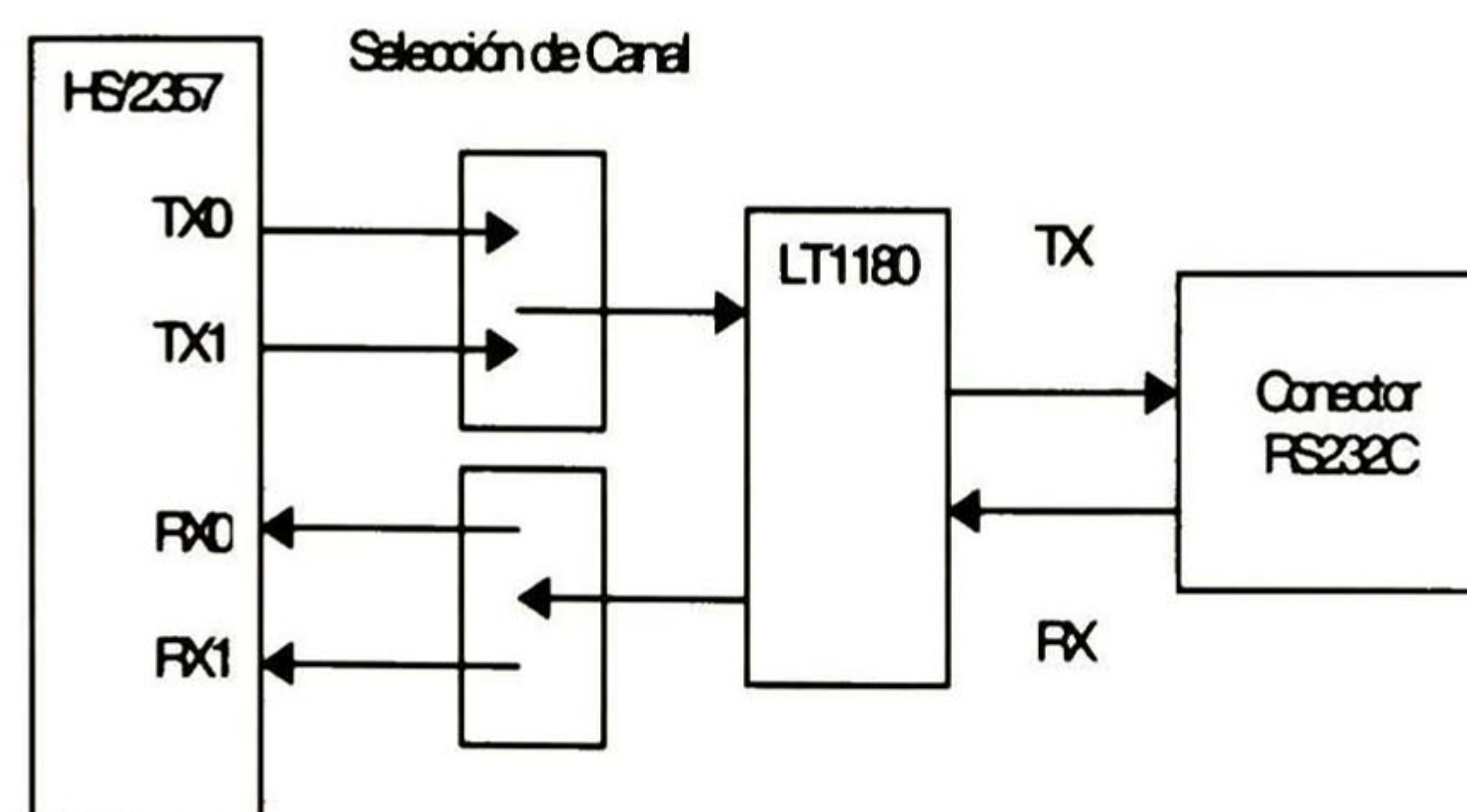


Figura 4-11 Comunicación serial del H8S/2357.

4.1.4 Conexión con el hardware de aplicación

Un conector paralelo de 40 pines se encarga de unir de manera física y lógica el hardware del sistema con el hardware de aplicación. Provee las señales requeridas por la aplicación para trabajar de acuerdo a su especificación. Las señales que pasan a través del conector son:

- a) El bus de datos del hardware de aplicación (DS0-DS7).
- b) El bus de direcciones del hardware de aplicación (AS0-AS4).
- c) Señales para el control de acceso a memoria externa (HWR, RD, CS3 y CS4).
- d) Señales de control para la electrónica de sensores (PLD_RESET, INT_EN, PLD_INT).
- e) Señales de control para manejadores de motor (MOT_ENABLE, DA0, DA1, PWM1 y PWM2).
- f) Fuente de alimentación (+5V, +26V, tierra digital, tierra analógica).

4.1.5 Fuente de alimentación

La energía necesaria para que el hardware de la plataforma de desarrollo funcione, es proporcionada por una fuente de alimentación. Esta fuente provee los niveles de voltaje y corriente necesarios para satisfacer los requerimientos de la lógica digital y analógica del sistema.

Mediante una entrada de 100-240 volts de corriente alterna, la fuente de poder es capaz de proveer al sistema de los siguientes niveles de voltaje de corriente directa:

- + 5V (hasta 1.5 amp).
- + 26V (hasta 5.0 amp).
- Tierra (digital, analógica).

El microprocesador y sus circuitos periféricos trabajan basándose en la señal de +5V y la señal de referencia digital. La línea de +26 V alimenta a los manejadores de motor que proporcionan la energía necesaria para mover motores de corriente directa y de pasos.

A fin de evitar interferencias entre las señales de alimentación, las tarjetas de circuitos impresos se dividen en dos planos: digital (+5V, señal de referencia digital) y analógico (+26V, señal de referencia analógica). Mediante este método se asegura que los niveles eléctricos de las señales se mantengan estables y que los elementos de alto consumo de energía se mantengan aislados del resto de los componentes.

Además, la separación de planos reduce considerablemente los niveles de radiación de las tarjetas, ya que agrupa en una región pequeña las señales de alta frecuencia del plano digital (señales de control principalmente) que emiten una mayor cantidad de ruido.

4.2 Hardware de aplicación

El hardware de aplicación, que se une física y lógicamente al hardware del sistema, a través de un conector, es un tipo de tarjeta universal que contiene los elementos necesarios para manejar motores de corriente directa y de pasos. Además tiene la capacidad para leer sensores y generar una interrupción cuando alguno de ellos cambie de activo a inactivo o viceversa.

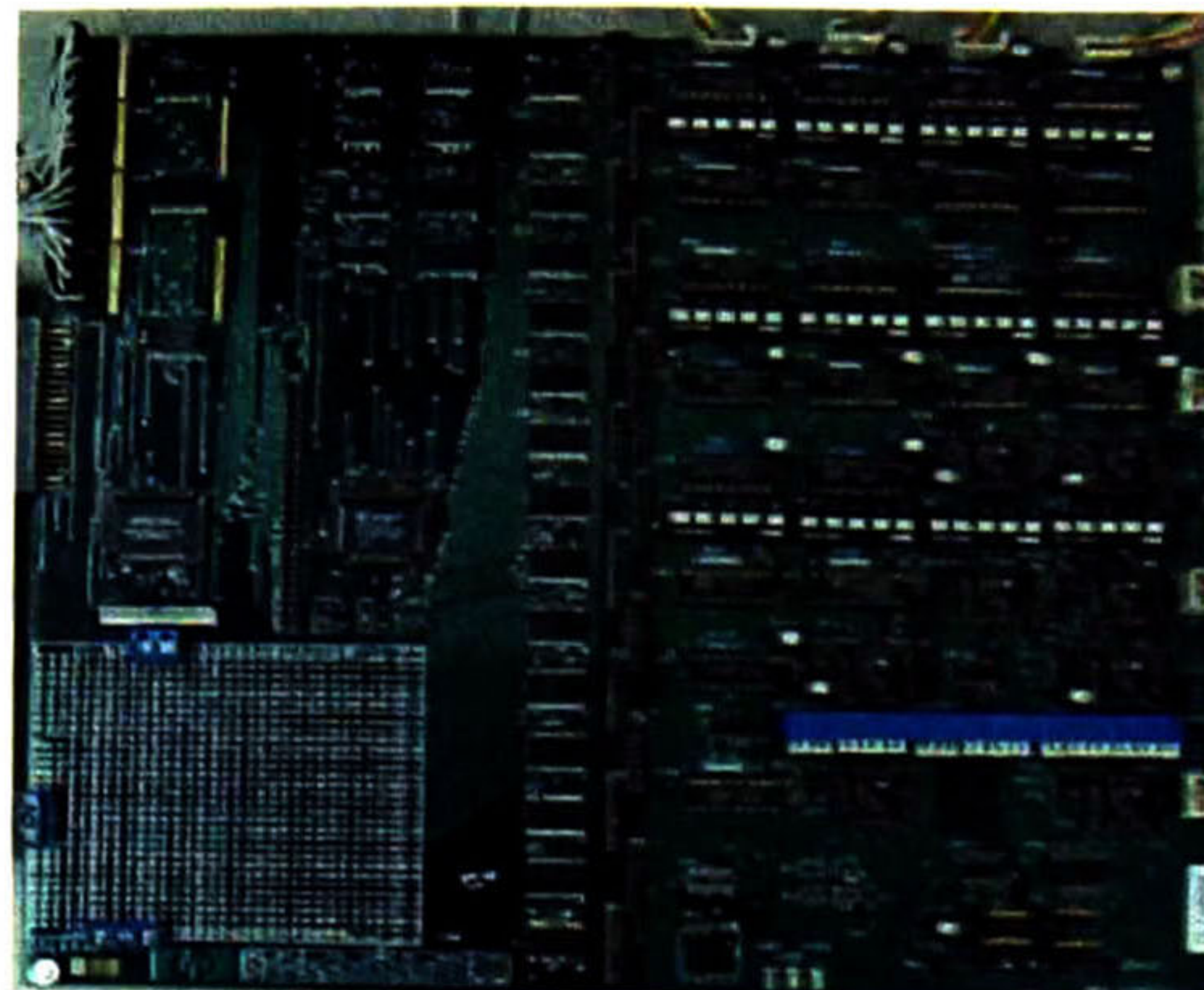


Figura 4-12 Hardware de aplicación.

Provee a la herramienta de la siguiente funcionalidad:

- Electrónica requerida para la lectura de 16 sensores.
- Lógica de interrupción de sensores, basada en un dispositivo lógico programable.
- Manejadores de propósito general configurables, con capacidad para controlar 14 motores de pasos y 31 motores de corriente directa.

El diagrama a bloques de la Figura 4-13 describe la arquitectura del sistema. Muestra cada uno de los bloques funcionales que lo componen y la forma en que interactúan entre ellos.

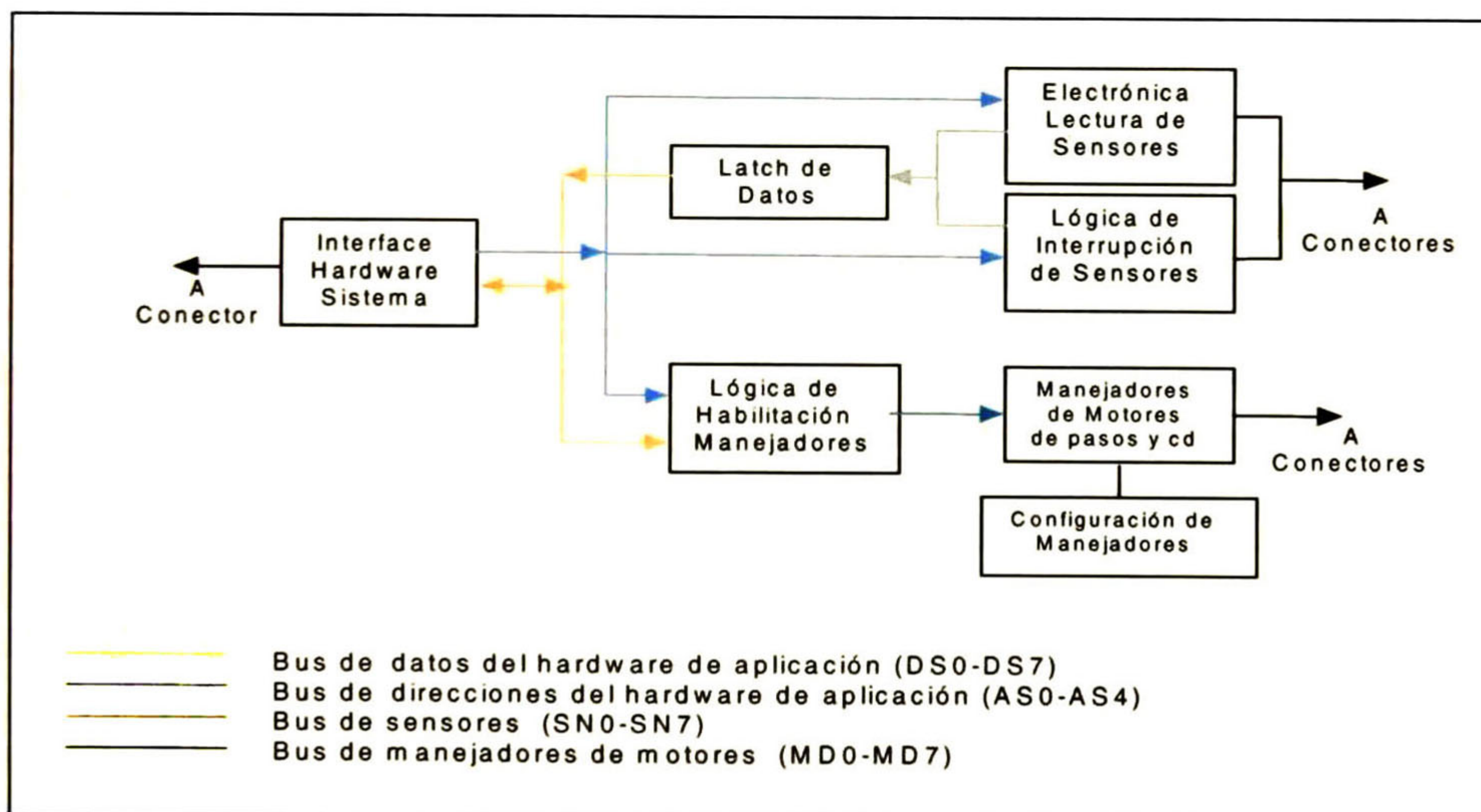


Figura 4-13 Arquitectura del hardware de aplicación.

La descripción de cada bloque se basa en el desarrollo de los siguientes puntos:

1. Mapeo de memoria externa.
2. Electrónica para lectura de sensores.
3. Lógica de interrupción de sensores.
4. Manejadores de motor.

4.2.1 Mapeo de memoria externa

La Tabla 4-3 que se muestra a continuación, describe la localización de los componentes del hardware de aplicación, que forman parte de la memoria externa del microprocesador.

Señal	Rango de direcciones	Elemento externo
CS3	0x 600 000-0x 7FF FFF	Sensores
	0x 600 000	Primer grupo de 8 sensores
	0x 600 001	Segundo grupo de 8 sensores
	0x 600 002	PLD generador de interrupción
CS4	0x 800 000-0x 9FF FFF	Manejadores de motor
	0x 800 000	Manejadores 1 y 2 (de propósito general, 1.2 Amp)
	0x 800 001	Manejadores 3 y 4 (de propósito general, 1.2. Amp)
	0x 800 002	Manejadores 5 y 6 (de propósito general, 1.2. Amp)
	0x 800 003	Manejadores 7 y 8 (de propósito general, 1.2. Amp)
	0x 800 004	Manejadores 9 y 10 (de propósito general, controlado por un convertidor digital-analógico interno)
	0x 800 005	Manejadores 11 y 12 (de propósito general, 1.2. Amp)
	0x 800 006	Manejadores 13 y 14 (de propósito general, 1.2. Amp)
	0x 800 007	Manejadores 15 y 16 (de propósito general, 1.2. Amp)
	0x 800 008	Manejadores 17 y 18 (de propósito general, 1.2. Amp)
	0x 800 009	Manejadores 19, 20 (de propósito general) y 21 (modo PWM)
	0x 800 00A	Manejadores 23, 24 (de propósito general) y 22 (modo PWM)
	0x 800 00B	Manejadores 25 y 26 (de propósito general, 1.8. Amp), LED 1
	0x 800 00C	Manejadores 27 y 28 (de propósito general, 1.8. Amp), LED 3
	0x 800 00D	Manejadores 29 y 30 (de propósito general, controlado por un convertidor digital-analógico externo), LED2
	0x 800 00E	Manejador 31 (5 Amp)
	0x 800 00F	Area de Wire Wrap
	0x 800 010	Convertidor digital-analógico externo, manejadores 29 y 30.

Tabla 4-3 Mapeo en memoria externa de los componentes del hardware de aplicación.

4.2.2 Electrónica para lectura de sensores

Se compone de dos circuitos integrados de 8 bits (74LS540) que detectan, acoplan y direccionan el estado lógico de los sensores, a la memoria externa del microprocesador. Las siguientes tablas muestran los dos grupos de 8 sensores, su distribución como parte de la memoria externa del procesador y el conector físico que le corresponde.

Bit	Conector
0	J9
1	J1
2	J13
3	J7
4	J12
5	J8
6	J14
7	J10

Tabla 4-4 Grupo 1 de sensores de 8 bits (dirección 0x 600 000).

Bit	Conector
0	J23
1	J17
2	J28
3	J16
4	J27
5	J18
6	J19
7	J24

Tabla 4-5 Grupo 2 de sensores de 8 bits (dirección 0x 600 001).

Cada conector tiene como entrada la señal de un sensor que pasa a través de un filtro pasa bajos encargado de eliminar el ruido en la tarjeta; y como salida, los niveles de voltaje Vcc (+5V) y tierra (GND), necesarios para la correcta operación de los sensores.

4.2.3 Lógica de interrupción de sensores

Por medio del dispositivo lógico programable (PLD) EPM7064S de Altera, el hardware de aplicación tiene la capacidad para generar una interrupción cuando el nivel lógico de uno de los sensores cambia de activo a inactivo o viceversa [12].

Basado en el análisis de las señales que el PLD requiere para su operación, y que se muestran en la Figura 4-14, a continuación se describe su funcionalidad.

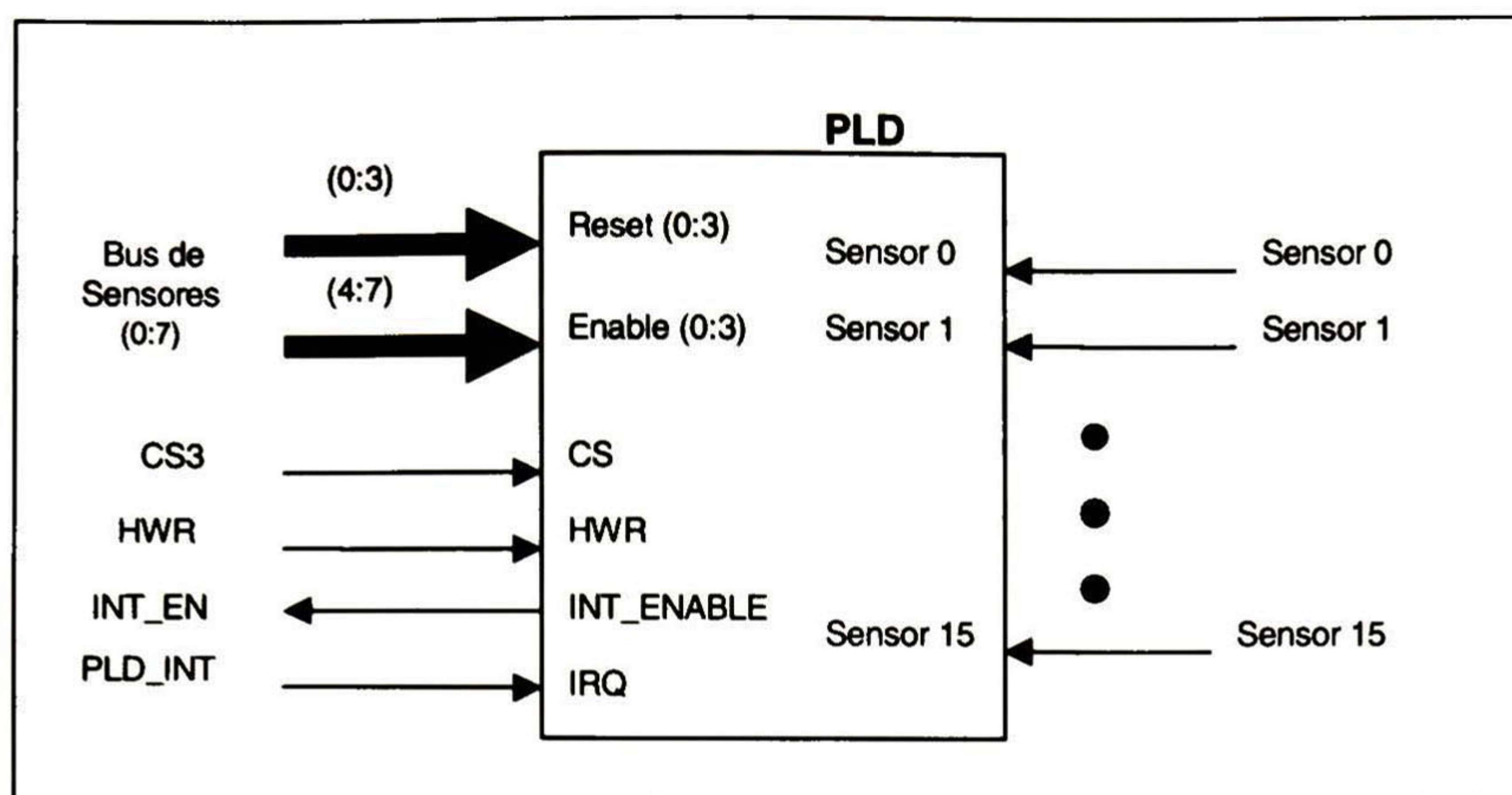


Figura 4-14 Dispositivo lógico programable (PLD) generador de interrupción.

Sensores (0:15)

Las señales físicas provenientes de los sensores son conectadas directamente a la tarjeta y al PLD donde son filtradas y acopladas, para su constante monitoreo.

Señal de Reset (0:3)

Parte baja del bus de sensores descrito en la arquitectura del hardware de aplicación; su función es reiniciar el estado de un sensor a su nivel actual. Requiere que las señales de CS y HWR se encuentren en un '0' lógico para que el nivel del sensor indicado por un número codificado en 4 bits sea actualizado.

Señal de Enable (0:3)

Parte alta del bus de sensores; su función es habilitar o deshabilitar sensores. Sólo requiere que las señales de CS y HWR se encuentren en un '0' lógico para habilitar o deshabilitar el sensor indicado por un número codificado en 4 bits.

CS3

Señal de control que habilita el PLD para los ciclos de escritura que reinician o habilitan algún sensor.

HWR

Señal de control que permite un ciclo de escritura sobre el PLD.

INT_EN

Señal que habilita la capacidad del PLD para interrumpir al microprocesador cuando un sensor ha cambiado de estado lógico.

IRQ

Señal que interrumpe al microprocesador, si y sólo si la señal INT_ENABLE se encuentre en '1' lógico, y cualquiera de los sensores conectados y previamente habilitados cambia de activo a inactivo o viceversa. La interrupción se genera por la transición de '1' a '0' lógico y vuelve a su estado normal una vez que el sensor que previamente ha cambiado es reiniciado a su valor lógico actual.

4.2.4 Manejadores de motor

Los manejadores de motor se encuentran habilitados por una lógica que permite su acceso por medio de direcciones externas de la memoria del microprocesador. Este bloque es implementado por 15 circuitos integrados flip-flops tipo D de 8 bits (74LS374) que direccionan las señales necesarias para la operación y control de 14 motores de pasos y 31 motores de corriente directa. El control de los motores es implementado por dos circuitos integrados manejadores de motor con propósito general, el PBL3717_2 (1.2 Amp) y el PBL3770A (1.8 Amp), y por un manejador de corriente directa, discreto, capaz de operar a 5 Amp. Esta lógica habilita también las señales necesarias para utilizar 3 LED's de señalización.

El hardware de aplicación cuenta con elementos de configuración (interruptores), que permiten el uso de los manejadores de propósito general, para mover motores, determinar su nivel de consumo de corriente máximo, y definir el tipo de control de operación de cada motor.

Así, la configuración de la tarjeta para mover los motores depende de las señales de los manejadores y la posición de los interruptores. En general, existen 6 configuraciones diferentes para manejar motores de pasos y motores de corriente directa. A continuación se explica cada uno de ellos.

4.2.4.1 Bloque de propósito general

A partir del par de manejadores 1 y 2 y hasta el par 19 y 20 se consideran como bloques de propósito general con la capacidad para operar motores de pasos o motores de corriente directa a un nivel máximo de corriente de 1.2 Amp. Se compone de manejadores del tipo PBL3717_2 y su configuración se muestra en la Figura 4-15.

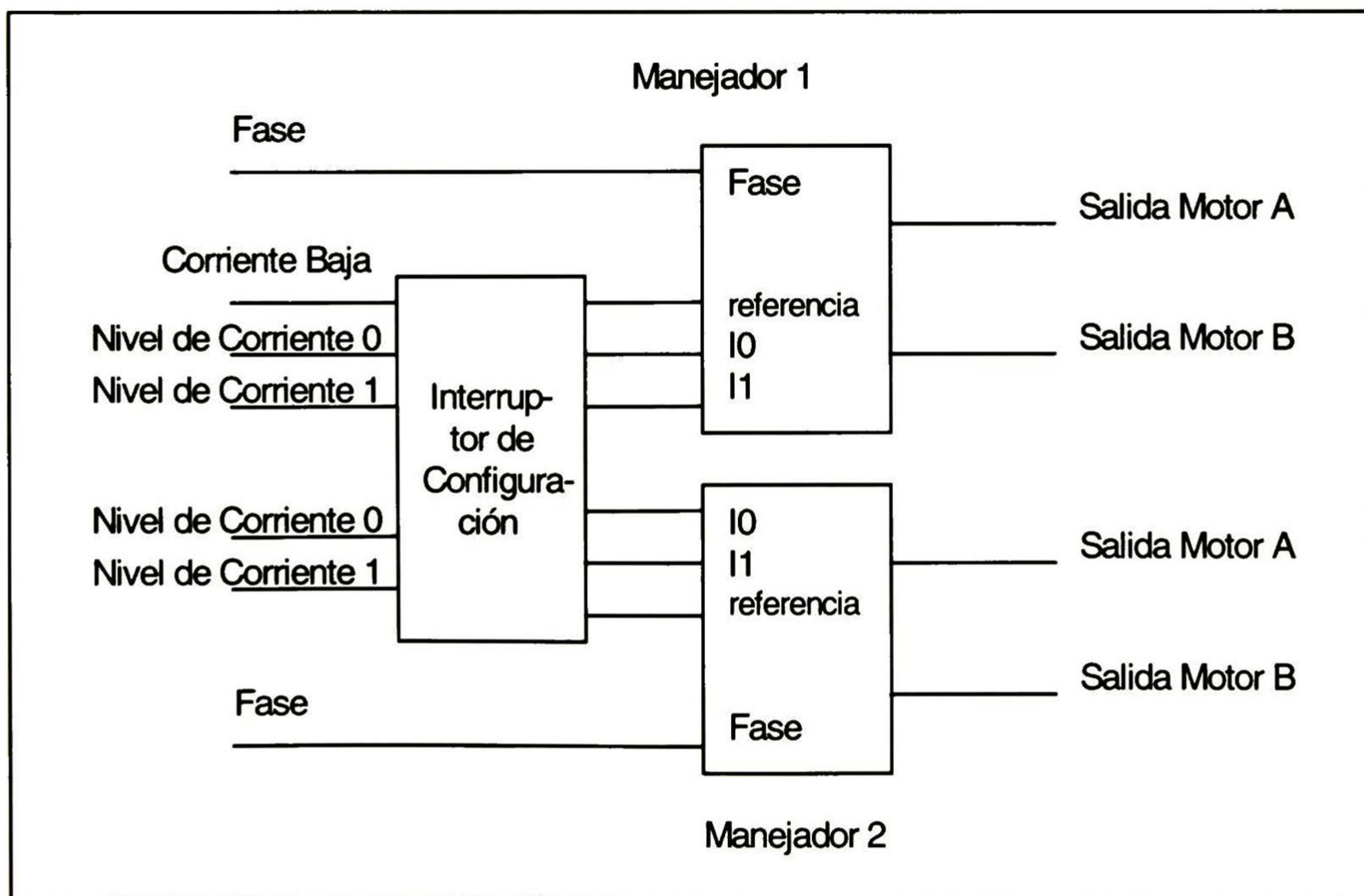


Figura 4-15 Par de manejadores de motor en configuración bloque de propósito general.

La señal de referencia fija un voltaje en el manejador proporcional a la máxima corriente permitida en la bobina del motor. Las señales I0 y I1 permiten que la corriente final que circule en la bobina sea un porcentaje de la referencia total. La Tabla 4-6 muestra la combinación lógica de estas señales.

I1	I0	Resultado
0	0	100 % de la referencia
0	1	60 % de la referencia
1	0	20 % de la referencia
1	1	0 % de la referencia

Tabla 4-6 Niveles de corriente en un manejador de motor.

El interruptor permite configurar el sistema para el manejo y control de motores, en base a 5 posiciones diferentes y puede ser utilizado para mover un motor de pasos mediante la técnica de pasos completos o sextos de pasos. Permite además, el uso de un nivel de corriente fijo o selección de alta y baja corriente para el motor, alternando la selección de la referencia. Tabla 4-7 y Tabla 4-8, describen las posibles combinaciones del interruptor con 5 posiciones; es importante señalar que no todas las opciones son válidas, por lo que se deben considerar cuidadosamente al momento de configurarlos. La señal "Corriente Baja" atenúa la referencia con un divisor de voltaje; cuando su valor lógico es '1', el voltaje cae a la mitad de su valor máximo, mientras que si su valor es '0', el máximo nivel de corriente es seleccionado.

Posición 1	Posición 2	Nivel de Corriente.
Off	Off	En los manejadores 1 y 2, el nivel de la corriente es controlado por la señal Corriente Baja.
Off	On	El nivel de corriente bajo y alto del manejador 1 es controlado por Corriente Baja. El nivel de corriente del manejador 2 es siempre alto.
On	Off	El nivel de corriente de los manejador 1 y 2 es siempre alto. La señal de Corriente Baja no se utiliza para estos casos.
On	On	

Tabla 4-7 Posiciones 1 y 2 de un interruptor de configuración para los manejadores.

4.2.4.2 *Bloque de propósito general con capacidad para manejo de alta corriente*

El par de manejadores 23 y 24 y hasta el par 27 y 28 es considerado como un bloque de propósito general con la capacidad para operar motores de pasos o motores de corriente directa, pero a diferencia del bloque anterior, pueden proporcionar hasta un máximo de corriente de 1.8 Amp. Este bloque se compone de manejadores del tipo PBL3770A y su comportamiento es similar al descrito en el punto anterior.

Posición 3	Posición 4	Posición 5	Señales de control de los manejadores: Nivel de Corriente 0 (I0) y Nivel de Corriente 1 (I1).
Off	Off	Off	La señal I0 del manejador 2, controla el nivel de corriente de los dos manejadores. Así se puede controlar un motor de pasos con pasos completos o dos motores de corriente directa que se muevan al mismo tiempo.
Off	Off	On	La I0 del manejador 1 controla el nivel de corriente del manejador 1, mientras que el nivel de corriente del manejador 2 es controlado por la señal I0 del manejador 2. Esta combinación es útil para el control independiente de 2 motores de corriente directa. No se recomienda para motores de pasos.
Off	On	Off	Esta combinación no se recomienda.
Off	On	On	La señal I0 del manejador 1 controla el nivel de corriente del Manejador 1, mientras que el nivel de corriente del Manejador 2 es controlado por sus señales I0 e I1. Esta combinación es útil para el control de 2 motores de corriente directa, pero no para un motor de pasos. Para el manejador 1 la señal I0 controla que haya o no corriente en la bobina. Para el manejador 2, se tiene la capacidad de manejar 0, 20, 60 y 100% de la corriente total.
On	Off	Off	Esta combinación no se recomienda.
On	Off	On	La señal I0 del manejador 2 controla, el nivel de corriente del manejador 2, mientras que el nivel de corriente del manejador 1 es controlado por sus señales I0 e I1. Esta combinación es útil para el control de 2 motores de corriente directa, pero no para un motor de pasos. Para el manejador 2 la señal I0 controla que haya o no corriente en la bobina. Para el manejador 1, se tiene la capacidad de manejar 0, 20, 60 y 100% de la corriente total.
On	On	Off	Esta combinación no se recomienda.
On	On	On	Los manejadores 1 y 2 son controlados por sus respectivas señales I0 e I1. Esta configuración es útil para el control de motores de pasos utilizando la técnica de sextos de pasos. Puede ser utilizado, también para mover 2 motores de corriente directa, con la posibilidad de seleccionar el nivel de corriente entre el 0, 20, 60 y 100%.

Tabla 4-8 Posiciones 3, 4 y 5 de un interruptor de configuración para los manejadores.

4.2.4.3 Manejadores para motores de corriente directa controlados por una señal PWM

La conexión de un manejador de motor de corriente directa controlado por una señal PWM, se muestra en la siguiente figura.

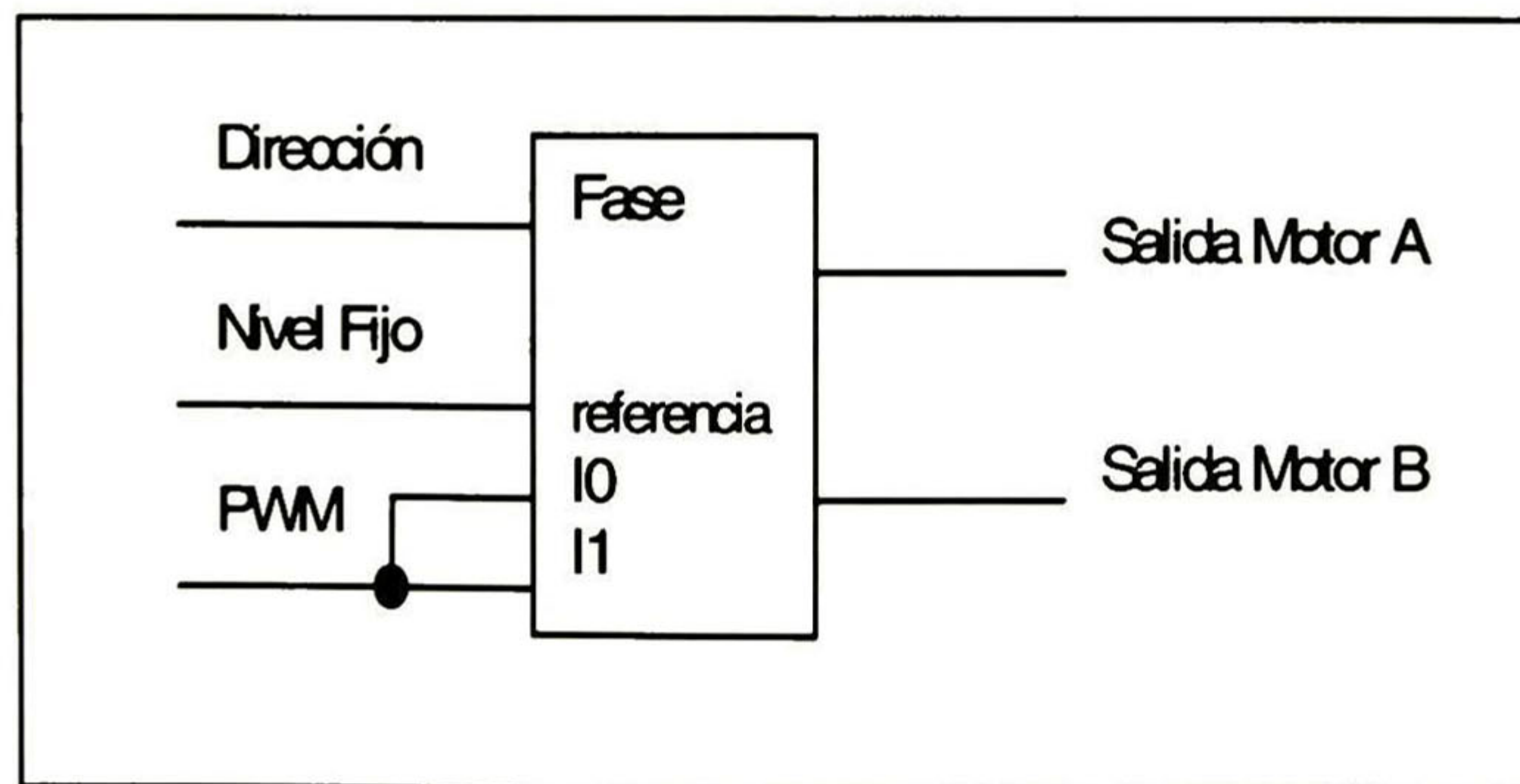


Figura 4-16 Configuración de un manejador de motor de corriente directa controlado por una señal PWM.

Los manejadores 21 y 22 son del tipo PBL3717, en estos manejadores, las señales I0 e I1 se encuentran unidas, de manera que sólo es posible aplicar el 0 ó 100% de la corriente máxima sobre las bobinas del motor. Una señal modulada por ancho de pulso (PWM) generada por un módulo interno del microprocesador, es utilizada con este propósito.

4.2.4.4 *Manejador discreto para un motor de corriente directa*

Existe un manejador implementado con transistores MOSFET discretos, que puede proporcionar hasta 5 amperes de corriente, este manejador es utilizado para mover motores de corriente directa con un alta demanda de energía (por ejemplo, el motor de una engrapadora).

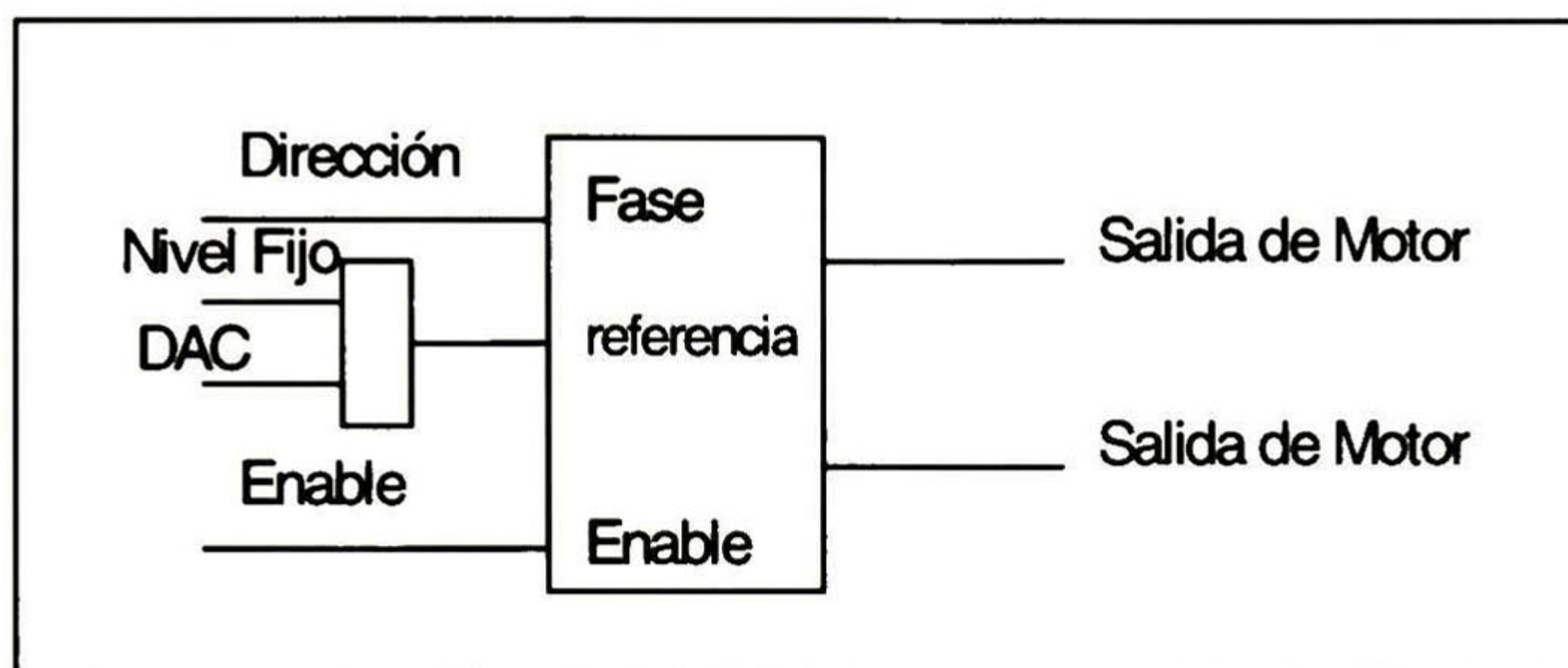


Figure 4-17 Configuración de un manejador discreto para un motor de corriente directa.

4.2.4.5 *Bloque controlado por un convertidor digital-analógico externo*

Este bloque es una combinación de los casos presentados anteriormente que se compone por los manejadores de propósito general 29 y 30. Su parte central funciona exactamente igual que el bloque de propósito general con capacidad para manejo de alta corriente (1.8 Amp). Adicionalmente, cuenta con la capacidad para cambiar la referencia de voltaje, por medio de un convertidor digital-analógico integrado (PBM3962), este circuito puede alimentar dinámicamente la referencia de los manejadores de acuerdo a su necesidad de operación. La Figura 4-18 muestra su configuración.

El interruptor adicional de 6 posiciones es un multiplexor, donde cada una de sus posiciones deben estar en "On" para que los manejadores funcionen como se describe en el caso 2 o todos en estado "Off" para alimentar al bloque con las señales de un convertidor digital-analógico (DA1, DA2, PH1, PH2 y ENABLE). Es importante configurar muy bien el interruptor, de acuerdo a lo antes descrito, ya que un error puede provocar resultados no esperados.

4.2.4.6 *Bloque controlado por un convertidor digital-analógico interno*

La última configuración para el manejo de motores, presenta el mismo comportamiento que el descrito en el caso anterior, la única diferencia es la manera en que está implementado. Los manejadores 9 y 10 son utilizados para este tipo de control, sólo que en este caso, la referencia dinámica de voltaje es generada directamente por un convertidor interno digital-analógico del microprocesador, que sustituyen a la señal "Corriente Baja" de ambos manejadores.

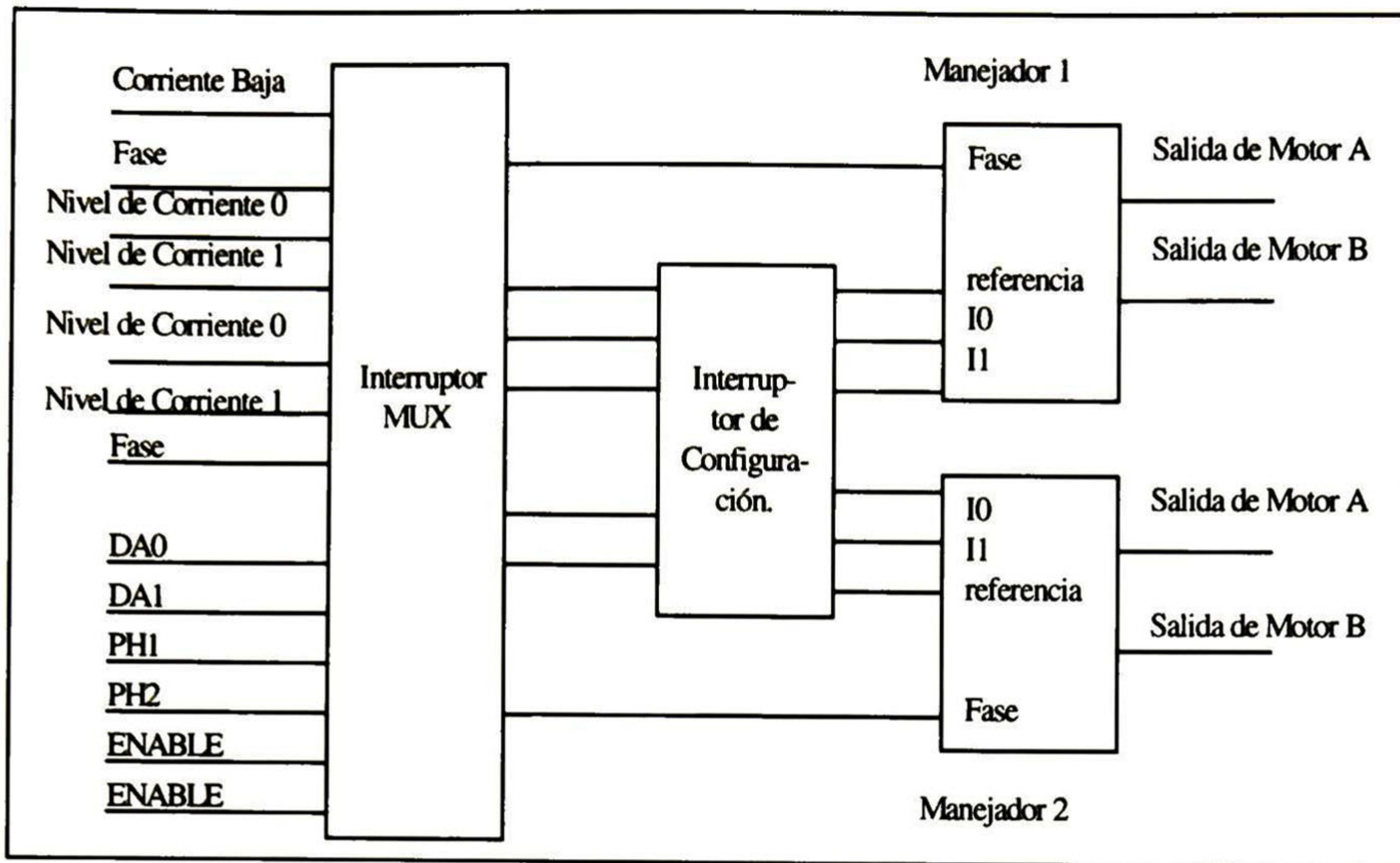


Figura 4-18 Configuración de dos manejadores controlados por un convertidor digital-analógico externo.

4.3 Diseño de PCB's

El diseño de las tarjetas de circuitos impresos (PCB's), que fue desarrollado con la herramienta de CAD Mentor Graphics 8.6_4, y que constituyen el hardware del sistema y el hardware de aplicación, se basa en la arquitectura funcional descrita en las secciones anteriores. La Figura 4-19 y la Figura 4-20 muestran la distribución física de sus componentes.

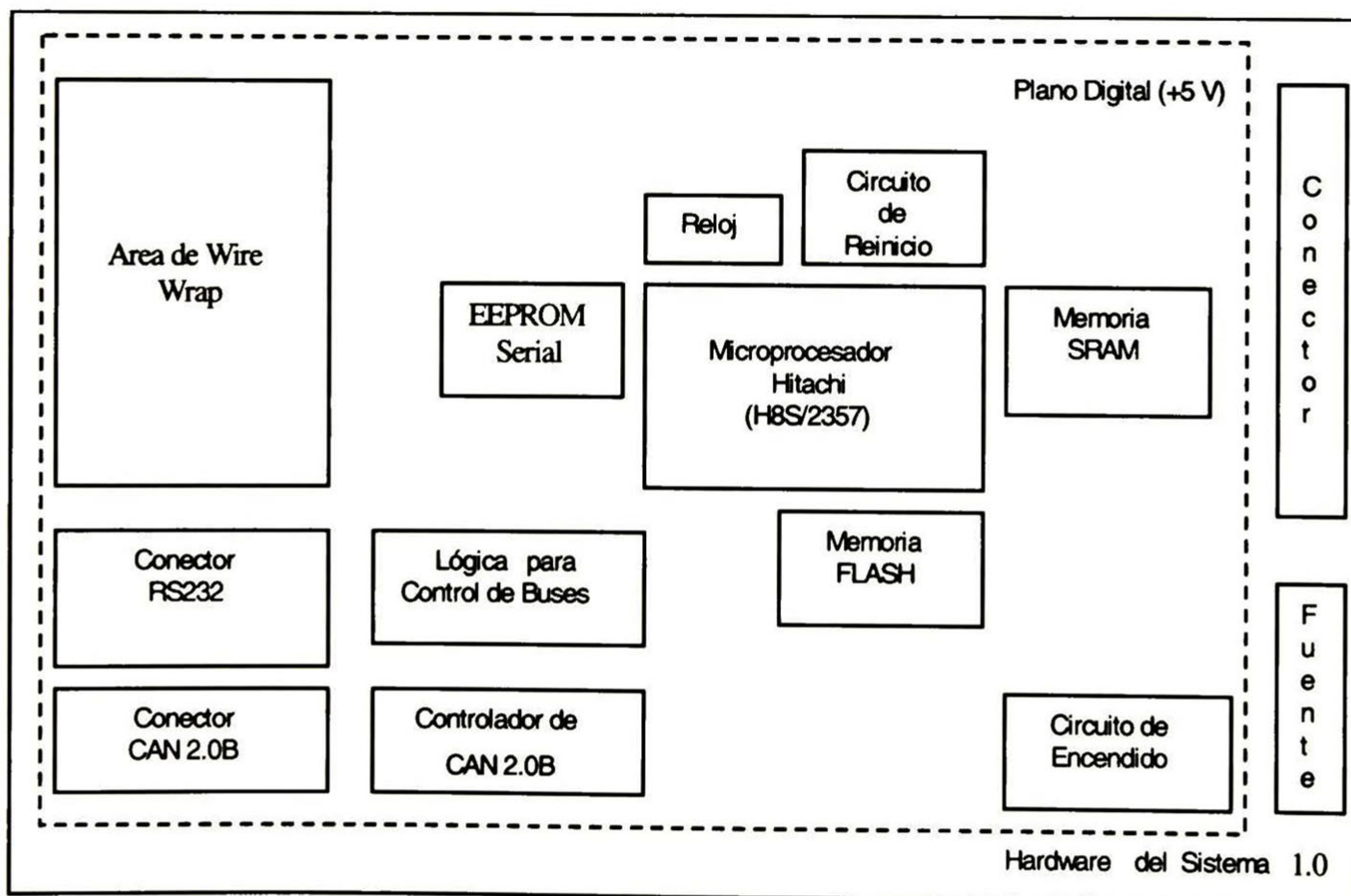


Figura 4-19 Diseño de PCB de hardware del sistema.

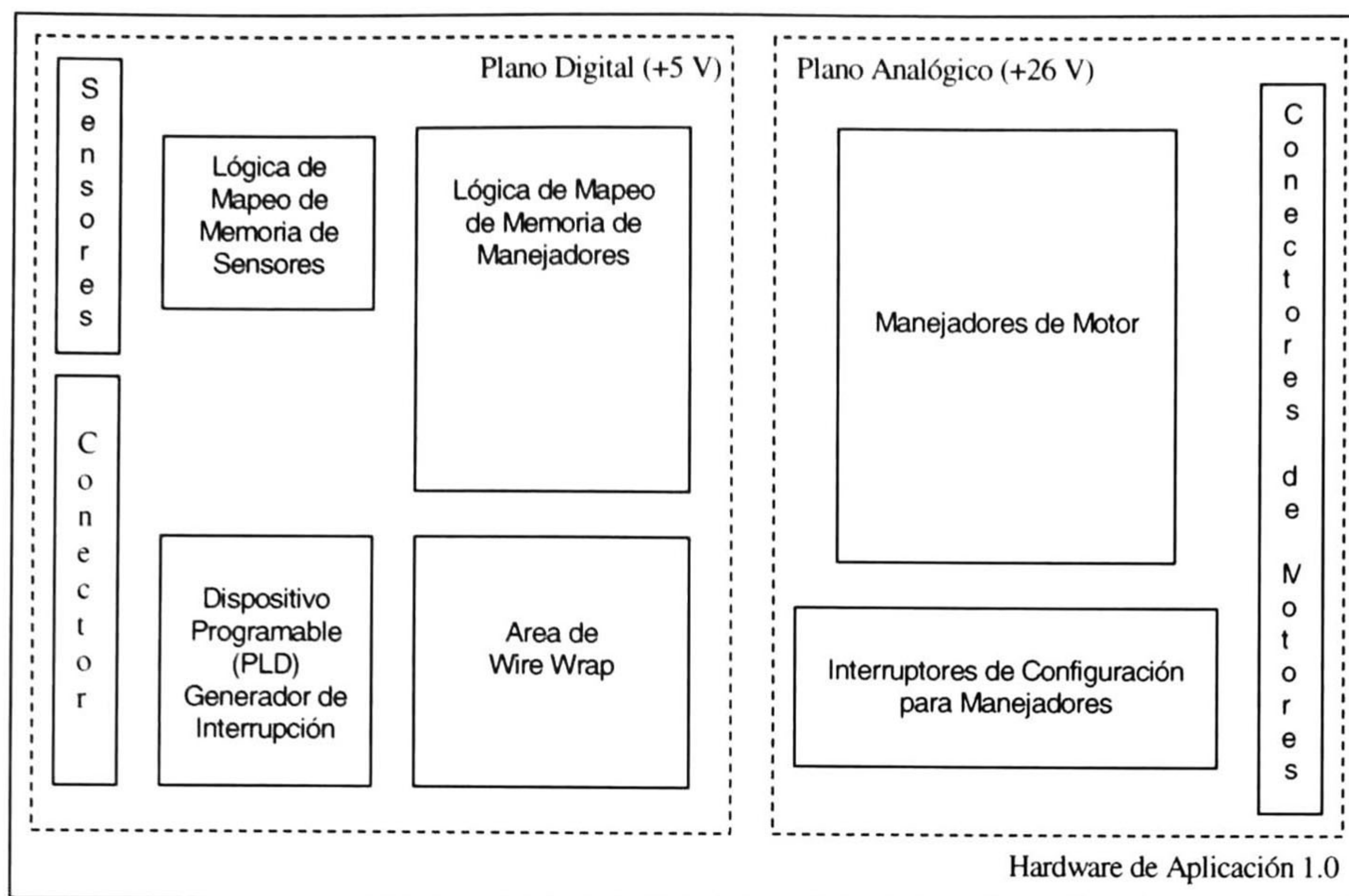


Figura 4-20 Diseño de PCB de hardware de aplicación.

La distribución de las señales de energía de las tarjetas está dividido en dos partes, el plano digital (+5V, señal de referencia digital) y el plano analógico (+26V, señal de referencia analógica). Mediante este método se asegura que los niveles eléctricos de las señales se mantengan estables y que los elementos de alto consumo de energía se mantengan aislados del resto de los componentes.

De igual forma, y a fin de evitar que las tarjetas se conviertan en fuentes importantes de radiación, su diseño agrupa en regiones pequeñas la mayor parte de las señales de alta frecuencia (señales de control principalmente) del plano digital.

El hardware del sistema y el hardware de aplicación físicamente están implementadas en tarjetas de 4 capas, hechas del material FR4, donde sus componentes electrónicos son conectados por medio de la tecnología de montaje superficial (PCA).

4.4 Pruebas de hardware

Una vez diseñado e implementado el hardware de la herramienta, el siguiente paso consiste en determinar si trabaja de acuerdo a su especificación. Los mecanismos de verificación aplicados a las tarjetas, hardware del sistema y hardware de aplicación, se dividen en pruebas eléctricas (hardware en general) y pruebas funcionales (específicas para cada elemento).

4.4.1 Pruebas eléctricas

El procedimiento necesario para cumplir con este requerimiento de hardware es:

1. Realizar una exploración visual para revisar que cada componente se encuentre en el lugar que le corresponde y libre de cualquier contacto que pueda ocasionar un corto circuito.
2. Medir la impedancia de las tarjetas, ya que representa uno de los principales parámetros para la detección de cortos y componentes dañados.

3. Energizar las tarjetas, con el fin de medir los niveles de voltaje y corriente nominales; el voltaje, para los componentes lógicos debe mantenerse sobre un rango de tolerancia de 5V +/- 5%, mientras que el voltaje requerido por los manejadores de motor no debe superar el rango de 26V +/-10%. El consumo de corriente, por otra parte, no debe ser mayor a la suma total del consumo máximo de cada uno de sus componentes y que puede ser proporcionado por la fuente.

4.4.2 Prueba funcional del hardware del sistema

Dado que su arquitectura se basa en un microprocesador, su principal función es ejecutar las tareas programables relacionadas con el control de cada uno de los elementos que conforman el sistema. Los pasos a seguir para cumplir con la prueba funcional del hardware del sistema son:

1. Determinar el funcionamiento correcto del microprocesador y la memoria FLASH, que contiene el código del programa a ejecutar; esto se logra mediante un programa que encienda y apague periódicamente un LED conectado al procesador.
2. Determinar el funcionamiento correcto del acceso a las memorias externas FLASH, SRAM y EEPROM, mediante un programa que escribe y lee una palabra conocida sobre cada una de las localidades de estos componentes. Es importante verificar la eficiencia de estos elementos ya que contienen el código y los datos utilizados por el procesador para llevar a cabo sus tareas.
3. Prueba del protocolo de comunicación RS232C, que constituye la interfaz principal entre el hardware y una PC, así como una herramienta de prueba mucho más robusta, para tareas más complejas. Dado que el controlador del puerto serial se encuentra dentro del microprocesador, sólo es necesario configurar y probar la interfaz física que acopla las señales digitales con la señal analógica del bus; un programa en el procesador, que espera por un carácter específico y que responde con un carácter asociado, cumple con el requerimiento de verificación de la interfaz.
4. Prueba de la interfaz entre el sistema y el protocolo de comunicación CAN. Para esto se requiere un código que configure el controlador para recibir y transmitir mensajes por el bus CAN, conectado a un sistema monitor que supervisa su funcionamiento.

4.4.3 Prueba funcional del hardware de aplicación

Las pruebas funcionales del hardware de aplicación se enfocan a la capacidad de la tarjeta para configurar y operar motores de pasos y motores de corriente directa, así como leer el estado de algunos sensores. A continuación se describen, basándose en cada uno de sus componentes, la estructura de las pruebas del sistema.

1. La interfaz con el hardware del sistema, a través de un conector paralelo, provee cada una de las señales requeridas por la aplicación para operar de acuerdo a su especificación, debido a esto, es necesario verificar que cada señal este presente en el conector antes de iniciar su operación, empezando por detectar las líneas de alimentación (+5V, +26V, GND).
2. El siguiente punto es la electrónica para lectura de sensores. La prueba consiste en, conociendo el estado de cada sensor, llevar a cabo ciclos de lectura sobre los dos grupos de 8 sensores mapeados en memoria, que refleje su estado actual; después, cambios aleatorios y constantes en los niveles de los sensores, representa una prueba

más confiable para este grupo de componentes, así como para el PLD encargado de la generación de una interrupción.

3. Por último es necesario probar los manejadores de motor en base a las 6 configuraciones diferentes ya descritas, dado que el mapeo de las señales requeridas para mover motores se implemente por medio de 15 flip-flops tipo D, de 8 bits, es importante verificar que cada una de las señales deseadas se reflejen sobre estos componentes y sus respectivos manejadores, escribiendo datos conocidos sobre cada uno de ellos.
4. Una vez demostrado que las señales se encuentran presentes en los manejadores, basta atender las indicaciones de operación de cada configuración para determinar su buen funcionamiento. Manejadores de motor, interruptores, señales de PWM, convertidores digital-analógico, comprueban su efectividad con el arranque y paro de motores de pasos y motores de corriente directa.

5 Software y protocolo de comunicación

El software de aplicación representa el medio para que el usuario interactúe con la plataforma de desarrollo a fin de probar el sistema electromecánico de los manejadores de papel Hewlett Packard.

A través de una interfaz gráfica interactiva que le permite configurar y manejar motores y sensores, un ingeniero es capaz de encontrar los parámetros de operación óptimos de los dispositivos. En este capítulo se describe el diseño e implementación del programa que proporciona dicha funcionalidad.

A fin de cumplir con su objetivo, la herramienta contiene la lógica necesaria para crear la interfaz física con el software de aplicación de usuario, a través del protocolo RS232C. La segunda parte de este capítulo describe la especificación e implementación de este estándar de comunicación.

5.1 Software de aplicación de usuario

El diseño del programa de aplicación de usuario, que fue diseñado para formar parte de la herramienta para el desarrollo del esquema de control de los dispositivos manejadores de papel, puede ser descrito de acuerdo a su estructura interna y su relación con el firmware y hardware. Su objetivo es proporcionar al usuario una interfaz amigable y clara con la plataforma física de la herramienta, para mover motores de corriente directa, motores de pasos y conocer el estado de sensores.

El software, desarrollado en objetos, empleando como herramienta C++ Builder 3.0, reside en una PC, y se ejecuta sobre el sistema operativo Windows 95/98/NT.

La información obtenida a través de su interfaz gráfica es codificada y transmitida hacia el firmware de control para que, por medio del hardware del sistema, implemente la tarea deseada.

El diagrama a bloques de la Figura 5-1, muestra la relación del software con el firmware de control y el hardware del sistema. El programa de aplicación, representa el medio para que el usuario interactúe y exponga al sistema de desarrollo su requerimiento de operación.

Una vez obtenida la información, el programa debe interpretarla, codificarla y transmitirla, en un formato conocido, hacia el firmware de control encargado de actuar directamente sobre el hardware del sistema. De igual forma, debe recibir, decodificar e interpretar la información proveniente de este módulo para mostrarla al usuario.

Dada la estructura y operación de esta herramienta, el diseño del módulo de software y su relación con los módulos de firmware y hardware que complementan la herramienta, se describe de acuerdo a los siguientes puntos:

1. Arquitectura del software.
2. Interfaz entre el programa y el usuario.
3. Comunicación entre el software de aplicación y el firmware de control.
4. Pruebas del software.

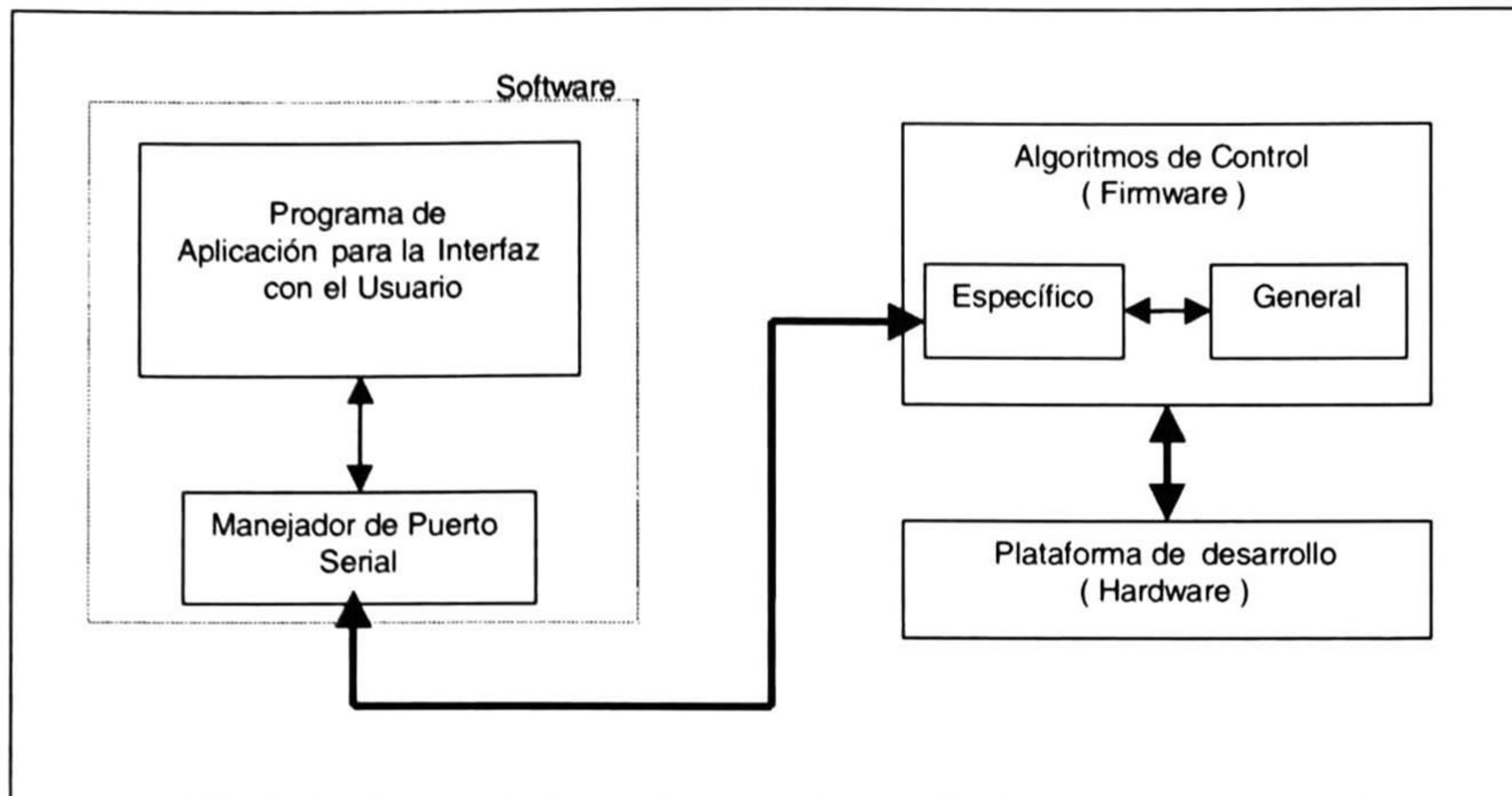


Figura 5-1 Relación entre el software de aplicación y los módulos hardware y firmware.

5.1.1 Arquitectura del software

La arquitectura del software de aplicación define la concepción general de control y operación de un programa, es decir, la manera en que se estructura y organiza, de acuerdo a sus especificaciones, para ser implementado [13].

Generalmente, la plataforma de desarrollo de software determina los recursos y el tipo de programación que se utiliza. En este caso, la herramienta C++ Builder 3.0 permite el manejo de la programación orientada a objetos que satisface la funcionalidad deseada del sistema.

El software de aplicación, representa el medio para que el usuario interactúe con la plataforma de desarrollo a fin de probar el sistema electromecánico del manejador de papel. A través de una interfaz gráfica interactiva que le permite configurar y manejar motores y sensores, un ingeniero es capaz de encontrar los parámetros de operación óptimos para que los dispositivos trabajen de la mejor manera.

La implementación del software, desarrollado en objetos, está basada en el diagrama a bloques de la Figura 5-2.

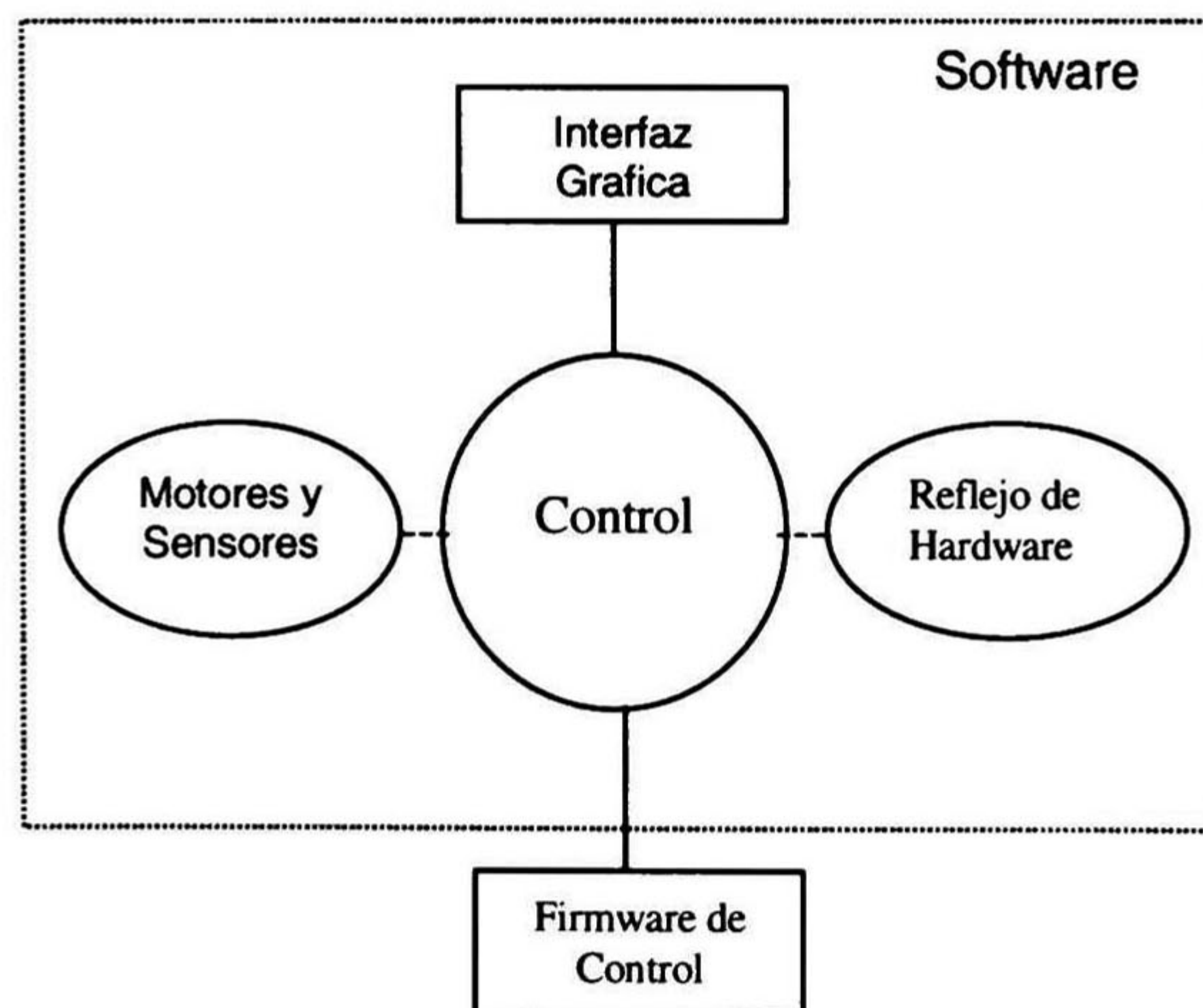


Figura 5-2 Arquitectura del software de aplicación.

5.1.1.1 Concepción en clases y objetos

Control

Constituye el elemento central del sistema. Es implementado por medio de un objeto único, derivado de la clase *Control*, que ejecuta acciones y supervisa el funcionamiento de todo el sistema. Sus principales tareas son:

1. Inicializar el sistema, mediante la creación de objetos funcionales que reflejan el hardware de la plataforma física que la herramienta soporta.
2. Implementar la interfaz con el usuario.
3. Crear, actualizar y controlar a otros objetos que representan a elementos activos del sistema (motores, sensores), de acuerdo a la interacción con el usuario y la operación del firmware de control.
4. Interpretar y codificar los datos de operación del sistema, para ser transmitido y ejecutado por el firmware de control. De igual forma se encarga de decodificar la información proveniente de este módulo. Contiene el manejador del puerto serial para la transmisión y recepción de datos.

Class Control. Clase que deriva el objeto único control. Contiene funciones propias y elementos de las clases *HW_Component*, necesarios para el adecuado funcionamiento del programa.

Sus principales métodos son:

```
void v_set_serial_communication (void);
void v_release_serial_communication (void);
void v_create_hw_sensor (void);
void v_create_hw_dc_motor (void);
void v_create_hw_stepper_motor (void);
uint8 ui8_start_a_dc_motor (uint8 ui8_dc_motor);
uint8 ui8_stop_a_dc_motor (uint8 ui8_dc_motor);
uint8 ui8_start_a_stepper_motor (uint8 ui8_stepper_motor);
uint8 ui8_stop_a_stepper_motor (uint8 ui8_stepper_motor);
uint8 ui8_stepper_speed_up (uint8 ui8_stepper_active, uint8 ui8_ramps);
uint8 ui8_stepper_speed_down (uint8 ui8_stepper_active, uint8 ui8_ramps);
uint8 ui8_give_x_number_steps (uint8 ui8_stepper_active, uint32 ui32_steps_number);
uint8 ui8_send_ramps (uint8 ui8_stepper_motor, uint8 ui8_ramp_up, uint8 ui8_ramp_down);
```

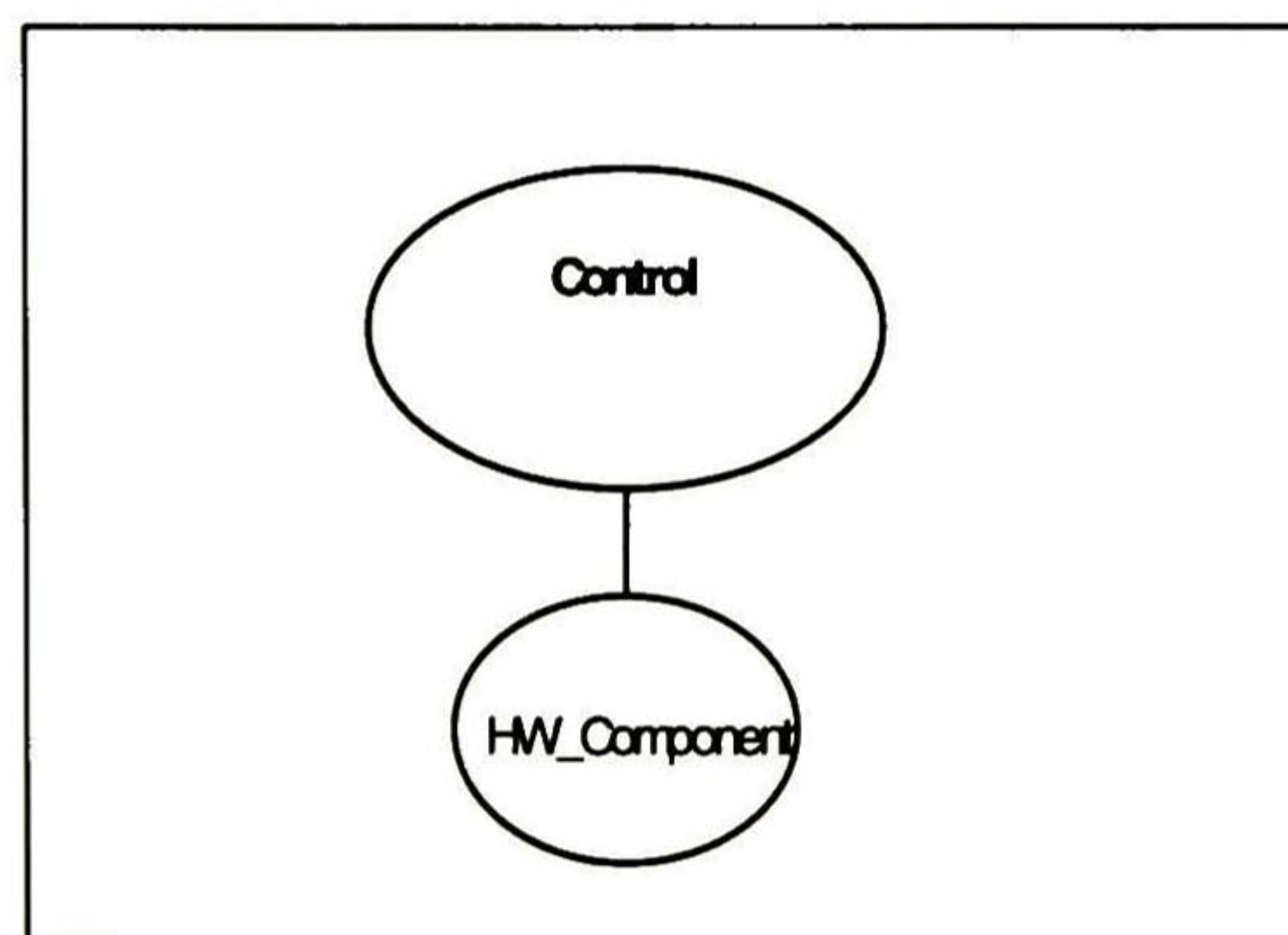


Figura 5-3 Relación entre las clases Control y HW_component.

Objetos funcionales

Se dividen en dos partes: una contiene toda la información acerca de los elementos de software que reflejan el hardware del sistema (manejadores de motor, electrónica para lectura de sensores), mientras que la segunda describe los motores y sensores que dinámicamente se activan y desactivan de acuerdo a la operación del programa.

Class HW_Component. Clase genérica utilizada para la representación en software de los elementos de la plataforma física y de los elementos creados por el usuario durante la ejecución del programa. De esta clase se derivan las clases Motor y Sensor.

Class Motor. Clase derivada de HW_Component, para la representación general de motores. De esta clase se derivan las clases DC_Motor y Stepper_Motor.

Class Sensor. Clase derivada de HW_Component, para la representación de sensores.

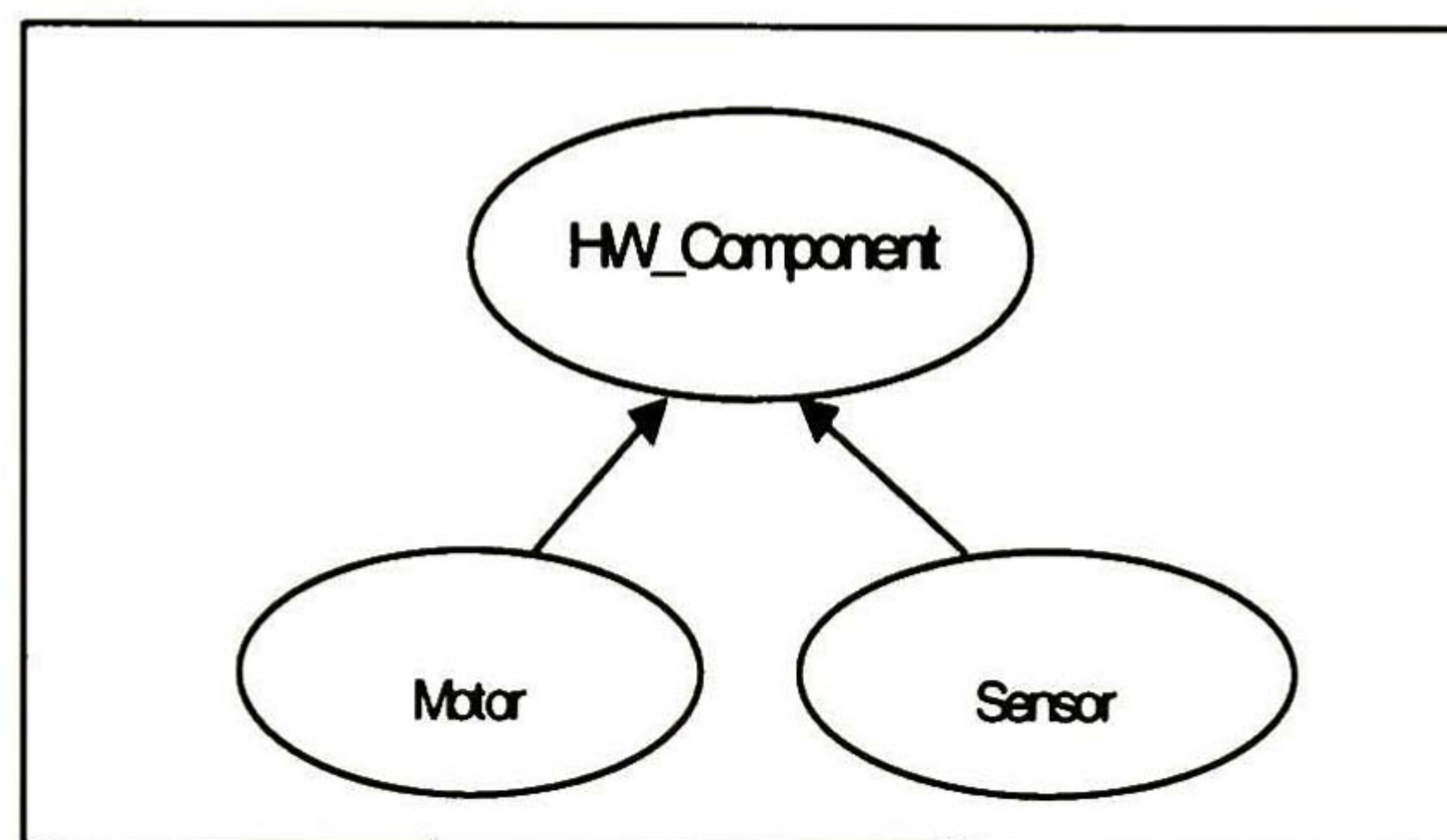


Figura 5-4 Herencia de la clase HW_Component.

Class DC_Motor (clase motor de corriente directa). Clase derivada de Motor, crea los objetos motores de corriente directa.

Class Stepper_Motor (clase motor de pasos). Clase derivada de Motor, crea los objetos motores de pasos.

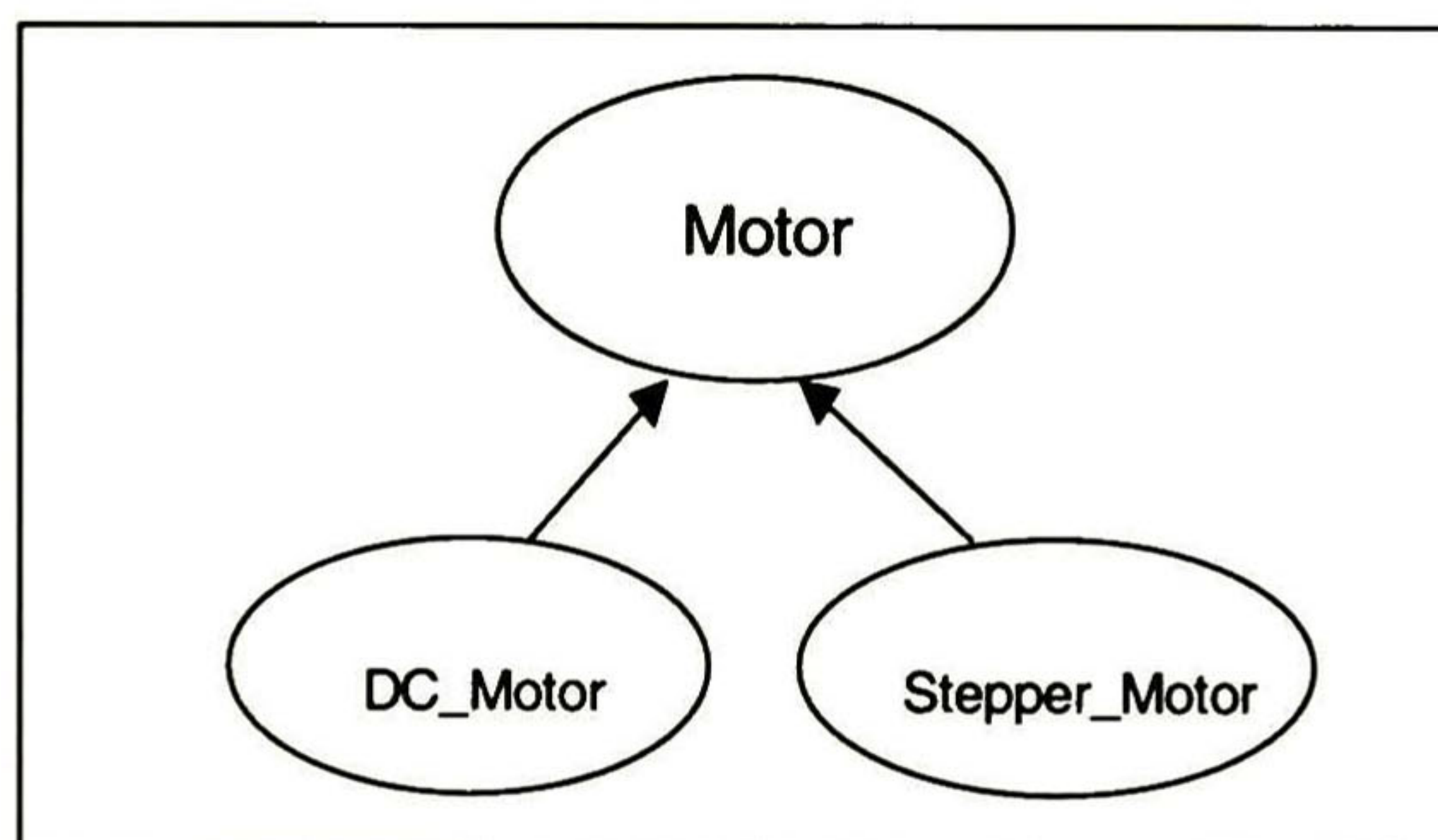


Figura 5-5 Herencia de la clase Motor.

Interfaz gráfica

Interfaz amigable y fácil de usar que permite al usuario configurar la herramienta para ejercitar motores de corriente directa y de pasos, así como conocer el estado de los sensores.

5.1.1.2 Características del software

El software fue desarrollado en objetos, con la herramienta C++ Builder 3.0 y se ejecuta como una aplicación del sistema operativo Windows 95/98/NT [14]. Las principales características de este programa son:

- Clases: 6
- Clases recipientes: 22
- Líneas de código: 2,050
- Líneas de compilación: 157,000
- Volúmen del ejecutable: 475 Kbytes

5.1.2 Interfaz de usuario

El presente capítulo describe paso a paso el contenido de la interfaz gráfica del programa y su interacción directa con el usuario, necesaria para trabajar con sensores y motores de corriente directa y pasos que conforman el sistema electromecánico de un manejador de papel Hewlett Packard. La interfaz gráfica, amigable y fácil de usar del software del sistema, está conformada por un programa que permite configurar y operar el sistema de acuerdo a los requerimientos del usuario. Su funcionalidad se describe basándose en tres etapas: inicialización, configuración y operación [15].

5.1.2.1 Inicialización

Al iniciar una sesión, no existe ningún motor o sensor activo, por lo que el programa espera por la creación y configuración de estos elementos. El programa muestra información y capacidad acerca del sistema (identificación del producto, plataforma física que soporta, número de motores y sensores que controla) y, su estado actual, incluyendo la notificación de errores. Presenta, además la capacidad para recuperar una sesión previa, definida por el usuario, con un número de motores y sensores activos.

La primera pantalla del programa (Figura 5-6), ofrece la opción de iniciar con una nueva sesión (motores de pasos y motores de corriente directa inactivos) o activar una sesión previa, con motores y sensores inicializados y configurados por el usuario. Al fondo aparece el formato general de la interfaz en estado inactivo.

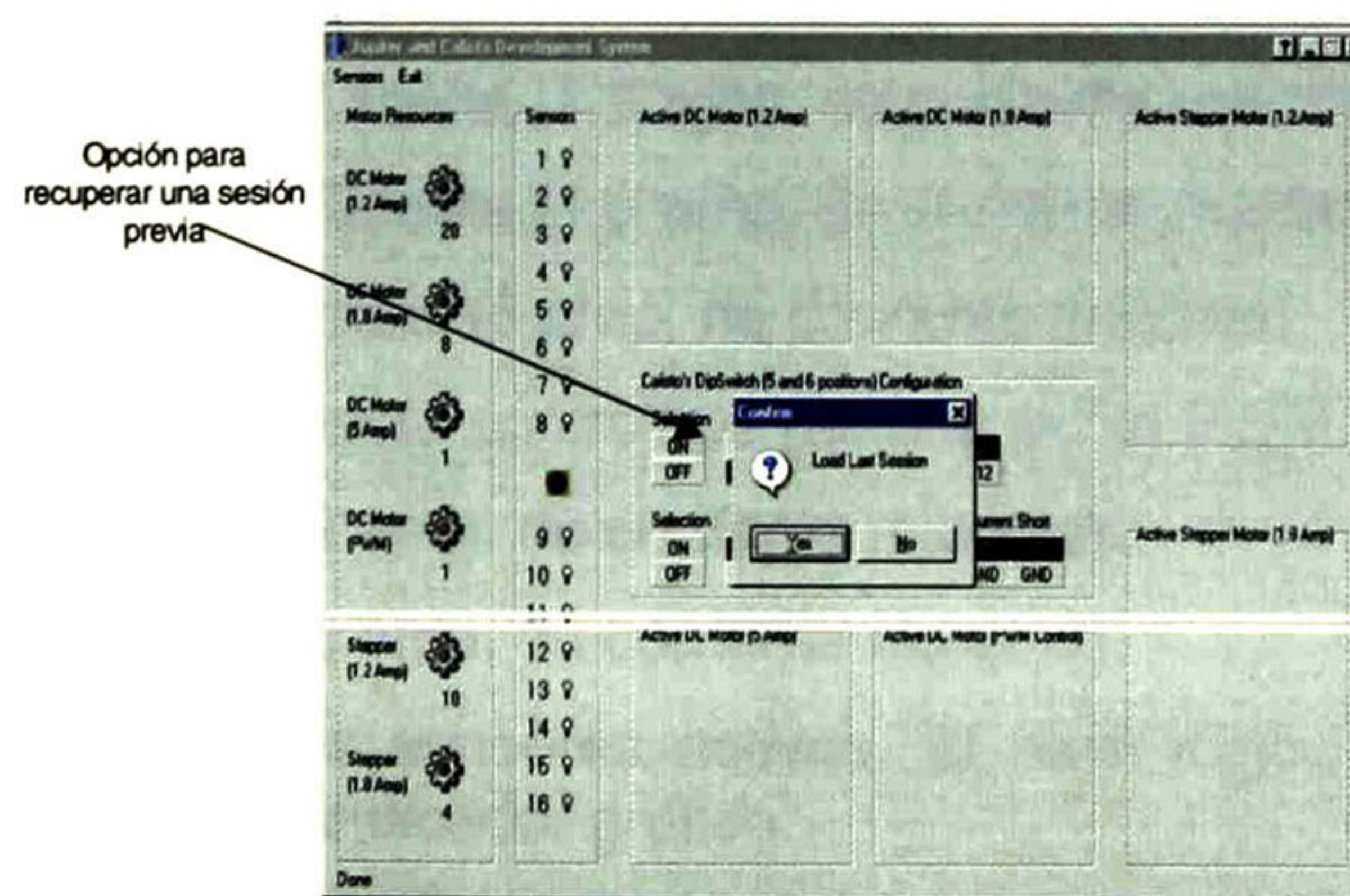


Figura 5-6 Opción para recuperar una sesión.

Si el usuario desea recuperar una sesión, el software recupera de un archivo la información referente a cada motor y sensor activado, para mostrarlo en pantalla. El programa actualiza su estado de acuerdo a estos parámetros y continúa su ejecución con la etapa de operación que se describe más adelante.

Si por el contrario, la selección es iniciar con una nueva sesión, el programa inmediatamente activa la pantalla de fondo, que ofrece al usuario su configuración inicial, por medio de la cual se muestra el número de motores y sensores que se pueden utilizar, así como la información de sus principales parámetros.

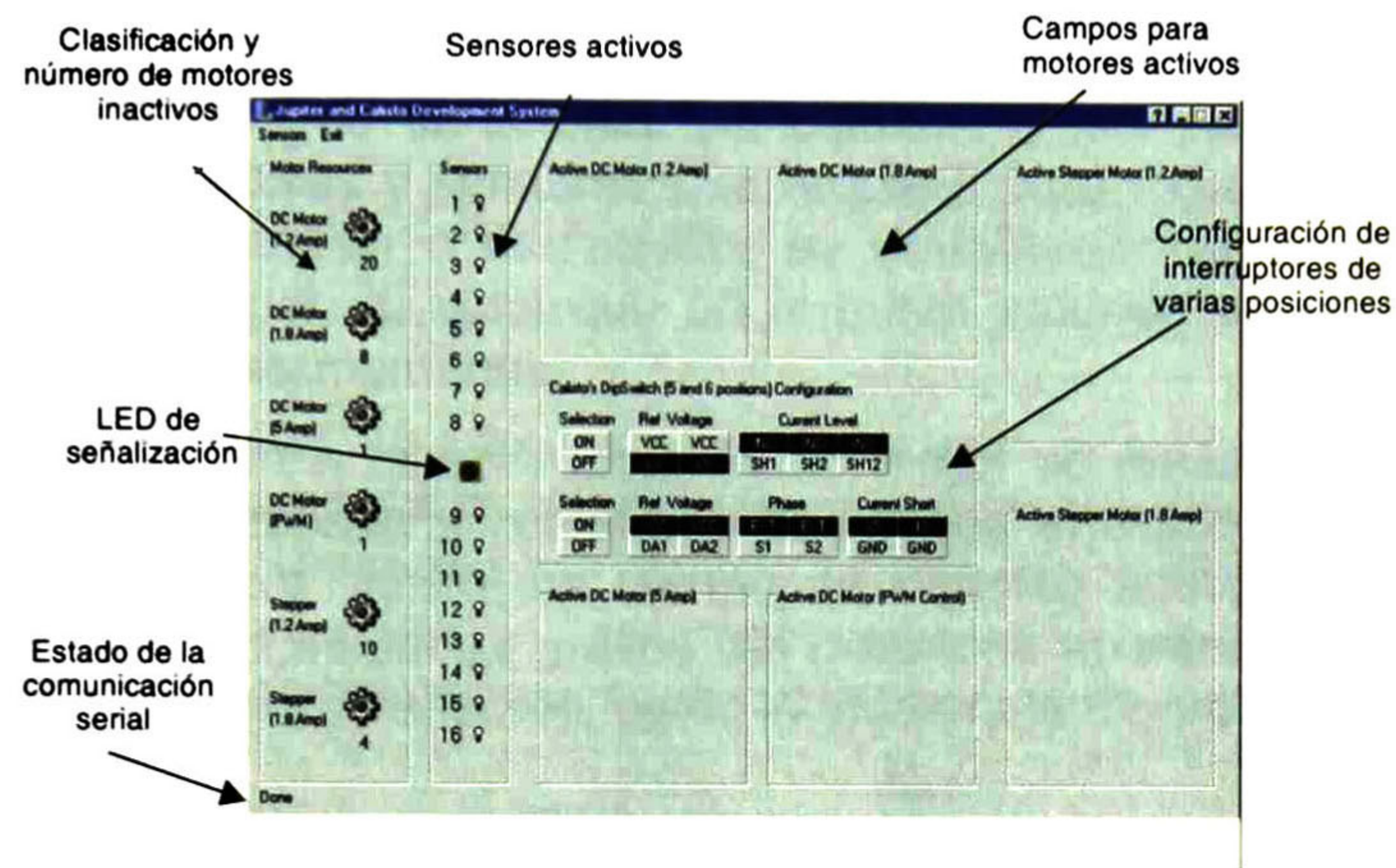


Figura 5-7 Pantalla inicial de la aplicación.

La información desplegada por esta pantalla es la siguiente:

- Estado de la comunicación serial. Indica si el programa puede establecer la comunicación serial con la plataforma física.
- Campo de texto con la información acerca de la capacidad actual de la herramienta:
 - 31 Motores de Corriente Directa.
 - 14 Motores de Pasos.
 - 16 Sensores.
- Campo que contiene el número de motores inactivos que pueden ser utilizados por el usuario, y que son clasificados de acuerdo a los siguientes parámetros:
 - 20 Motores de Corriente Directa (1.2 amp. de corriente máxima).
 - 8 Motores de Corriente Directa (1.8 amp. de corriente máxima).
 - 1 Motor de Corriente Directa (5 amp. de corriente máxima).
 - 1 Motor de Corriente Directa controlado por una señal PWM.
 - 10 Motores de Pasos (1.2 amp. de corriente máxima).
 - 4 Motores de Pasos (1.8 amp. de corriente máxima).
- Campo con que contiene los sensores activos. El nivel lógico de cada sensor es desplegado por medio de un indicador gráfico.
- Opción (Sensors) para activar y desactivar la visualización de los sensores.

- f) Configuración de los interruptores de varias posiciones del hardware de aplicación.
- g) Campos para los motores activos, clasificados de acuerdo a su tipo.
- h) Opción para salir (Exit) de la aplicación.

5.1.2.2 Configuración

Enseguida, el usuario activa, desactiva y configura motores y sensores de acuerdo a su necesidad. La notificación del estado y los errores de la aplicación permanece activa. El programa muestra la información más relevante de cada motor o sensor habilitado.

La etapa de configuración se caracteriza principalmente por que es aquí cuando el usuario determina los motores y sensores que requiere para trabajar. En el caso de los sensores, la opción es habilitar o deshabilitar su visualización por medio de la opción 'Sensors' del menú principal. Al igual que en el punto anterior, el nivel lógico de cada sensor es actualizado constantemente.

Para el caso de los motores, la activación de un motor se realiza colocando el cursor sobre el tipo de motor deseado y pulsando dos veces el botón izquierdo del mouse. Automáticamente aparecerá dentro del campo de motores activos correspondiente, la figura del motor activado. La interfaz gráfica del programa aparece, después de habilitar un cierto número de motores y sensores, como lo muestra la Figura 5-8.

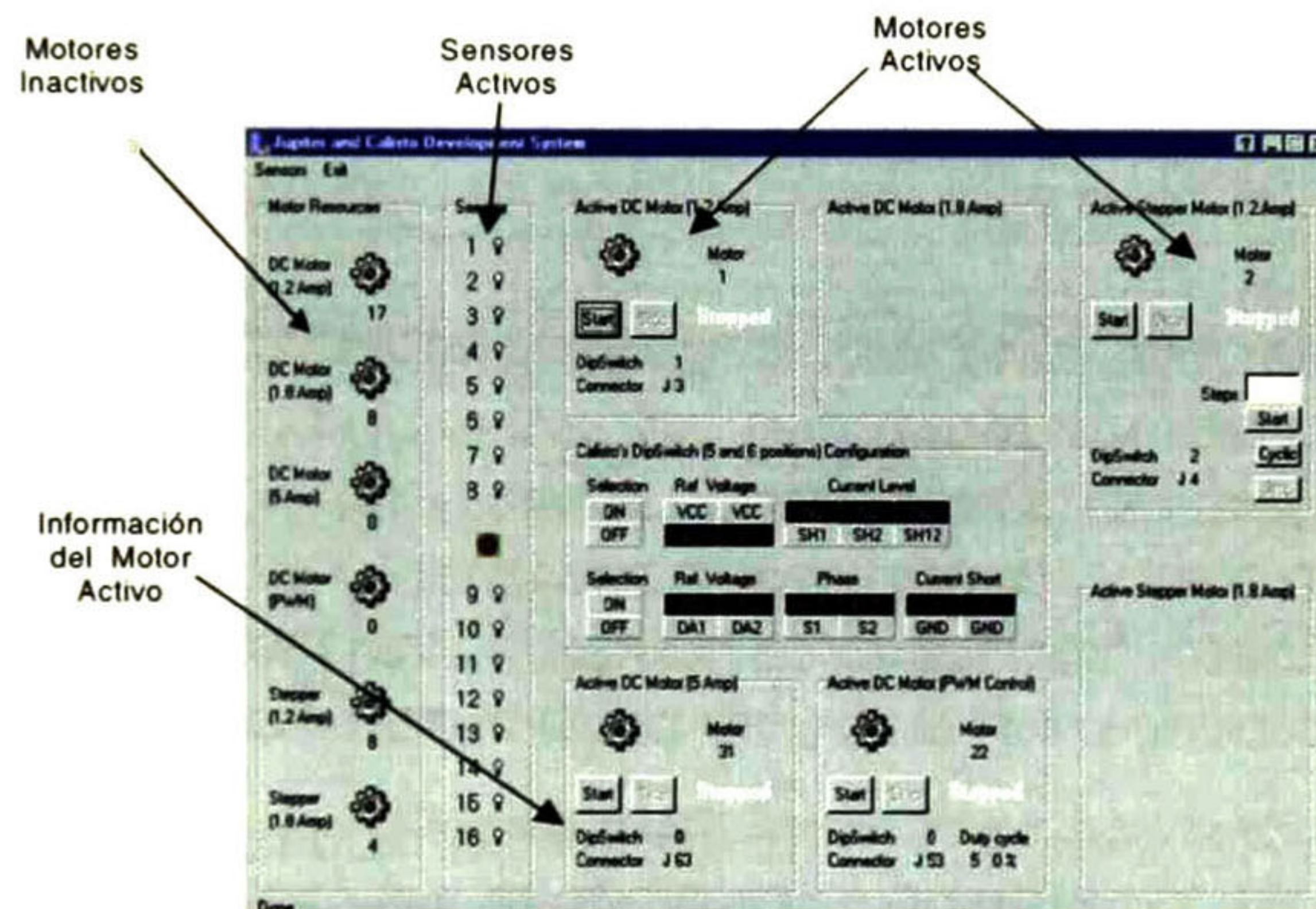


Figura 5-8 Habilidad de motores y sensores.

Si más de dos motores de un mismo tipo son habilitados, el menú de configuración de los campos para motores activos ofrece la capacidad para seleccionar cualquiera de ellos. Es importante notar además, que al seleccionar un motor, automáticamente se reduce el número de motores de su mismo tipo, que todavía se pueden activar. La interfaz proporciona al usuario la capacidad para:

- a) Visualizar la información esencial de cada motor o sensor activo:

- Motor.
 - Número y tipo.
 - Estado.
 - Conector de salida.
 - Elemento de configuración asociado (interruptor de varias posiciones).

- Sensor.
 - Número.
 - Estado.
 - Conector de entrada.
- b) Activar y desactivar motores de pasos, motores de corriente directa y sensores.
- c) Configurar los parámetros de operación más importantes de los motores de corriente directa o motores de pasos activos, mediante los menús y submenús de configuración
- d) Para activar el menú principal solo es necesario colocar el cursor sobre la figura del tipo de motor a configurar y pulsar la tecla derecha del mouse.
- Motor de corriente directa (1.2 amp, 1.8 amp y 5 amp).
 - Motor activo.
 - Dirección.
 - Nivel de Corriente.
 - Eliminar motor.
 - Salir de menú.
- Motor de corriente directa controlado por señal PWM.
 - Motor activo.
 - Dirección.
 - Nivel de Corriente.
 - Ciclo de trabajo.
 - Eliminar motor.
 - Salir de menú.
- Motor de corriente directa controlado por señal PWM.
 - Motor activo.
 - Dirección.
 - Nivel de Corriente.
 - Ciclo de trabajo.
 - Rampas de aceleración y desaceleración.
 - Eliminar motor.
 - Salir de menú.

Si la selección del menú es 'Select a Motor', a continuación aparecen todos los motores habilitados de este tipo. La selección de un motor lo activa para ser configurado y operado.

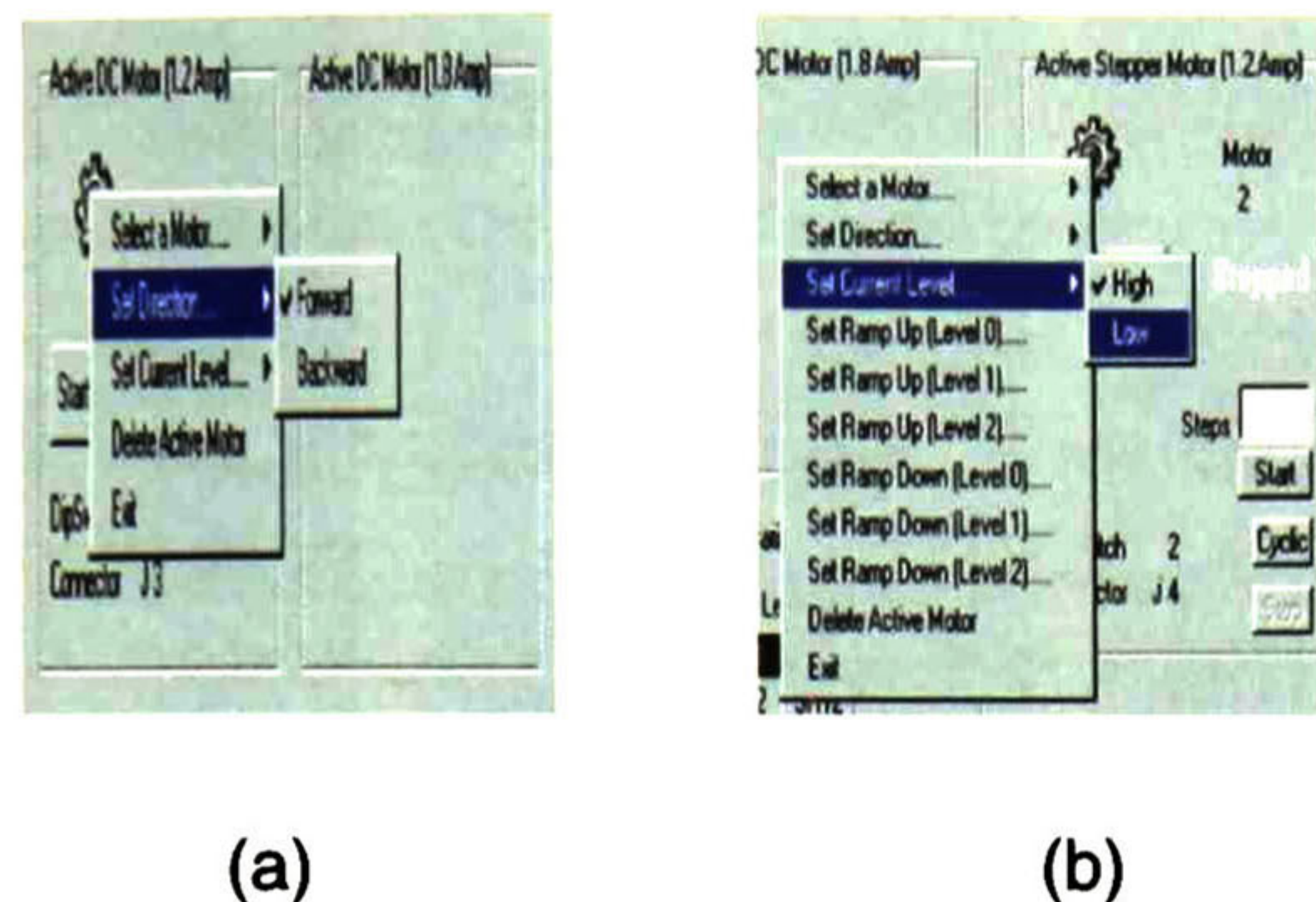


Figura 5-9 Menús de configuración.

(a) Motor de corriente directa, (b) Motor de pasos.

Para el caso de las opciones 'Set Ramp Up' y 'Set Ramp Down' aparece una tabla que es utilizada por el usuario de acuerdo a su necesidad. Los datos de la tabla reflejan los tiempos en números enteros de milisegundos (ms), entre los pasos de un motor de pasos, ya sea para rampas de subida (aceleración) o bajada (desaceleración). El dato final de la rampa siempre deberá ser 0 y se pueden definir rampas de hasta 255 niveles. La figura muestra la tabla de configuración de una rampa de un motor de pasos.

- e) Activar y desactivar sensores. Los sensores no requieren configuración y solo existe la opción para hacer visible o no sus niveles lógicos actuales.
- f) Guardar la sesión actual, incluyendo motores y sensores habilitados con su respectiva configuración.
- g) Salir de la aplicación.
- h) La notificación activa de mensajes de información y error para el usuario.

5.1.2.3 Operación

El usuario interactúa directamente con los elementos habilitados. Maneja motores de corriente directa, motores de pasos y conoce el estado de sensores. Provee, por medio de su interfaz, la capacidad para:

- a) Arrancar y parar cada motor habilitado y configurado. En el caso de motor de pasos, opción para dar un cierto número de pasos definidos por el usuario en modo simple o cíclico y capacidad de operar sobre tres niveles diferentes de velocidad definidos por sus rampas de subida y rampas de bajada.

La Figura 5-10 muestra las interfaces de operación para un motor de corriente directa, un motor de pasos con la opción para dar 'x' pasos y un motor de pasos operando de acuerdo a sus diferentes niveles de velocidad.

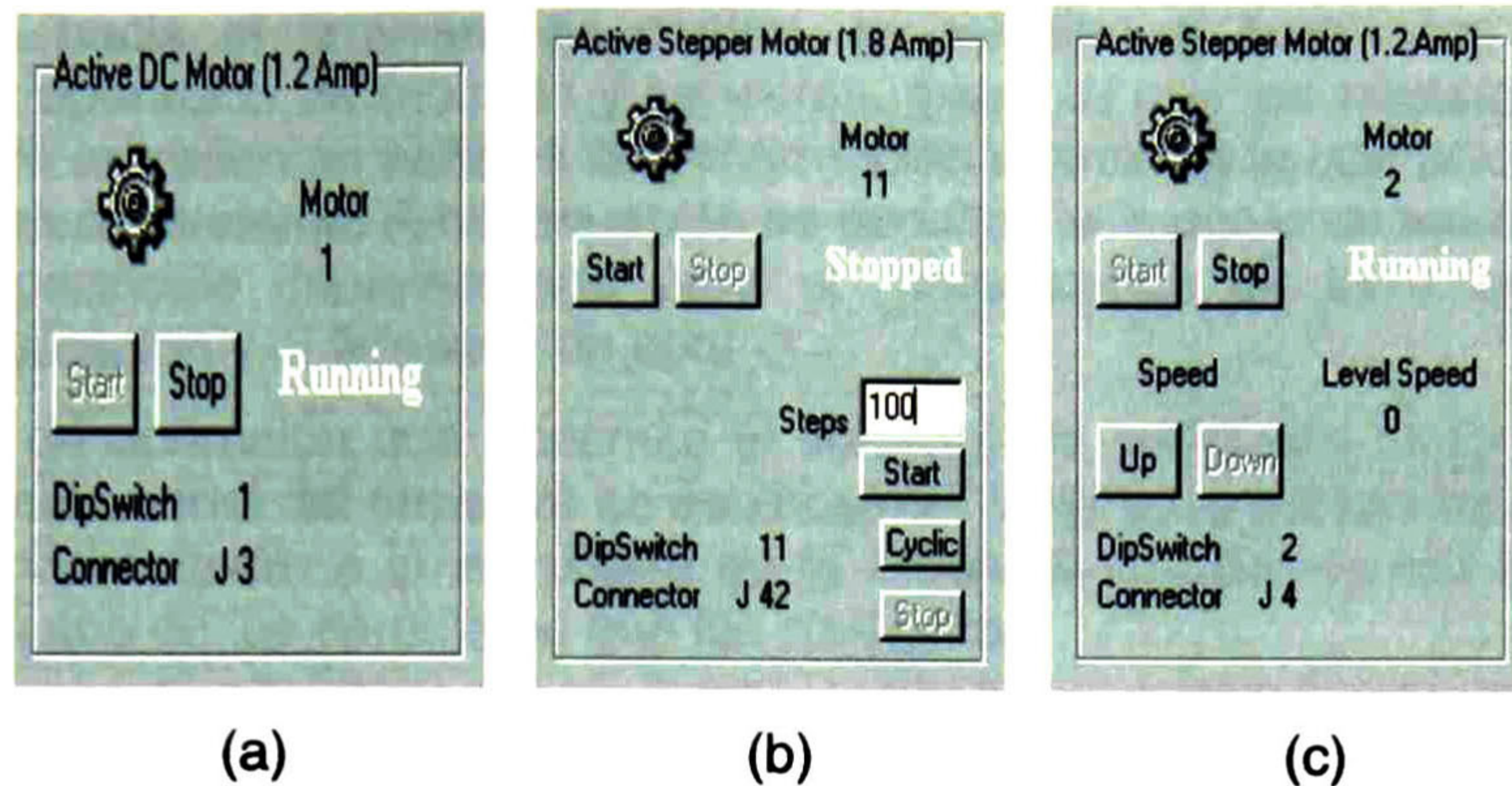


Figura 5-10 (a) Operación de un motor de corriente directa, (b) Opción de un motor de pasos para dar un determinado número de pasos en forma simple o de manera cíclica (hacia adelante y hacia atrás) y (c) Operación de un motor de pasos con 3 diferentes niveles de velocidad.

- b) Guardar la sesión actual, incluyendo motores y sensores habilitados con su respectiva configuración (Figura 5-11).

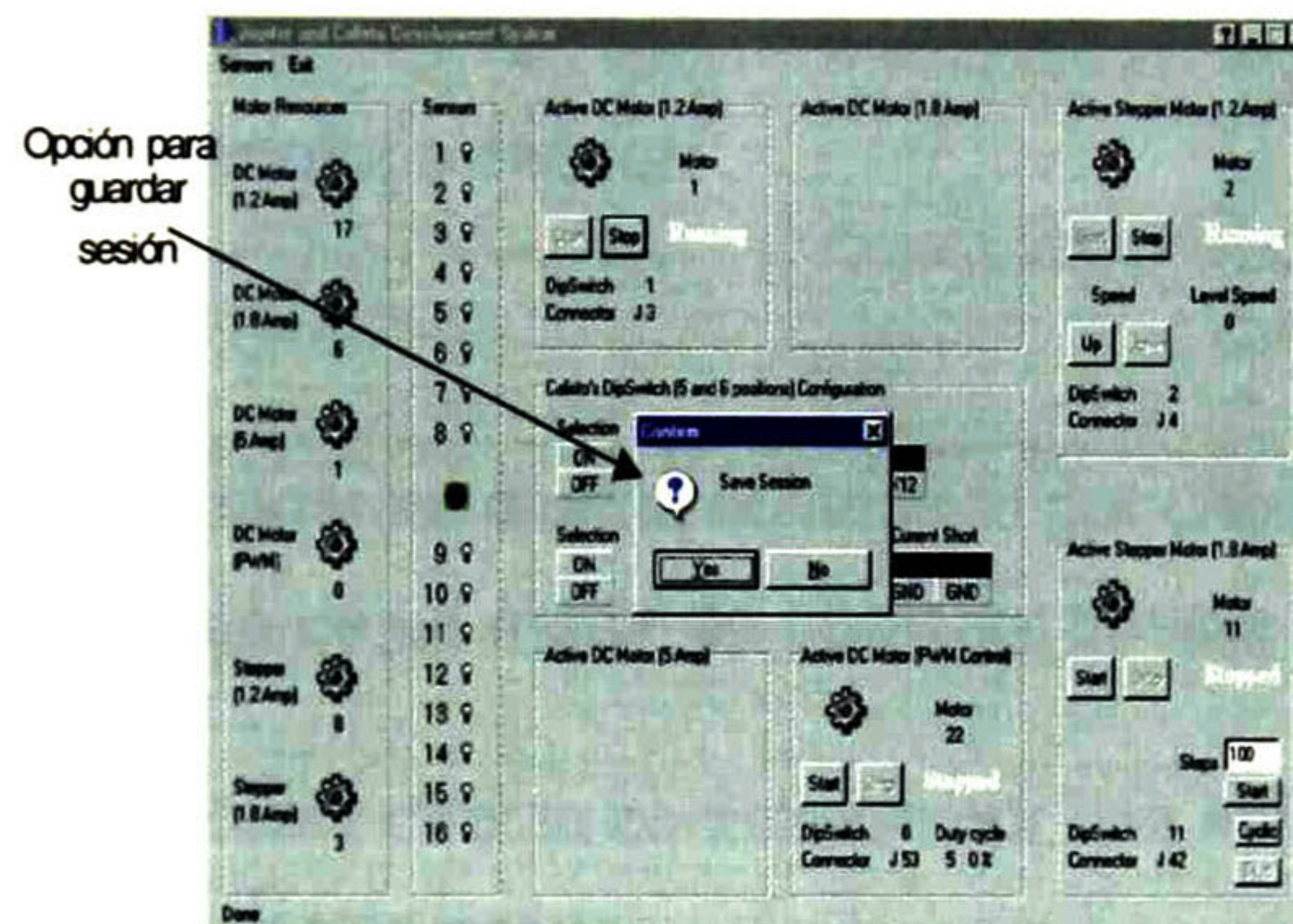


Figura 5-11 Opción para guardar una sesión.

5.1.3 Comunicación con el firmware de control

Una vez que el usuario ha configurado al sistema para trabajar de acuerdo a su necesidad, el software identifica y codifica, basándose en un protocolo definido, los parámetros que requiere transmitir hacia el módulo de firmware para que implemente la funcionalidad deseada. El medio físico que hace posible el paso de información de un módulo a otro es el puerto serial de la PC, conectado al puerto serial de la plataforma física donde el firmware de control. La interfaz física de transmisión se implementa de acuerdo al protocolo de comunicación RS232C [16].

La información generada por el programa representa el elemento principal para la creación del código que es transmitido hacia el firmware de control del sistema a través de comandos, ya que determina el número y configuración de motores y sensores que el usuario desea operar. Sin embargo, los datos específicos requeridos y proporcionados por el firmware de control determinan, en su mayor parte, el formato de la interfaz.

La información hacia el firmware de control, la constituyen comandos de creación, configuración y operación de motores y sensores, mientras que los mensajes que viajan hacia el software representan señales de confirmación acerca de la operación y estado de los componentes del sistema. A continuación se describe el formato de los comandos que conforman el protocolo implementado para el intercambio de información entre el programa de aplicación y el firmware de control.

Los comandos de operación que transmite el software de aplicación hacia el manejador de la comunicación serial del firmware se codifican en cadenas o tramas de caracteres en formato ASCII, de acuerdo a la estructura de la Figura 5-12. Enseguida se describe la función de cada uno de los caracteres que los conforman.

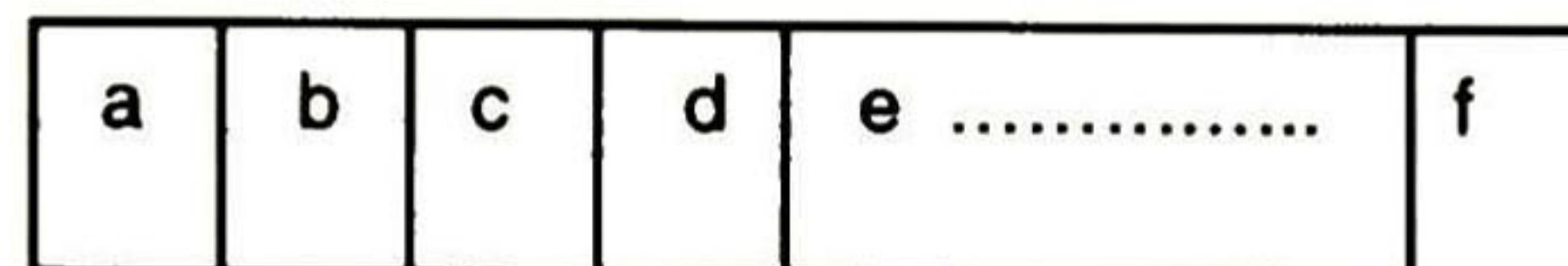


Figura 5-12 Estructura de un comando de operación del software de aplicación.

a. *Inicio de comando.*

b. *Componente electromecánico.* Sensor, motor de corriente directa, motor de pasos.

c. *Comando.*

Conocer estado de un motor o sensor.

Arrancar un motor.

Parar un motor.

Arrancar un motor de corriente directa controlado por una señal de PWM.

Cambiar ciclo de trabajo de un motor de corriente directa controlado por una señal PWM.

Parar un motor de corriente directa controlado por una señal de PWM.

Definir rampa de un motor de pasos.

Mover un motor de pasos 'x' número de pasos.

Cambiar el nivel de velocidad de un motor de pasos.

d. *Número de componente electromecánico.*

e. *Parámetros adicionales.* Dirección, nivel de corriente, ciclo de trabajo, tipo de rampa, número de rampa, tamaño de rampa, datos de la rampa, nivel de velocidad.

f. *Fin de comando.*

La Tabla 5-1 describe a detalle los comandos transmitidos desde la aplicación residente de una PC, hacia el firmware de control del sistema.

La comunicación física entre el programa de la herramienta de desarrollo y el firmware de control se implementa por medio del puerto serial de la PC (protocolo estándar RS232C) y, lógicamente, a través de un manejador del puerto.

El manejador del puerto serial está basado en la interfaz para la programación de una aplicación (API por sus siglas en inglés) que proporciona la plataforma Win32, en su apartado referente a la comunicación.

Comando	Código / Parámetros	Fuente	Descripción
Estado de sensor	'c0enx'	PC	La aplicación indica al sistema que le informe acerca del estado del sensor 'n'
Estado de motor de corriente directa	'c1enx'	PC	La aplicación indica al sistema que le informe acerca del estado del motor de corriente directa 'n'
Arrancar motor de corriente directa	'c1andrx'	PC	La aplicación indica al sistema que arranque el motor de corriente directa 'n' con la dirección 'd' y un nivel de corriente 'r'
Parar motor de corriente directa	'c1pnx'	PC	La aplicación indica al sistema que pare el motor de corriente directa 'n'
Arrancar motor de corriente directa controlado por señal de PWM	'c1wndtx'	PC	La aplicación indica al sistema que arranque el motor de corriente directa 'n' controlado por una señal de PWM, con la dirección 'd' y un ciclo de trabajo 't'
Cambiar ciclo de trabajo de un motor de corriente directa controlado por señal de PWM	'c1yntx'	PC	La aplicación indica al sistema que cambie el ciclo de trabajo del motor de corriente directa 'n' controlado por una señal de PWM, a un valor 't'
Parar motor de corriente directa controlado por señal de PWM	'c1pnx'	PC	La aplicación indica al sistema que pare el motor de corriente directa 'n' controlado por una señal de PWM
Estado de motor de pasos	'c2enx'	PC	La aplicación indica al sistema que le informe acerca del estado del motor de pasos 'n'
Arrancar motor de pasos	'c2andrx'	PC	La aplicación indica al sistema que arranque el motor de pasos 'n' con la dirección 'd' y un nivel de corriente 'r'
Definir rampa de aceleración o desaceleración de un motor de pasos	'c2rnzdatax'	PC	La aplicación indica al sistema que defina de acuerdo a 'data', la rampa de aceleración o desaceleración 'z' para el motor de pasos 'n'
Mover un motor de pasos	'c2mndsx'	PC	La aplicación indica al sistema que mueva el motor 'n', 's' número de pasos con una dirección 'd'
Cambiar nivel de velocidad de un motor de pasos	'c2snax'	PC	La aplicación indica al sistema que cambie al nivel de velocidad 'a', el motor de pasos 'n'
Parar motor de pasos	'c2pnx'	PC	La aplicación indica al sistema que pare el motor de pasos 'n'

Tabla 5-1 Comandos de operación del firmware de control.

Las funciones de entrada y salida de archivo (CreateFile, CloseHandle, ReadFile, ReadFileEx, WriteFile and WriteFileEx) proveen la interfaz básica para activar y desactivar un canal de comunicación. Además, proporciona otros elementos necesarios para configurar y operar dicha fuente de comunicación [17].

Las tareas que esta interfaz implementa son:

- a) Abrir un manejador para un canal específico de comunicación.
- b) Configurar un canal de comunicación serial.
- c) Transmitir y recibir datos sobre un canal de comunicación.
- d) Monitorear un determinado número de eventos que pueden ocurrir durante la operación de un canal de comunicación.
- e) Enviar comandos de control sobre el manejador de dispositivo (device driver) del canal de comunicación utilizado.

Las funciones de la interfaz de comunicación de la API, utilizadas por el programa para el manejo del puerto serial son:

CreateFile(), crea el manejador para el puerto de comunicación serial.

ReadFile(), lee los datos almacenados en el contenedor de entrada del puerto serial.

WriteFile(), escribe datos sobre el contenedor de salida del puerto serial.

CloseHandle(), destruye el manejador del puerto de comunicación serial.

SetCommTimeOuts(), determina si el manejador puede activar el puerto serial dentro de un cierto tiempo.

SetCommState(), determina si el manejador puede activar correctamente el puerto serial dentro de un cierto tiempo.

Los elementos de la interfaz de comunicación del API, utilizados por el programa para el manejo del puerto serial son:

HANDLE, manejador del puerto de comunicación serial.

DCB, estructura que define los parámetros de control del puerto serial.

COMMTIMEOUTS, estructura utilizada para configurar los parámetros de tiempos de espera para el puerto serial.

5.1.4 Pruebas del software

Las pruebas del software del sistema pretenden determinar si la aplicación cumple con sus requerimientos de diseño. La metodología de prueba se describe a continuación:

1. El primer paso consiste en verificar la comunicación serial entre la herramienta de desarrollo y el software del sistema; para esto es necesario transmitir repetitivamente diversos comandos conocidos hacia el firmware de control, para que ejecute las tareas que le corresponden. Mover un motor o monitorear un sensor son algunas de los ejercicios que se pueden implementar y que aseguran el paso de parámetros eficiente entre los dos módulos.
2. La siguiente prueba requiere de aislar el programa del firmware de control y activar la pantalla que implementa la interfaz con el usuario. El objetivo es comprobar que el software proporciona al usuario las herramientas para habilitar, configurar y operar motores de pasos, motores de corriente directa y sensores, además de que cada uno de los procesos cumpla con los requerimientos de su diseño.

3. Enseguida es importante verificar que la información que refleja el estado real del sistema, mantenga al usuario con el control absoluto de la herramienta de desarrollo en cualquier momento. Este proceso permite también, determinar la robustez real del sistema ante la infinidad de acciones, esperadas e inesperadas del usuario.
4. Una vez que el programa cumple íntegramente con los procesos de verificación antes mencionados, el siguiente paso consiste en interconectar los módulos hardware, firmware y software para que un usuario pruebe cada una de las opciones proporcionadas por el sistema: habilita, deshabilita, configura, arranca, motores de corriente directa y motores de pasos; activa, desactiva, lee el estado de sensores; utiliza toda la funcionalidad proporcionada por la herramienta para probar el sistema electromecánico de un manejador de papel Hewlett Packard.
5. Ya que el software de aplicación satisface las tareas encomendadas por el usuario, la herramienta puede ser utilizada con los fines para los cuales es diseñada e implementada. Es importante verificar la correcta operación del programa, ya que representa el principal medio para aprovechar cada una de las funciones ofrecidas por la plataforma. De igual forma es necesario repetir este procedimiento, ejecutando el programa de manera independiente sobre cada uno de los diferentes sistemas operativos que soporta: Windows 95, Windows 98 y Windows NT, a fin de reconocer su compatibilidad.

5.2 Protocolo de comunicación RS232C

El protocolo de comunicación RS232C es un estándar que implementa la comunicación serial y asíncrona entre dos dispositivos. En este tipo de enlace, la información se transmite bit por bit (serial) y no en base a un tiempo definido (asíncrono). La transferencia de datos puede iniciar en cualquier instante y es una tarea del receptor detectar cuando un mensaje inicia y termina [18].

5.2.1 Principio de operación

El estándar RS232C describe un método de comunicación donde la información es enviada bit por bit por medio de un canal físico. La información en palabras de datos, donde la longitud de estas palabras es variable. En las PC's, la longitud oscila entre los 5 y los 8 bits. Esta longitud representa el contenido puro de información en cada palabra. Para una mayor confiabilidad en la transmisión algunos bits pueden ser añadidos con propósitos de verificación de errores y sincronización. Es importante, que el transmisor y el receptor utilicen el mismo número de bits, ya que de otra manera la palabra de datos puede ser malinterpretada, o no identificada.

Para la comunicación síncrona, el sistema debe contar con un reloj o un disparo de una señal que indiquen el inicio de cada transferencia. La ausencia de esta señal de reloj, hace que la comunicación asíncrona a través de un canal sea mucho más barata. Un punto importante es que se requieren del mínimo de líneas o cables, aunque tiene la desventaja de que el receptor puede iniciar a recibir la información en forma errónea. Para evitar al máximo éste problema, se utiliza la re-sincronización, la cual se traduce en una pérdida de tiempo, ya que durante estos lapsos de tiempo no se transmite información alguna. Otra desventaja es que se requieren de bit adicionales en la trama de datos que indiquen el inicio y el fin de la información. Estos bits incrementan el ancho de banda requerido para transmitir.

Los bits de datos son enviados con una determinada frecuencia (baud rate). Tanto el transmisor como el receptor deben estar programados para trabajar a la misma frecuencia. Después de el primer bit es recibido, el receptor calcula el momento en el que el siguiente dato va a ser recibido, para lo cual detectará los niveles de voltaje en ese momento. Este nivel de voltaje puede tener solo dos estados, el estado de encendido, también conocido como marcado, y el estado de apagado, conocido también como espaciado. Cuando la línea se encuentra desocupada, se mantiene en el estado de marcado [18].

La siguiente figura muestra la manera en que se compone una trama de datos para este protocolo.

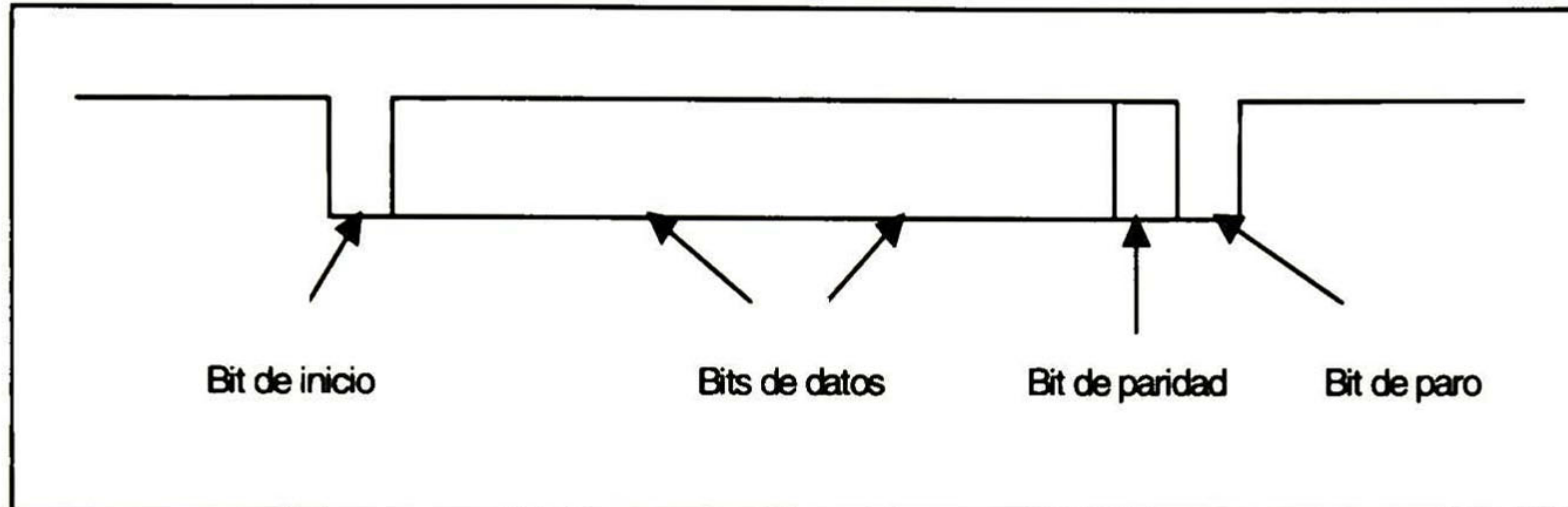


Figura 5-13 Trama de datos del protocolo RS232C.

RS232C define un tipo de comunicación asíncrona, lo que significa que una palabra de datos puede aparecer sobre la línea de transmisión en cualquier momento. Si ocurre esto, el receptor puede tener algunos problemas para saber cual es el primer bit a recibir. Para solucionar este problema cada palabra de datos inicia con un bit de atención, el cual es conocido como el bit de inicio, el cual está definido como un espaciado, debido a que en estado desocupado, el estado es marcado, el inicio de bit es fácilmente identificado.

Enseguida del bit de inicio, los bits de datos son enviados por el bus, en donde un bit con valor 1 causa que la línea vaya al estado marcado, mientras que cada bit con valor 0 lógico es representado por un espaciado; el bit menos significativo siempre se envía primero.

Después de los datos, y con el fin de detectar errores, es posible agregar un bit extra a la palabra transmitida (bit de paridad). El transmisor calcula su valor dependiendo de la información enviada; el receptor lleva a cabo la misma operación y determina si el valor de su bit de paridad concuerda con el calculado.

Por último, suponga que el receptor pierde el bit de inicio debido al ruido presente en la línea de transmisión, e inicia con el segundo nivel de apagado. Esto causa que los datos sean mal interpretados por el receptor. Debido a esto, debe de existir un mecanismo que vuelva a sincronizar la comunicación. Para hacer esto, se implementa lo que se conoce como ventaneo o manejo de tramas, las cuales están formadas por todos los bits que se encuentran entre los bits de inicio y de paro. El período de tiempo entre estos dos es una constante definida por su tasa de transmisión (baud rate), el número de bits de datos y el bit de paridad. Si esta constante de tiempo cambia, entonces se detecta una falla en la sincronización. Para volver a una correcta transmisión, a continuación se debe volver a sincronizar la transmisión transmitiendo el bit de inicio y detectando de nuevo los bits de paro.

El estándar RS232C describe un método de comunicación capaz de transmitir sobre una gran diversidad de ambientes. Esto tiene un gran impacto en los niveles máximos de voltajes y el número de pines de transmisión.

Los niveles de las señales pueden ser dos: para un bit en 1 lógico o estado marcado, el cual es identificado como un voltaje negativo, y el de un bit 0 lógico o estado espaciado, al cuál le corresponde un nivel positivo. El voltaje para un bit 1 lógico es de +3 hasta +12 V, mientras que para el estado 0 lógico, éste va desde -3 a -12 V; la diferencia entre +3 y -3 V es utilizada para eliminar los problemas de ruido sobre la línea.

La longitud máxima permisible para las líneas en este estándar de comunicación es de 50 pies, que es igual a una capacitancia de 2500pF. Debido a que la limitante la constituye este factor, existen ya líneas que pueden llegar a alcanzar mayores longitudes, para este mismo valor de capacitancia, lo que se traduce en el manejo de mayores distancias de transmisión.

5.2.2 Implementación

Los canales 0 (TX0, RX0) y 1 (TX1, RX1) de comunicación serial del microprocesador son utilizados para crear su interfaz con una PC, por medio del protocolo de comunicación RS232. A través de elementos de configuración se selecciona uno de los dos canales del microprocesador. El circuito integrado LT1180 acopla las señales lógicas del micro, a los niveles analógicos establecidos por el protocolo RS232C y viceversa [19].

La implementación lógica del protocolo se realiza mediante el manejador del puerto serial que recibe los datos enviados por la PC y provee la funcionalidad necesaria para transmitir información por medio de este formato. Este manejador forma parte del módulo de comunicación integrado del microprocesador H8S/2357. La Figura 5-14 muestra el diagrama a bloques de los componentes que implementan esta funcionalidad.

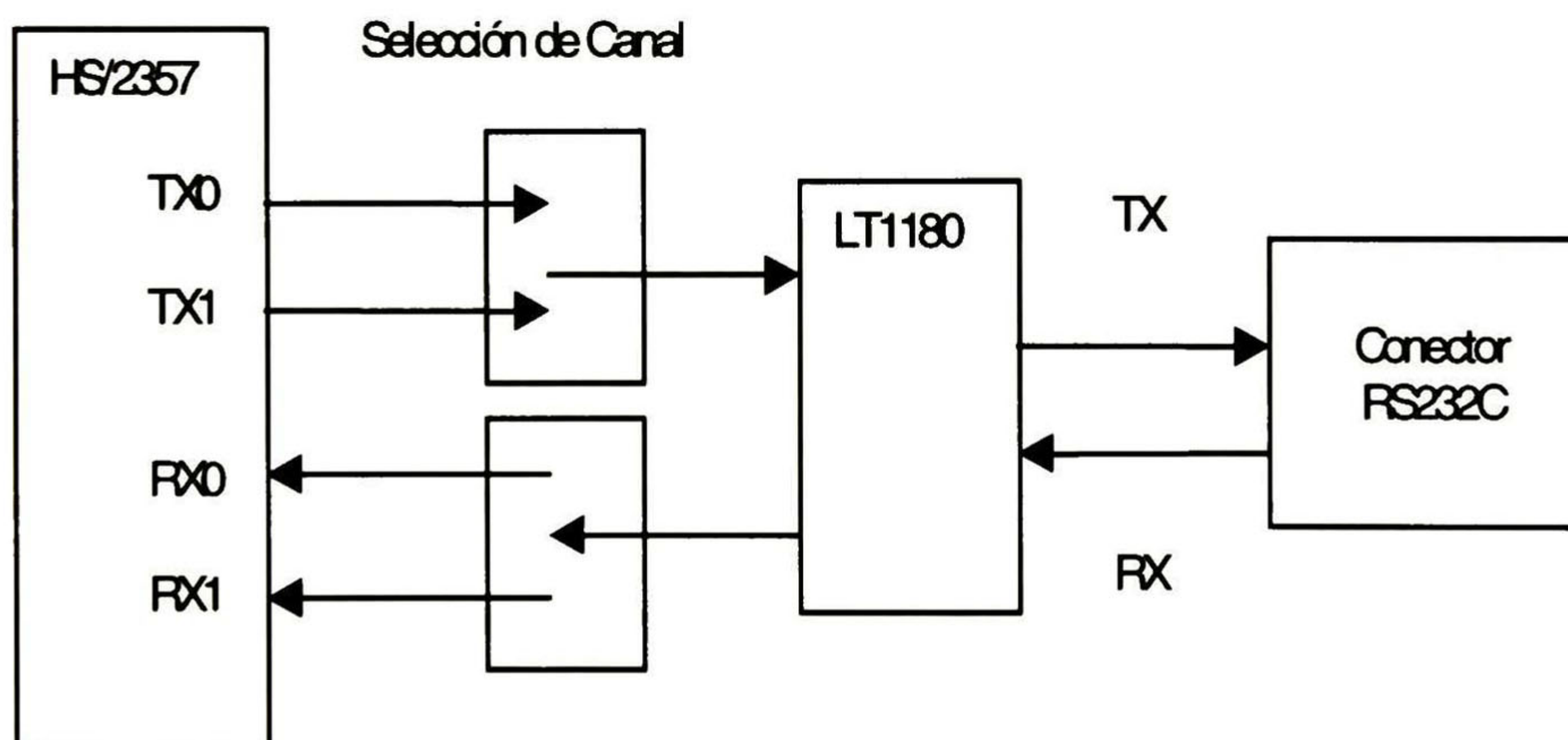


Figura 5-14 Implementación del protocolo RS232C.

6 Conclusiones y trabajo futuro

Dado que el objetivo del presente trabajo es crear una plataforma que ayude a reducir los tiempos de diseño del esquema de control de la nueva generación de productos manejadores de papel Hewlett Packard, a continuación se describen los objetivos específicos alcanzados durante la realización de este proyecto:

1. Se diseñó una arquitectura de hardware, basada en un microprocesador de alta eficiencia (Hitachi, familia H8S/235X) de 16 bits.
2. Se diseñaron y construyeron dos tarjetas de circuitos impresos denominadas: hardware del sistema y hardware de aplicación, que proveen al sistema la capacidad para implementar muy diversas tareas de propósito general, acceder al protocolo de comunicación CAN 2.0B (interfaz manejador de papel con la impresora), comunicarse con una PC por medio de un puerto serial, conocer el nivel lógico de sensores electrónicos y proporcionar las señales necesarias para manejar de manera eficiente motores de pasos y motores de corriente directa.
3. Se desarrolló, en lenguaje C, el firmware de control requerido por el sistema implementado, para constituir la interfaz entre el hardware y el software de aplicación diseñado. Su función es, utilizando los algoritmos desarrollados y proporcionados por los ingenieros de HP para ejercitar motores y leer sensores, genere el código necesario para arrancar y parar motores de corriente directa y motores de pasos, así como conocer el estado de los sensores, de acuerdo a las necesidades del usuario, proporcionadas a través de la interfaz gráfica.
4. Se programó el PLD EPM7064S de Altera, para que implemente la funcionalidad de generar una interrupción cuando detecte cambios en los niveles lógicos de los sensores conectados al hardware de aplicación.
5. Se diseñó e implementó un software de aplicación que, por medio de una interfaz gráfica, le proporciona al usuario la capacidad para configurar, operar y controlar los motores y sensores, conectados al sistema electromecánico del manejador. De esta forma se generan los movimientos específicos requeridos para transportar el papel a través del dispositivo.
6. Se conformó la arquitectura de hardware del esquema de control de la nueva generación de productos manejadores de papel que Hewlett Packard desarrolla.

La herramienta simplifica de manera eficiente y disminuye el tiempo de diseño e implementación del esquema de control para la nueva generación de dispositivos manejadores de papel Hewlett Packard.

Por medio de este proyecto, tuve la oportunidad de adquirir la experiencia y un mayor conocimiento del mercado, referente a la selección de partes electrónicas basadas en sus características técnicas, precio y disponibilidad, proceso que no siempre es fácil de realizar en un trabajo de tesis debido a la falta de recursos para realizar este tipo de estudios, pero que determina en gran medida la arquitectura final de un sistema implementado.

De igual forma me dio la oportunidad de formar parte del equipo de trabajo del área de investigación y desarrollo de la empresa Hewlett Packard, que refleja fielmente los estándares de calidad en el trabajo de una entidad industrial transnacional.

Trabajo futuro

El uso de esta herramienta representa una plataforma para el desarrollo de los manejadores de papel Hewlett Packard, debido a que cada nuevo producto se origina a partir de una arquitectura ya probada, que sólo debe ser ajustada a las necesidades de cada dispositivo en particular, lo que se traduce en una considerable reducción del tiempo requerido para el diseño y prueba de los prototipos, que conducen a un producto final.

El sistema representa una base para la investigación de los perfiles de movimiento de las trayectorias del papel a través de los manejadores, utilizados en los diversos productos de calidad que esta empresa desarrolla.

7 Referencias bibliográficas

1. S. Burrows, *A Laser Printer Book*, (1994).
<http://cp.ru/goods/Inform/prl/books/00-toc.html>
2. Hewlett Packard, *Hardware Drivers Documentation for Paper Handling Devices*, (Guadalajara), 1-58, (2000).
3. Hewlett Packard, *Printing and Digital Imaging*, (2000).
http://www.pandi.hp.com/pandi-db/dds_supp_access.show?p_catgy_id=18
4. Reliance Motion Control Inc, *DC Motors, Speed Controls and Servo Systems Handbook*, (Minnesota), 2-114, (1996).
5. Ericsson Components, *Stepper Motor Control Handbook*, (Estocolmo), 11-82, (1995).
6. J. Jacobson, *Safety of Distributed Machine Control Systems*, Swedish National Testing and Research Institute, (Boras), 15-46, (1996).
7. Hewlett Packard, *Paper Handling Firmware Development Standard*, (Guadalajara), 1-32, (2000).
8. Hitachi Semiconductors, *H8S/2357 Series, Hardware Manual*, (Tokyo), 1-1084, (1998).
9. L. Wolfhard, *CAN System Engineering*, (Stuttgart), 10-55, (1997).
10. R. Bosch, *CAN Specification version 2.0*, (1991).
<http://www.can.bosch.com>
11. Philips Semiconductors, *Interfacing a Stand Alone CAN controller*, (Amsterdam), 1-66, (2000).
12. Altera, *Max Plus II, Programmable Logic Development System*, (San Jose), 12-300, (1999).
13. J. Liberty, *Beginning Object-Oriented Analysis and Design: With C++*, Ed. Wrox Press Inc, 35-68, (1998).
14. H. Schildt, *Borland C++*, Ed. McGraw Hill, 55-255, (1997).
15. R. Kent, *Aprendiendo Borland C++ Builder 3 en 21 días*, Ed. Prentice Hall, 15-55, (1998).
16. A. Merrill, *Microprocessor Hardware/Software Interfacing*, Ed. B. Brey, (1984).
17. P. Kuliniewicz, *Windows API Guide*, (1999).
<http://www.vbapi.com/>
18. B. Brown, *RS232 Serial Communications*, (1995).
http://www.cit.ac.nz/smac/dc100www/dc_008.htm#rs232
19. Beyond Logic, *Serial/RS-232 Interfacing*, (2000).
<http://www.beyondlogic.org/index.html#SERIAL>
20. IAR Systems, *H8 C Compiler, a Programming Guide*, (San Francisco), 10-117, (1999).

El artículo "Herramienta para el desarrollo y pruebas de dispositivos manejadores de papel para impresoras Hewlett Packard", que se presenta a continuación, ha sido aceptado para su presentación en el congreso:

SOMI XV

CONGRESO DE INSTRUMENTACION

16 al 20 de Octubre de 2000

Guadalajara, Jal. Mexico

Herramienta para el desarrollo y pruebas de dispositivos manejadores de papel para impresoras Hewlett Packard

N. Castillo, D. Torres, H. Gutiérrez¹

Centro de Investigación de Estudios Avanzados del IPN unidad Guadalajara
Av. López Mateos Sur 590
Col. Tlajomulco de Zuñiga
Guadalajara, Jalisco
Tel. (013) 684-15-80

RESUMEN

El presente trabajo implementa una herramienta que simplifica de manera eficiente, el diseño del esquema de control de los dispositivos manejadores de papel que Hewlett Packard desarrolla. Por medio de dos tarjetas de circuitos impresos, el sistema proporciona una arquitectura de hardware genérica que sólo debe ser ajustada a la necesidad de operación de cada elemento, a fin de reducir los tiempos de desarrollo y prueba de los prototipos que conducen a un producto final. Debido a que la operación de un manejador está basado en un sistema electromecánico, encargado de mover el papel desde una impresora hacia un contenedor final seleccionado, la herramienta se complementa con un software de aplicación que permite al usuario ejercitar, de acuerdo a diferentes parámetros, los motores de corriente directa, motores a pasos y sensores que hacen posible el transporte del papel a través del dispositivo.

ABSTRACT

A useful development tool for Hewlett Packard paper handling devices control scheme is presented. The system provides a microcontroller based general architecture integrated by two PCB cards and a software application, used to improve their new products design and testing. It allows to reduce considerably the time intervals between prototypes and final devices. Due the paper handling devices operation is based in electromechanical systems control, an efficient graphical user interface were developed in order to configure, drive and test dc motors, steppers motors and sensors components.

1. INTRODUCCION

La concepción e implementación de un esquema de control, para cada nuevo dispositivo en el área de la electrónica, se lleva a cabo mediante un procedimiento que consta de varias etapas: concepción, análisis y diseño, implementación y pruebas. Este proceso se repite constantemente en el departamento de desarrollo de una empresa, aún cuando cada nuevo producto lanzado al mercado presente características similares, en estructura y operación, a su predecesor.

Una línea o familia de dispositivos la constituyen todos los elementos que cumplen con esta condición de generalidad, y que además pueden ser controlados a partir de una misma arquitectura. Aprovechando este concepto, y a fin de reducir los tiempos de desarrollo de cada producto, es posible crear una herramienta que sirva como base o plataforma para el diseño e implementación de los nuevos dispositivos de la familia.

Hoy en día, existen impresoras diseñadas para satisfacer altas necesidades de impresión, y que mediante el uso de manejadores de papel, proporcionan además la capacidad para acomodar, separar, distribuir y engrapar documentos. Un dispositivo manejador de papel tiene como función principal mover, con un alto grado de precisión y en coordinación con la impresora, hojas de papel.

¹ Hewlett Packard de México, S.A de C.V.

Dado que estos dispositivos están basados en la operación y control de un sistema electromecánico (motores, sensores), con características similares, pueden ser controlados de acuerdo a un mismo esquema.

A fin de proporcionar una plataforma que simplifique el desarrollo de los sistemas de control de la nueva familia de productos manejadores de papel Hewlett Packard, y dado que su función principal es mover con alto grado de precisión, en coordinación con una impresora, y a través de trayectorias definidas, las hojas de papel, el sistema debe contar con los elementos para:

1. Proveer las señales requeridas para generar los movimientos de los motores de corriente directa y de pasos que hacen posible el transporte de papel.
2. Detectar por medio de sensores el estado de una hoja de papel durante su transporte.
3. Almacenar y ejecutar códigos programables por medio de un sistema de control basado en un microprocesador.
4. Permitir la comunicación del manejador con otros dispositivos físicos, empleando un protocolo estándar.
5. Comunicarse con una PC.
6. Interactuar con el usuario a través de un programa que se ejecute en una PC.

El objetivo es diseñar e implementar una herramienta que proporcione a ingenieros los elementos de configuración y control necesarios para implementar en períodos cortos de tiempo, la funcionalidad de los prototipos que conducen a un producto final de alta calidad.

Definiciones

Impresora. Dispositivo que acepta texto y gráficos de una computadora y transfiere la información, usualmente, sobre hojas de papel de tamaño estándar.

Manejador de papel. Dispositivo que permite el manejo del papel a la salida de una impresora y que proporciona la capacidad para acomodar, separar, distribuir y engrapar, de acuerdo a las necesidades de un usuario, documentos impresos [1].

CAN 2.0B. Protocolo de comunicación serial utilizado en procesos de control y automatización donde la red de comunicación se establece a través de microprocesadores. Físicamente se constituye por un bus de dos cables que tienen la característica de establecer a través de ellos una comunicación bidireccional (half duplex) a muy alta velocidad. Entre sus principales ventajas se encuentran robustez, confiabilidad y aceptación de la industria, creando circuitos dedicados a este protocolo. Teóricamente, puede enlazar hasta 2032 dispositivos en una misma red, sin embargo, dadas las limitaciones prácticas del hardware, solo puede soportar hasta 110 nodos. El bus de CAN 2.0B ofrece enlaces de comunicación hasta de 1Mbit/s, lo que permite el control de sistemas en tiempo real. Además, dada su baja probabilidad de error, y capacidad de detección, representa un sistema de alta confiabilidad para trabajar en ambientes con niveles altos de ruido [2].

2. MANEJADORES DE PAPEL HEWLETT PACKARD

El contenedor de salida de papel en una impresora determina el formato de entrega de los documentos que el dispositivo ofrece al usuario. La figura 1 muestra las posibles rutas que ofrece una impresora Hewlett Packard como destino final de las hojas de papel impresas.

Los contenedores de papel con impresión hacia arriba o hacia abajo, constituyen el formato de salida normal del dispositivo. Sin embargo, existe la posibilidad de agregar un elemento "manejador de papel" en lugar del contenedor de impresión hacia arriba, para obtener otro resultado.

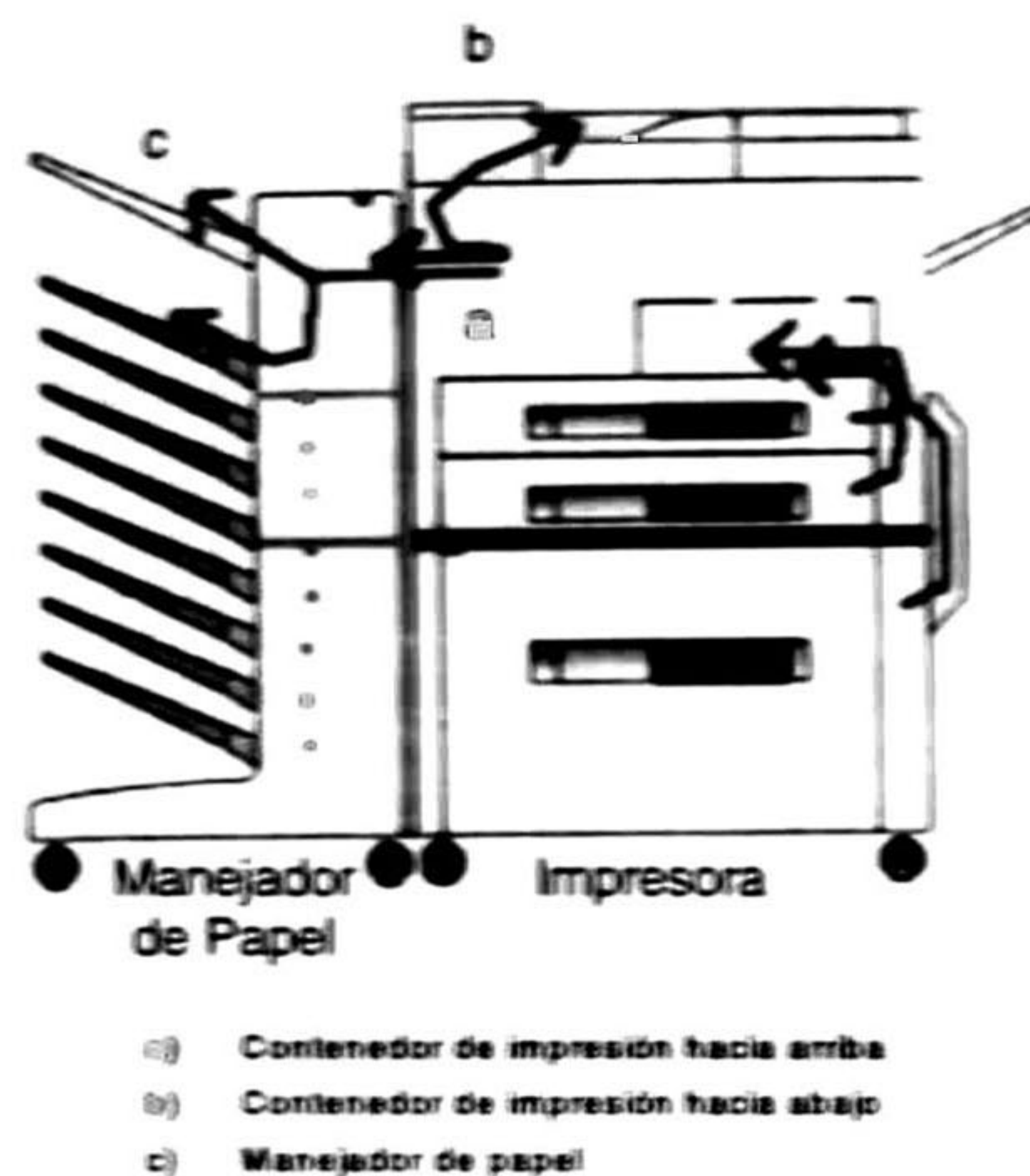


Figura 1. Salida hacia los contenedores de papel, en una impresora Hewlett Packard.

Un manejador de papel Hewlett Packard ofrece la oportunidad de apilar, separar, distribuir e incluso engrapar trabajos con el propósito de hacer más eficiente el uso de impresoras de gran capacidad. Así, el objetivo principal del manejador es controlar el movimiento, con alto grado de precisión, de hojas de papel desde una impresora hacia un contenedor de salida propio, con la idea de proporcionar un acabado y distribución final diferente al usual o clásico [1].

2.1. Tipo de papel

El tipo de papel utilizado por los manejadores Hewlett Packard, y en general por las impresoras, debe ser de buena calidad y especial para trabajos de fotocopiado, a fin de que su tamaño (variación +/- 2 mm) y estado general (libre de polvo e imperfecciones) sea aceptable. De igual forma, el papel demasiado delgado no es recomendable, ya que su uso ocasiona frecuentes problemas como hojas atoradas o atascadas, por lo que la mayoría de los dispositivos son diseñados para trabajar con papel de 80 a 90 gmc (gramos por metro cuadrado) [3].

Algunas veces es deseable la impresión de documentos sobre diferentes tipos de papel. El uso de hojas plastificadas, membretadas o recicladas, es permitido por la mayor parte de impresoras y manejadores, con la aclaración de que los errores de operación pueden incrementarse debido a la dificultad de su manejo. Otros tipos de materiales que un manejador soporta son tarjetas, sobres o etiquetas y en menor escala transparencias, cuyo uso se restringe a dispositivos especialmente diseñados para ello.

2.2. Movimiento de papel en un manejador

El movimiento de papel en un manejador Hewlett Packard, se describe por un conjunto de trayectorias $T = \{T_1, T_2, \dots, T_n\}$ que determinan las posibles rutas que las hojas de papel pueden seguir desde su salida de la impresora, hasta el contenedor seleccionado del dispositivo. A cada una de estas trayectorias de papel está asociado un perfil de movimiento específico M_i , necesario para transportar la hoja a través del manejador. Cada M_i es definido por un conjunto de movimientos simples m_k que pueden formar parte de cualquier trayectoria, $M_i = \{m_1, m_2, \dots, m_k\}$ y donde cada m_k a su vez, es producto de la acción asociada f_k de un motor con uno o varios sensores, $m_k = f_k \{e_1, e_2, \dots, e_n\}$.

Dados los m_k requeridos por estos dispositivos, movimientos con carga mínima, de alta precisión y control, los elementos e_n representan en su mayoría sensores detectores de posición que regulan

el paso de las hojas a través del dispositivo; motores de pasos, encargados de girar rodillos basándose en diferentes parámetros como velocidad y posición; y en menor proporción, motores de corriente directa que cumplen con la funcionalidad de mover los elementos mecánicos del sistema (apiladores de papel, engrapadora).

Uno de los usos de motores en un manejador de papel se muestra en la figura 2. El motor está acoplado generalmente a un sistema de engranes o bandas que cambian la relación de velocidad, para trabajar en un punto de operación deseado y bajo los parámetros de operación óptimos del motor, de acuerdo a los datos que el fabricante proporciona. Estos engranes a su vez, transmiten el movimiento a una flecha, en donde se encuentran los rodillos transportadores.

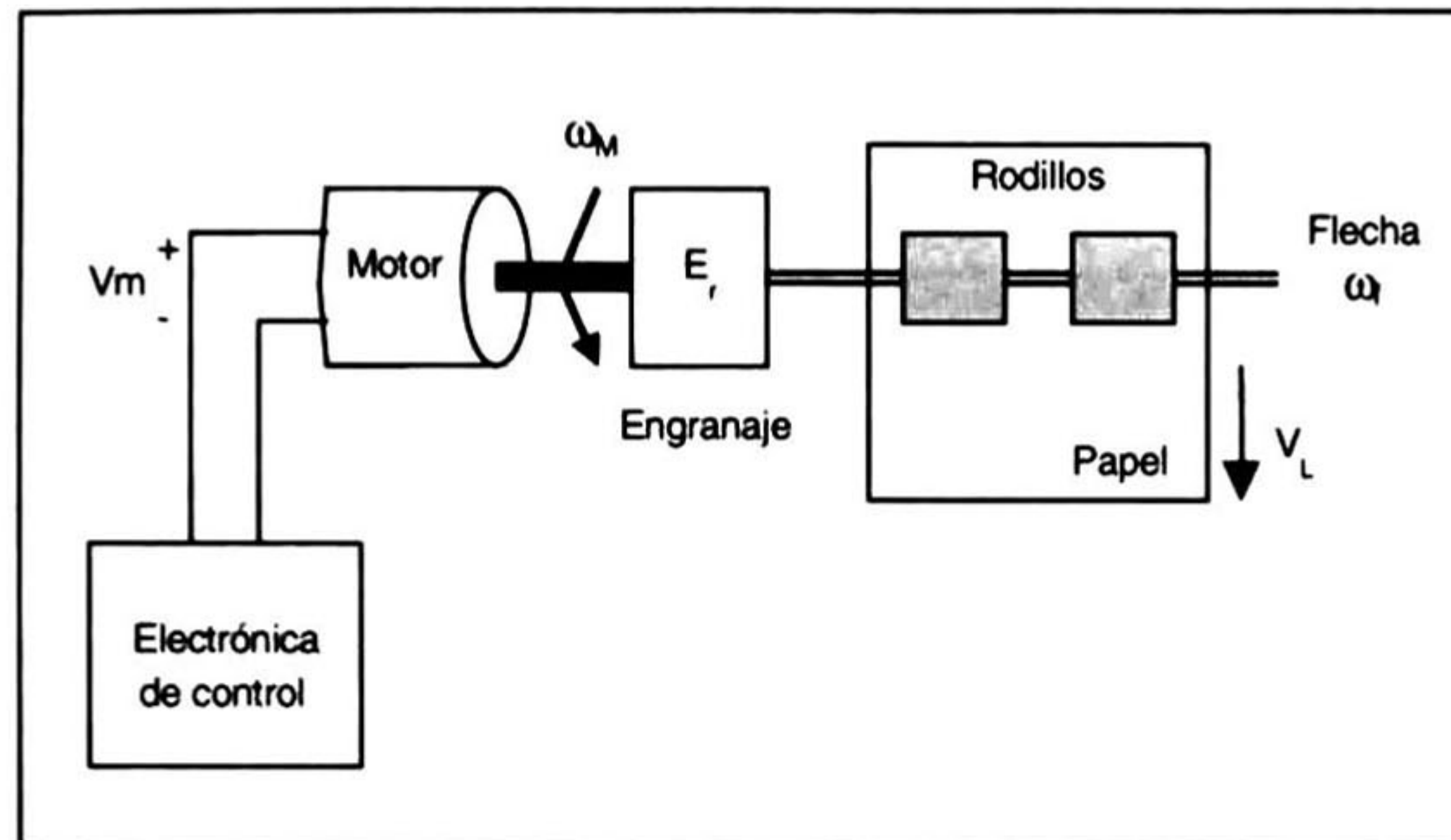


Figura 2. Movimiento de papel producido por un motor.

Los rodillos proveen una fuerza proporcional a la atracción entre ambos elementos y el coeficiente de fricción del material con el que están elaborados. Bajo condiciones óptimas de operación, el movimiento angular de la flecha es transmitido en su totalidad al papel, para que éste experimente un movimiento lineal. Así, la velocidad y aceleración a los que el papel está sometido son directamente controlados por el motor.

Motores de corriente directa.

Para el caso de un motor de corriente directa, y de acuerdo al esquema de la figura 2, la velocidad lineal (V_L) está representada por la ecuación $V_L = P\omega_f$, donde $P = 2\pi R_r$ constituye un factor de atracción ejercido entre los rodillos de radio R_r sobre las hojas de papel, cuyo valor es multiplicado por la velocidad angular final de la flecha ω_f . Este parámetro depende de la relación de engranaje E_r y la velocidad angular del motor ω_M ($\omega_f = E_r\omega_M$) [4].

Dado que ω_M a su vez, es el producto del voltaje aplicado al motor y su constante de fabricación ($\omega_M = V_M K_M$), la velocidad lineal puede ser descrita de acuerdo a la siguiente ecuación:

$$V_L = 2\pi R_r E_r V_M K_M$$

El análisis anterior muestra la relación entre el voltaje aplicado al motor y la velocidad lineal final proporcionada. Esta velocidad es proporcional al movimiento lineal (M_L) del papel ejercido por el motor, al introducir un factor de tiempo, representado por la función escalón $\mu(t-t_0)$; así, el papel se mueve finalmente de acuerdo a la relación:

$$M_L = V_L \mu(t-t_0)$$

En este esquema, la electrónica de control aplica y regula el voltaje para obtener una velocidad lineal en el papel según se requiera. La dirección del movimiento proporcionado por un motor de corriente directa está dada por la polaridad del voltaje aplicado, ya sea en uno u otro sentido.

Motor de pasos.

La utilización de un motor de pasos en los manejadores de papel es similar al uso de motores de corriente directa. Sin embargo, para este caso, la velocidad angular del rotor es proporcional a la frecuencia con que se le introducen pulsos eléctricos a las bobinas del motor. Al avance en radianes del rotor que es provocado por un pulso en las bobinas del estator se le denomina 'paso'. De acuerdo a la técnica de construcción del motor, el número de pasos por revolución del rotor varía. El resto del análisis de la velocidad y movimiento es similar al mostrado con los motores de corriente directa [5].

Dado que el voltaje V_M aplicado al motor, tal y como se menciona anteriormente, es ahora sustituido por una fuente de frecuencia f , la velocidad lineal del papel utilizando un motor a pasos esta dada por:

$$V_L = 2\pi R_r E_r f K_M$$

De igual forma, su factor de fabricación puede desglosarse de acuerdo a sus características de construcción por medio de la relación $K_M = 2\pi/n$, donde n representa el número de pasos por revolución en el rotor, la velocidad lineal del papel proporcionado por este dispositivo es:

$$V_L = 4\pi^2 R_r E_r f / n$$

La gran ventaja que presenta un motor a pasos frente a un motor de corriente directa es un control de la velocidad más preciso, sin la necesidad de un sistema de retroalimentación. Además, este tipo de dispositivo permite mantener un torque o fuerza constante aún cuando no se esté moviendo, con el propósito de fijar la posición de una hoja de papel en un punto dado, sin posibilidades de que se mueva o desajuste de su posición fácilmente.

En un esquema controlado por un motor a pasos, la dirección del movimiento está dada por la secuencia con la que los pulsos son introducidos en las bobinas del motor.

Perfiles de movimiento, velocidad y movimiento.

La figura 3 muestra una gráfica de la relación entre aceleración, velocidad y movimiento que generalmente se transmiten a una hoja de papel en un manejador. De acuerdo a la figura, los perfiles de aceleración y desaceleración son constantes, pero en la realidad pueden tener diferentes formas. Este parámetro proporciona los niveles de velocidad lineal necesarios para generar el movimiento que hace posible el transporte de papel a través del dispositivo. Estas señales son de gran utilidad durante el desarrollo de un manejador de papel.

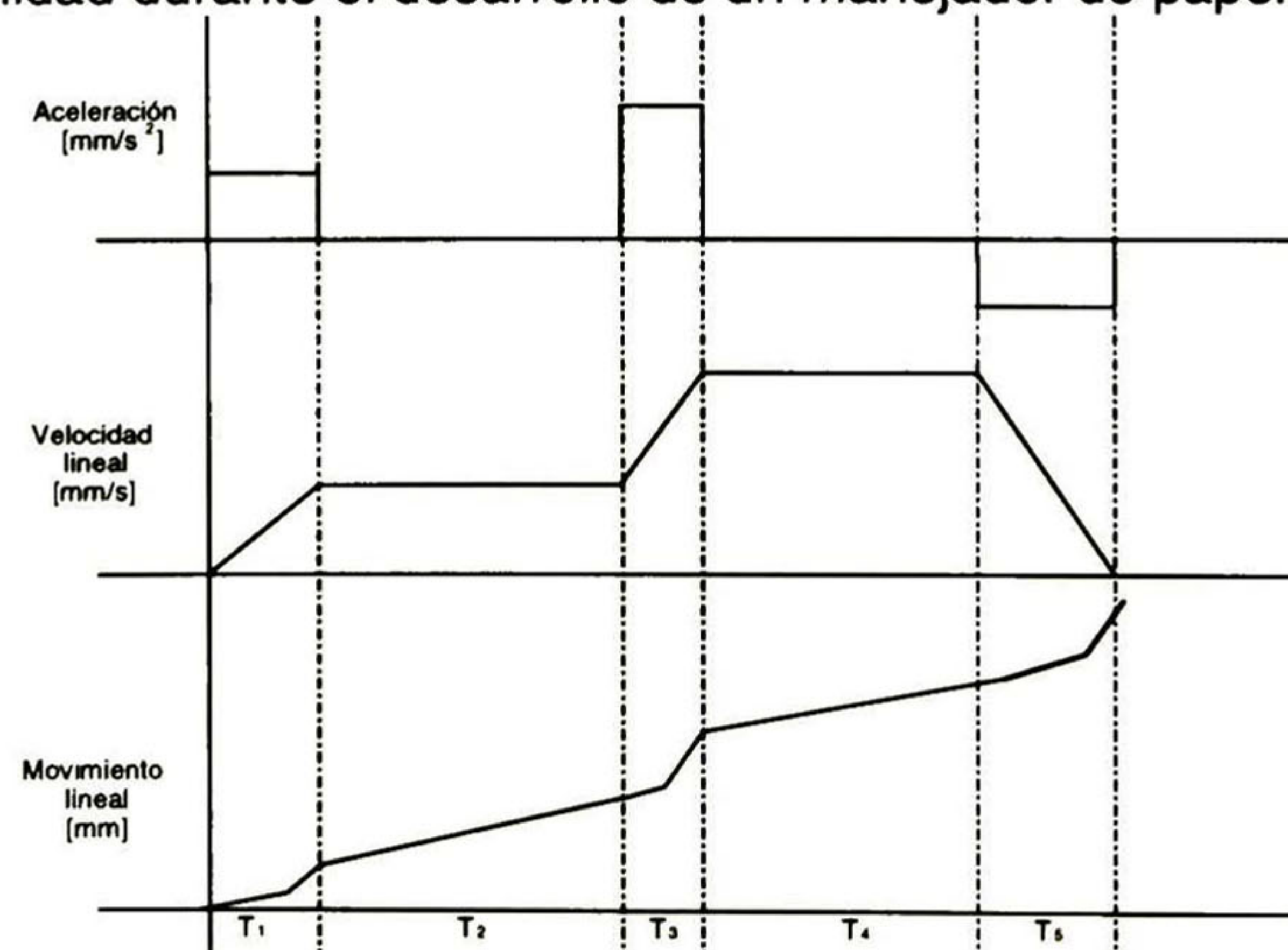


Figura 3. Relación entre aceleración, velocidad y movimiento de papel en un manejador.

Cuando deseamos analizar los tiempos y posiciones del papel durante su transporte dentro del manejador, los relativos a la aceleración y desaceleración se desprecian debido a que son mucho menores que los de transporte, $[T_1, T_3, T_5] \ll [T_2, T_4]$.

3. ARQUITECTURA

Gran parte de los productos o dispositivos en el área electrónica, están implementados basándose en una arquitectura definida por un sistema de control programable, constituido por un microprocesador, componentes periféricos al procesador, y ciertos módulos de aplicación [6].

Para el caso de los manejadores de papel, dispositivos basados en un sistema electromecánico, cuya operación depende de motores y ciertas señales de monitoreo (sensores), la reutilización de su arquitectura de control y sus manejadores de hardware es posible, ya que son diseñados en base a subsistemas electrónicos bien definidos, operados por un mismo esquema de control, y que sólo es necesario adaptar a las necesidades específicas de cada producto.

La plataforma física común para el desarrollo de los manejadores, se complementa con el diseño de una aplicación de software que permita la interacción, por medio de una interfaz gráfica, del sistema con el usuario. El objetivo de este programa es que el diseñador o ingeniero mecánico pueda configurar y probar motores y sensores de un dispositivo, de acuerdo a su necesidad.

La arquitectura general de la herramienta (figura 4) se compone de tres bloques funcionales básicos: *hardware*, *firmware de control* y *software de aplicación*.

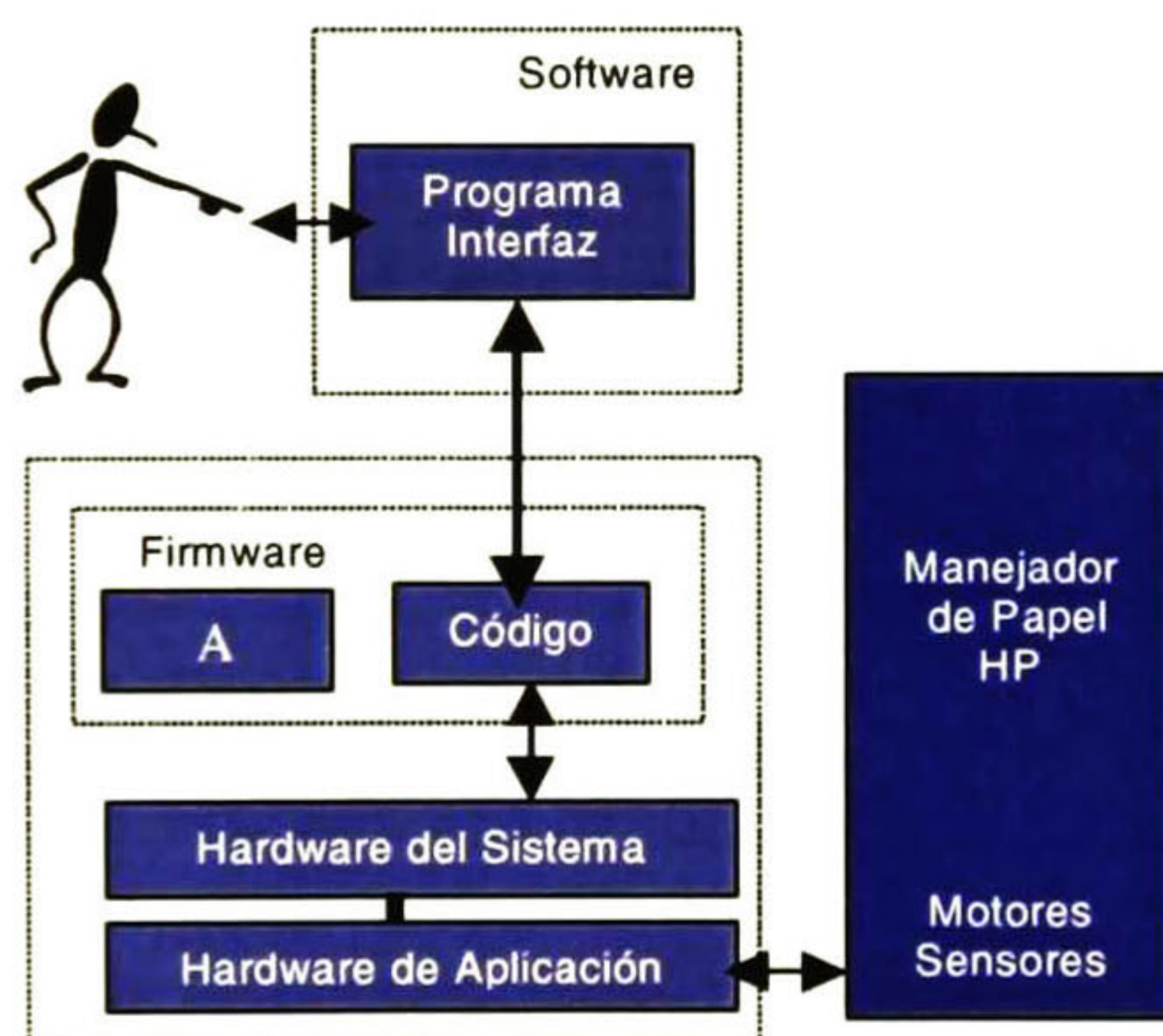


Figura 4. Arquitectura general para la herramienta de desarrollo.

3.1. Hardware

Proporciona los componentes físicos que hacen posible el movimiento de los motores y la lectura de los sensores que conforman un manejador de papel. Se compone de varios elementos como un microprocesador, circuitos periféricos al procesador (memorias, controladores), manejadores de motores (también conocidos como "drivers"), electrónica para lectura de sensores, circuitos de comunicación, entre otros. Está integrado por dos tarjetas de circuitos impresos: el hardware del sistema y el hardware de aplicación.

Hardware del sistema. Elemento central de la plataforma física, que constituye el sistema mínimo de control de la herramienta, y proporciona, sobre la base de un microprocesador programable, la capacidad para implementar muy diversas tareas de propósito general. Se compone de un microprocesador y de elementos periféricos al procesador.

Hardware de aplicación. Implementa una función específica, en este caso, la generación de las señales físicas que hacen posible el movimiento de los motores y la lectura de los sensores del manejador de papel. Su diseño está determinado por la arquitectura del hardware del sistema, con quién se comunica a través de un conector.

3.2. Firmware de control

Representa la interfaz lógica entre el hardware y el software de aplicación de la herramienta. Se compone de un conjunto de instrucciones programables dirigidas al microprocesador, que hacen posible que éste genere las señales requeridas por los circuitos electrónicos periféricos, para llevar a cabo las operaciones deseadas en el sistema [7].

Se divide en dos componentes básicos: el primero proporciona un conjunto $A = \{ A_0, A_1, \dots, A_n \}$ de algoritmos para ejercitar motores de corriente directa y de pasos, donde A_0, A_1, A_n implementan la funcionalidad f_k necesaria para generar los movimientos simples m_k que definen cada uno de los perfiles M_i de movimiento, que se encargan de transportar las hojas de papel sobre el conjunto T de trayectorias en un manejador. El segundo componente consiste en un código que accede al conjunto A para mover motores y, a su vez, se comunica al exterior a través de un puerto serial estándar a una velocidad de 38400 bps. La información que este módulo maneja es capturada e interpretada por el tercer bloque funcional, el software de aplicación de usuario.

3.3. Software de aplicación

El software de aplicación de usuario reside en una PC y, además de comunicarse por el puerto serial con el firmware, provee una interfaz gráfica, desarrollado en objetos, empleando como herramienta C++ Builder 3.0, para que un usuario interactúe con el sistema de acuerdo a sus propias necesidades. El programa que se ejecuta sobre el sistema operativo Windows 95 o Windows NT, permite configurar los parámetros necesarios para controlar el funcionamiento de los componentes electromecánicos de un manejador. La información obtenida por el programa es codificada y transmitida hacia el firmware de control para que ejecute la operación requerida.

3.4. Interfaces de comunicación

Los dispositivos manejadores de papel, requieren llevar a cabo el control de un determinado número de procesos: sensado de variables, configuración de elementos, activación y desactivación de motores, relacionados con el movimiento de papel en el producto. Sin embargo, una de las principales características de este tipo de dispositivos es que no se encuentran aislados, sino que tiene una interacción directa con impresoras y otros elementos auxiliares para el manejo de papel.

Debido a esto, cada manejador requiere establecer un tipo de comunicación que le permita conocer las condiciones externas a él mismo, sobre las cuales trabaja. El protocolo de comunicación CAN versión 2.0B ha sido adoptado por Hewlett Packard para definir la interfaz entre los manejadores de papel y las impresoras que desarrolla.

3.4.1. Protocolo CAN 2.0B

Las principales características del protocolo CAN 2.0B, definido por el estándar ISO 11898, son:

- Su capa física utiliza la transmisión diferencial por medio de un par de cables a muy alta velocidad (1Mbit/s).
- Control de acceso al bus eficiente, implementado por una secuencia de bits arbitraria no destructiva.

- Para el intercambio de información utiliza mensajes pequeños (ocho bytes de datos como máximo), protegidos por el método de verificación CRC (Cyclic Redundancy Check).
- No existe una dirección específica en los mensajes, en su lugar, cada uno contiene un valor numérico que controla su prioridad en el bus.
- Capacidad para manejar errores.

Los mensajes de datos transmitidos por cualquiera de los nodos conectados al bus CAN no contienen direcciones y tampoco la identidad del nodo que lo transmite o el nodo destino. En su lugar, el contenido del mensaje es etiquetado por un identificador que es único a través de la red. Una vez liberado el mensaje, todos los nodos en la red reciben el mensaje y cada uno de ellos lleva a cabo un proceso de prueba con el identificador, para determinar si el mensaje, y su contenido, es relevante para cada uno en particular. Si el mensaje es relevante, es procesado, de lo contrario se ignora.

El identificador único, también determina la prioridad del mensaje. Mientras más bajo sea el valor numérico del identificador, mayor es su prioridad. Esto permite el control de tráfico del bus, en caso de que dos o más nodos compitan por el acceso al mismo tiempo. Este control de acceso al bus garantiza que el mensaje con la mayor prioridad, aparezca como el único mensaje que está siendo transmitido. Los mensajes de menor prioridad, mientras tanto, son automáticamente retransmitidos en el siguiente ciclo del bus, o en un ciclo subsecuente, si existen todavía otros mensajes de mayor prioridad esperando a ser transmitidos [8].

Enseguida se describe, en base la topología de comunicación de una impresora y un manejador de papel Hewlett Packard (figura 5), el principio de funcionamiento básico del protocolo CAN:

1. El bus CAN contiene tres nodos: el nodo 1 es el controlador principal de la impresora, el 2 lo representa un componente auxiliar encargado de la entrada de papel a la impresora, mientras que el nodo 3 es un manejador conectado a la salida del dispositivo de impresión.
2. El controlador en el nodo 1 transmite un mensaje, que indica que una hoja ya impresa, se dispone a salir.
3. El mensaje y su identificador, pasan al circuito integrado controlador de CAN 2.0B del nodo 1. Este circuito construye el mensaje, de acuerdo al protocolo y lo transmite sobre el bus.
4. El mensaje transmitido por el nodo 1 convierte a los nodos 2 y 3 en receptores.
5. Enseguida, cada uno de estos nodos lleva a cabo una evaluación de la información contenida en el mensaje, para determinar si es relevante para su elemento de control o no.
6. Si el mensaje no es relevante para un nodo, éste lo ignora (por ejemplo, para el nodo 2, ya que este sólo controla la entrada de papel).
7. Por el contrario, si la información si tiene relevancia, el mensaje es aceptado y procesado, en este caso por el nodo 2 que inmediatamente activa el dispositivo para recibir la hoja de papel.

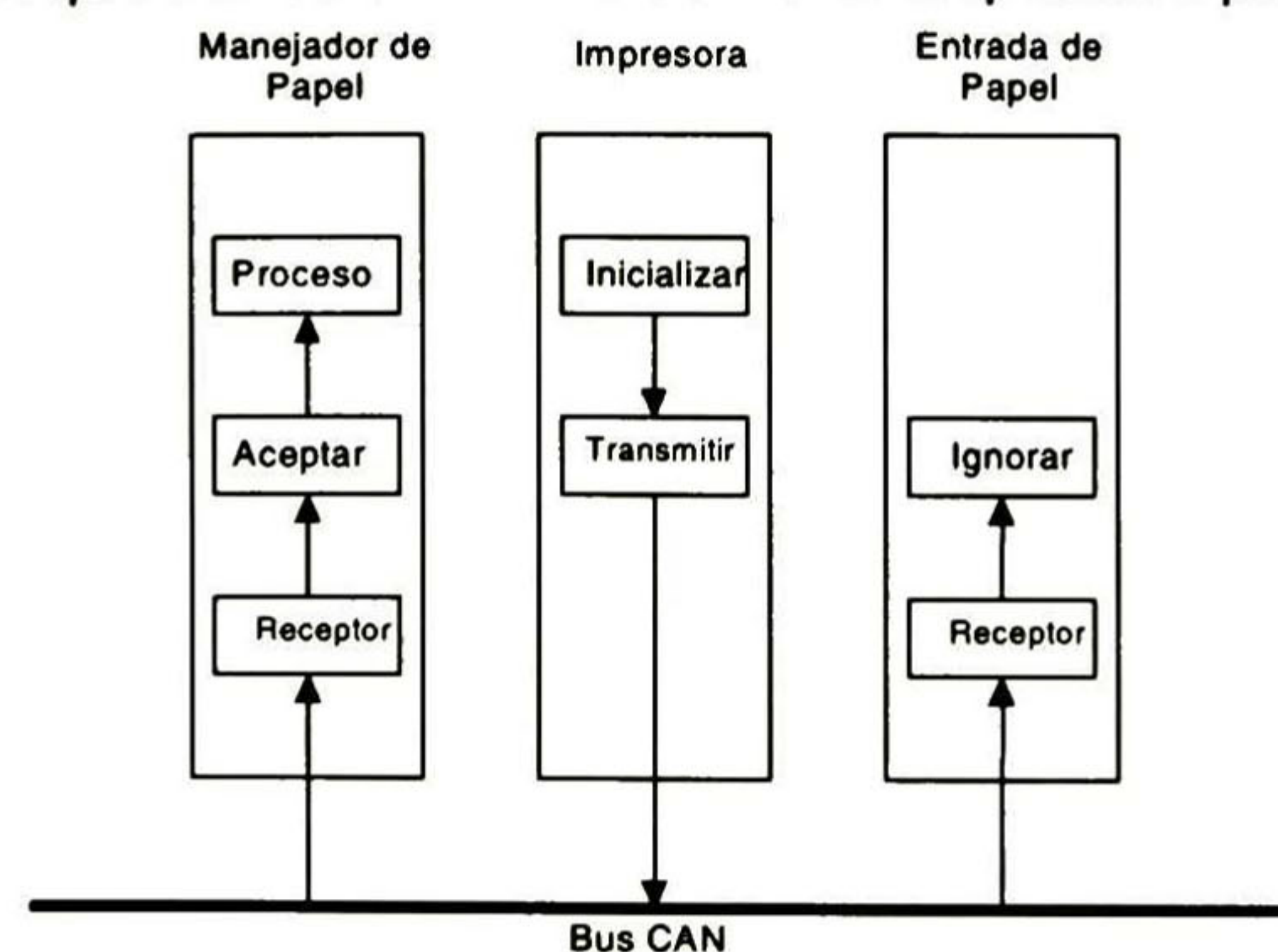


Figura 5. Transmisión de mensajes entre una impresora y un manejador por medio de CAN.

La herramienta debe integrar a su esquema, los elementos necesarios para cumplir con las especificaciones de este protocolo estándar, a fin de cubrir con cada aspecto relacionado a los dispositivos que maneja. De igual forma, y de acuerdo a su arquitectura debe soportar el protocolo RS232C necesario para comunicar el software de aplicación ya descrito, con el firmware de control.

4. HARDWARE

De acuerdo a la arquitectura definida para la herramienta, este módulo se divide física y lógicamente en dos tarjetas de circuitos impresos denominadas: hardware del sistema y hardware de aplicación.

Hardware del sistema. El diseño del hardware del sistema (figura 6), está definido por la selección del microprocesador y sus componentes periféricos, su requerimiento de funcionalidad y su interfaz con otros sistemas. Se conforma de los siguientes elementos:

- Microprocesador H8S/2357 (familia H8S/2000 de Hitachi), de alta eficiencia, basado en una arquitectura de 16 bits [9].
- Memoria Flash de 2 Mbits, organizada como 128 Kbytes x 16 bits.
- Memoria RAM estática (SRAM) de 512 Kbits, organizada como 32 Kbytes x 16 bits.
- Memoria no volátil (EEPROM serial) de 512 bytes.
- Controlador del protocolo de comunicación CAN 2.0B.
- Canal de comunicación serial.

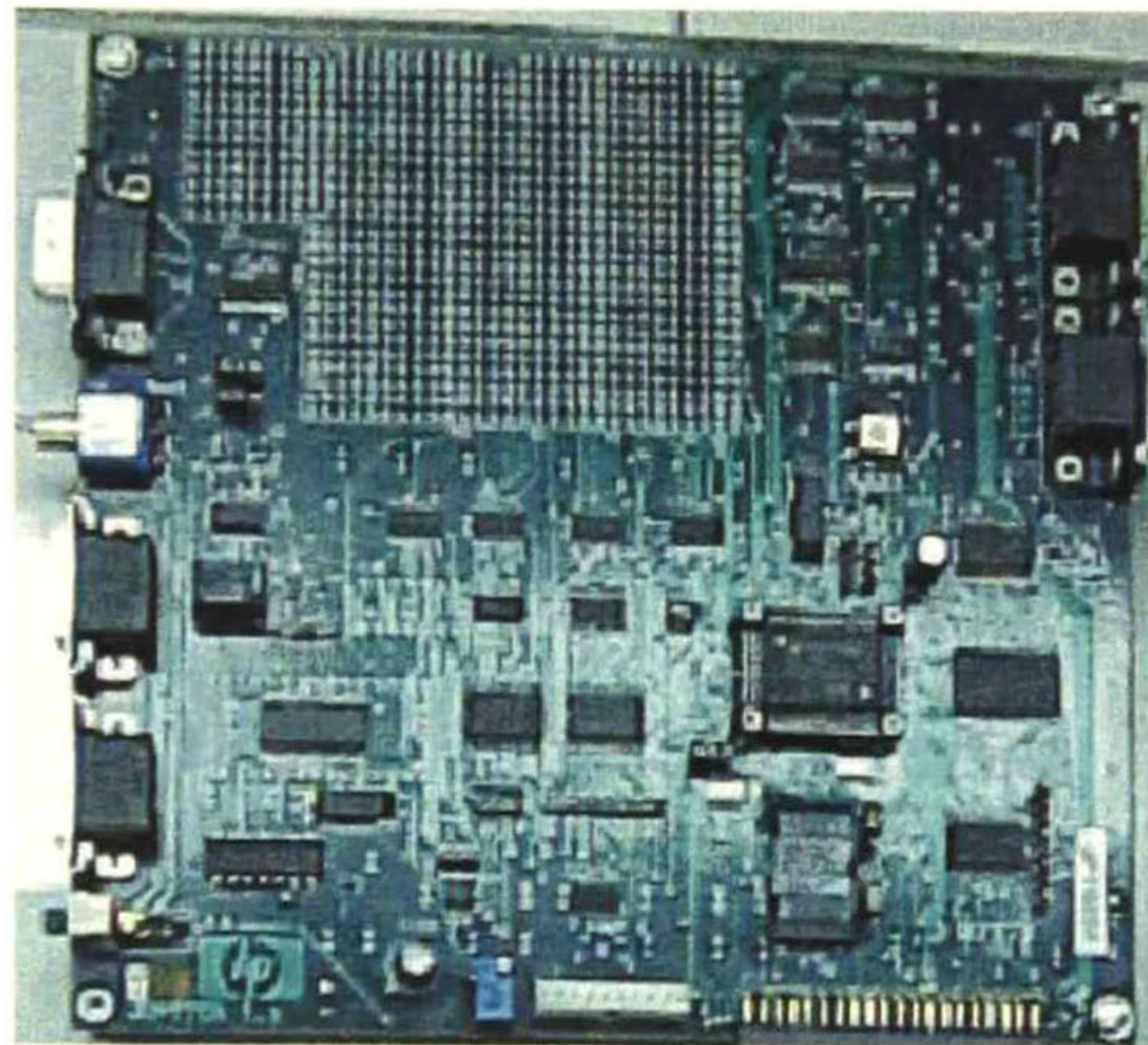


Figura 6. Hardware del sistema.

Su implementación proporciona a la herramienta de la capacidad para:

- Realizar muy diversas tareas de propósito general.
- Accesar al protocolo de comunicación CAN 2.0B, que habilita la interfaz entre el manejador de papel y una impresora.
- Comunicarse con una PC por medio de un puerto serial.

Hardware de aplicación. El hardware de aplicación (figura 7), que se une física y lógicamente al hardware del sistema, a través de un conector, es un tipo de tarjeta universal que contiene los elementos necesarios para manejar motores de pasos y motores de corriente directa. Además, tiene la capacidad para leer el estado de un cierto número de sensores y generar una interrupción cuando alguno de ellos cambie de activo a inactivo o viceversa. Esta formado por:

- Conector que permite la conexión con el hardware del sistema.
- Electrónica requerida para la lectura de 16 sensores.
- Lógica de interrupción de sensores, basada en el dispositivo lógico programable (PLD) EPM7064S de Altera [10].
- Manejadores de motor de propósito general configurables, con capacidad para controlar 14 motores de pasos y 31 motores de corriente directa.

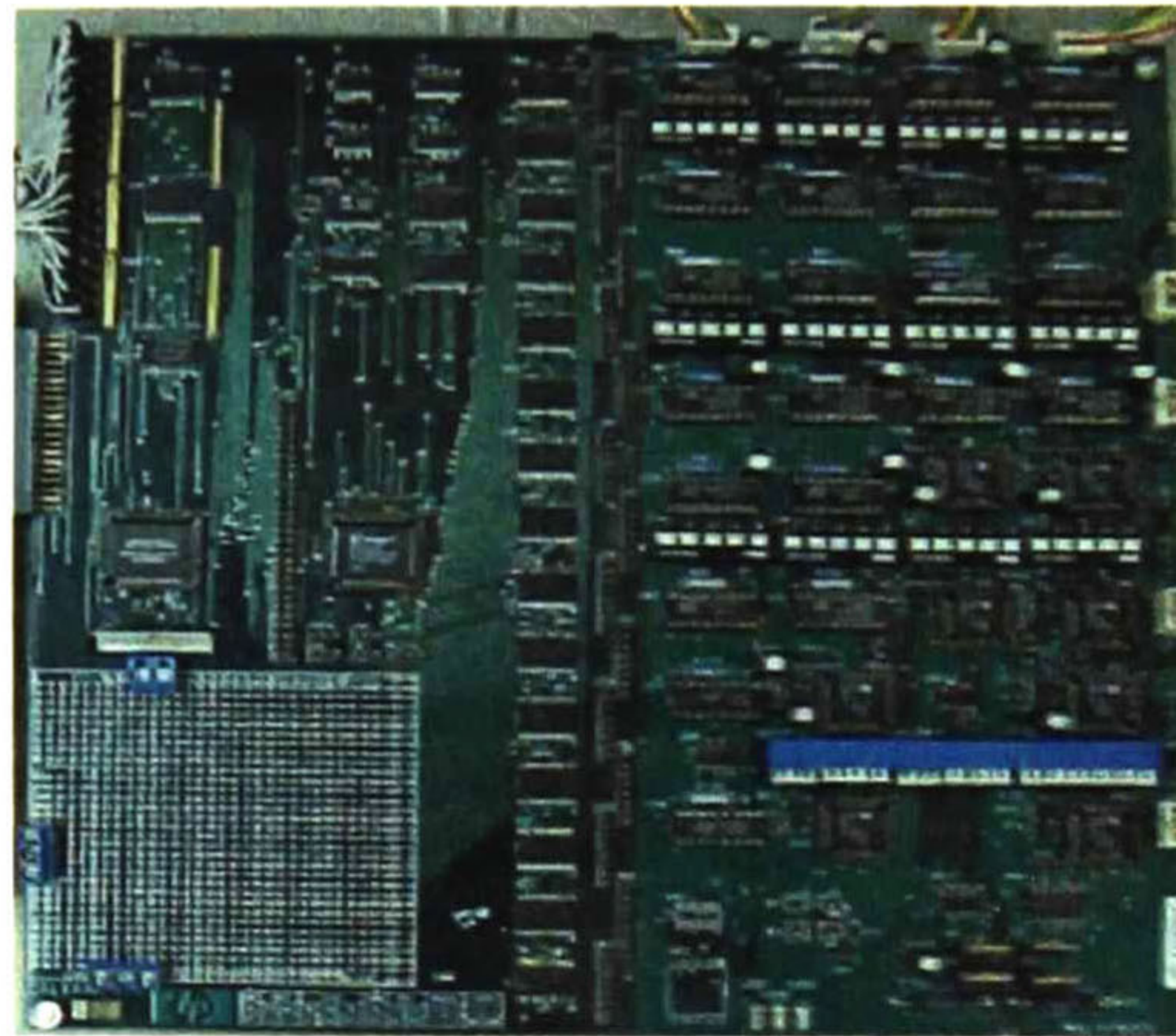


Figura 7. Hardware de aplicación.

5. SOFTWARE

El software de aplicación, representa el medio para que el usuario interactúe con la plataforma de desarrollo a fin de probar el sistema electromecánico del manejador de papel. A través de una interfaz gráfica interactiva que le permite configurar y manejar motores y sensores, un ingeniero es capaz de encontrar los parámetros de operación óptimos de los dispositivos.

Dada la funcionalidad requerida por el programa su implementación, desarrollada en objetos, empleando como herramienta C++ Builder 3.0, se basa en los elementos que constituyen el diagrama de la figura 8 [11].

Control. Constituye el elemento central del sistema. Es implementado por medio de un objeto único que ejecuta acciones y supervisa el funcionamiento de todo el sistema. Las principales tareas que ejecuta son:

- a) Inicializar el sistema, mediante la creación de objetos funcionales que reflejan el hardware de la plataforma física que la herramienta soporta.
- b) Implementar la interfaz con el usuario.
- c) Crear, actualizar y controlar objetos que representan el elemento activo del sistema (motores, sensores), de acuerdo a la interacción con el usuario y la operación del firmware de control.
- d) Interpretar y codificar los datos de operación del sistema, para ser transmitido y ejecutado por el firmware de control. De igual forma se encarga de decodificar la información proveniente de este módulo. Contiene el manejador del puerto serial para la transmisión y recepción de datos.

Objetos funcionales. Se dividen en dos partes: una contiene toda la información acerca de los elementos de software que reflejan el hardware del sistema (manejadores de motor, electrónica para lectura de sensores), mientras que la segunda describe los motores y sensores que dinámicamente se activan y desactivan de acuerdo a la operación del programa.

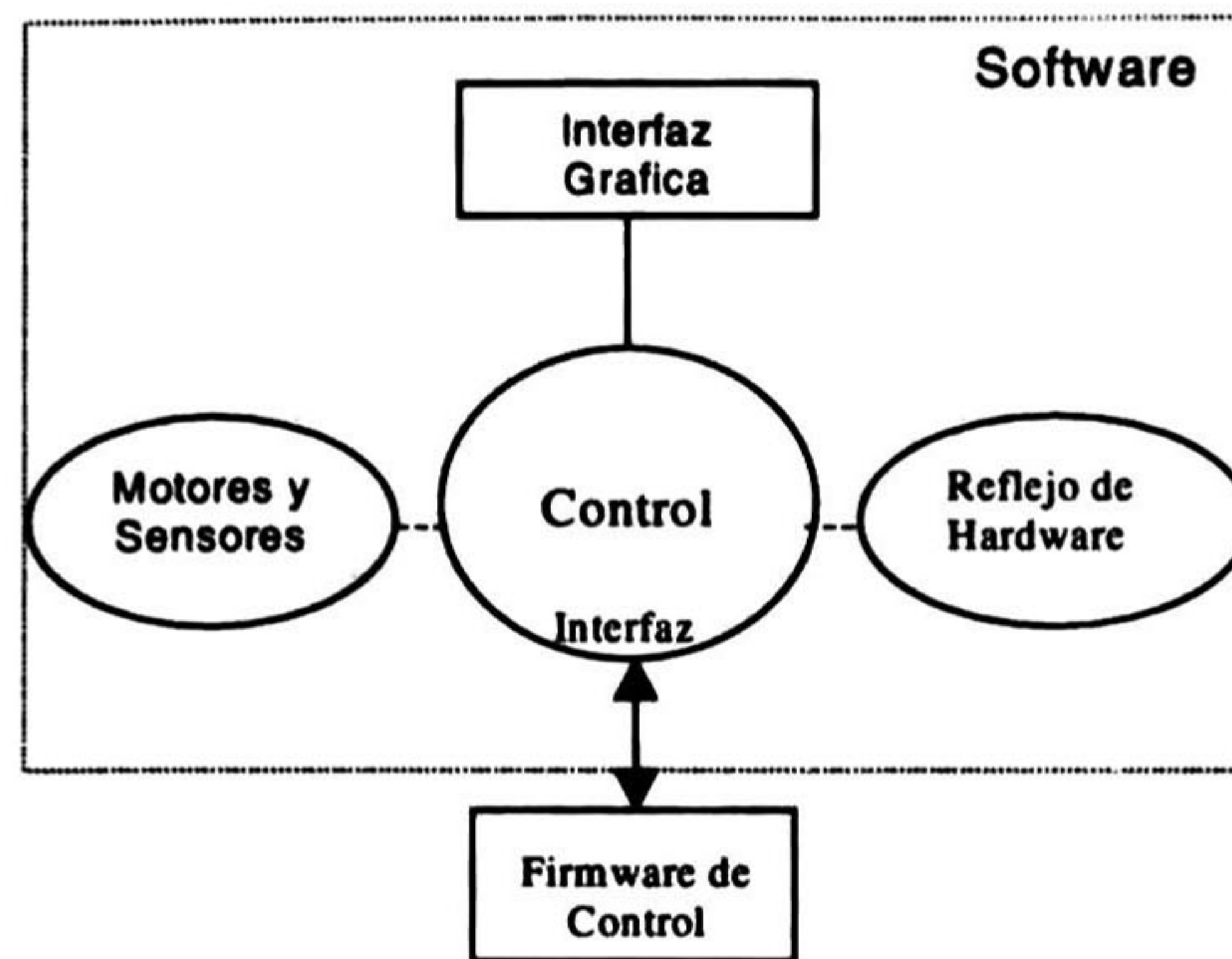


Figura 8. Arquitectura del diseño e implementación del software de aplicación.

Interfaz gráfica. Interfaz amigable y fácil de usar (figura 9) que permite al usuario configurar la herramienta para ejercitar motores de corriente directa y de pasos, así como conocer el estado de los sensores [12]. El programa ofrece al usuario la capacidad para manejar: *motores de corriente directa*, donde cada motor puede ser operado independientemente de acuerdo a parámetros como tipo de control, dirección y nivel de corriente; *motores de pasos*, con la capacidad para trabajar en base a una dirección, un nivel de corriente y diferentes niveles de velocidad (definidos por rampas de aceleración y desaceleración); y *sensores*, por medio de la detección constante de sus niveles lógicos.

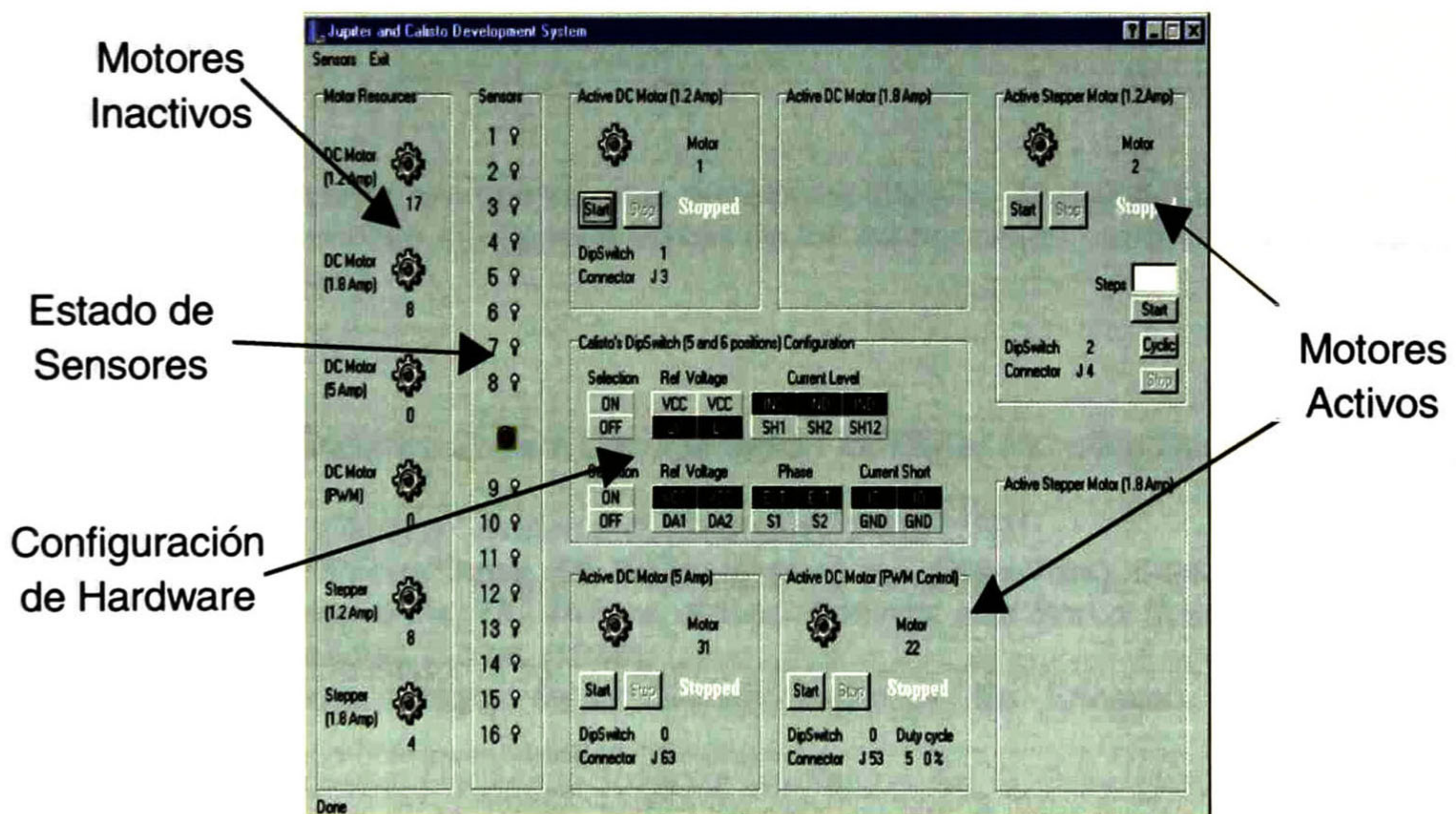


Figura 9. Interfaz gráfica del software de aplicación.

Características del software

El software fue desarrollado en objetos, con la herramienta C++ Builder 3.0 y se ejecuta como una aplicación del sistema operativo Windows 95/98/NT.

Las principales características de este programa son: 6 clases propias, 22 clases recipientes, 2050 líneas de código, 157000 líneas de compilación y un volumen de ejecutable de 475Kbytes.

6. CONCLUSIONES Y TRABAJO FUTURO

Durante la realización de este proyecto, se desarrolló una herramienta que simplifica de manera eficiente y disminuye el tiempo de diseño e implementación del esquema de control para una nueva generación de dispositivos manejadores de papel Hewlett Packard.

Se diseñaron y construyeron dos tarjetas de circuitos impresos denominadas: hardware del sistema y hardware de aplicación. Para el hardware de aplicación se diseñó e implementó un PLD, empleándose un EPM7064S de Altera.

Para el acceso a los algoritmos de HP (del manejo de motores y sensores) se elaboró el código correspondiente en lenguaje C para el firmware.

De igual forma, se diseñó e implementó un software de aplicación que contiene una interfaz gráfica, permitiéndole al usuario la capacidad para configurar, operar y controlar los motores y sensores, conectados al sistema electromecánico del manejador. De esta forma se generan los movimientos específicos requeridos para transportar el papel a través del dispositivo.

La herramienta contiene la capacidad para realizar muy diversas tareas de propósito general, acceder al protocolo de comunicación CAN 2.0B (interfaz manejador de papel-impresora), comunicarse con una PC por medio de un puerto serial, conocer el nivel lógico de sensores electrónicos y proporcionar las señales necesarias para ejercitar motores de corriente directa y de pasos.

Trabajo futuro

El uso de esta herramienta representa una plataforma para la investigación de los perfiles de movimiento de las trayectorias del papel a través de los manejadores, para diversos productos de calidad de Hewlett Packard.

REFERENCIAS

1. Hewlett Packard, *Hardware Drivers Documentation for Paper Handling Devices*, (Guadalajara), 1-58, (2000).
2. R. Bosch, *CAN Specification version 2.0*, (Stuttgart), 2-68, (1991).
3. S. Burrows, *A Laser Printer Book*, Ed. BlackLightning Inc, (New York), 5-25, 1994.
4. Reliance Motion Control Inc, *DC Motors, Speed Controls and Servo Systems Handbook*. Ed. Reliance Inc, (Minnesota), 2-114, (1996).
5. Ericsson Components, *Stepper Motor Control Handbook*, Ed. Ericsson, (Estocolmo), 11-82, 1995.
6. C. Merrill, *Microprocessor Hardware/Software Interfacing*, Ed. B. Brey, (1984).
7. Hewlett Packard, *Paper Handling Firmware Development Standard*, (Guadalajara), 1-32, (2000).
8. Philips Semiconductors, *Interfacing a Stand Alone CAN controller*, (Amsterdam), 1-66, (2000).
9. Hitachi Semiconductors, *H8S/2357 Series, Hardware Manual*, (Tokyo), 1-1084, (1998).
10. Altera, *Max Plus II, Programmable Logic Development System*, (San Jose), 12-300, (1995).
11. J.Liberty, *Beginning Object-Oriented Analysis and Design: With C++*, Ed. Wrox Press Inc, 35-68, (1998).
12. H. Schildt, *Borland C++*, Ed. McGraw Hill, 55-255, 1997.



CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL IPN UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, aprobó la tesis: "Herramienta para el desarrollo y pruebas de algoritmos computacionales, con aplicación a manejadores externos de papel Hewlett Packard" del Sr. Nahúm Vladimir Castillo Félix, el día 16 de Octubre de 2000.

Dr. José Luis Leyva Montiel
Investigador Cinvestav 3B
CINVESTAV DEL IPN
Guadalajara

Dr. Deni Librado Torres Román
Investigador Cinvestav 2C
CINVESTAV DEL IPN
Guadalajara

Dr. Aldo Gustavo Orozco Lugo
Investigador Cinvestav 2A
CINVESTAV DEL IPN
México, D.F.

M.E.I. Hernán Ildefonso Gutiérrez Vázquez
Ingeniero de Diseño
Hewlett Packard de México, S.A. de C.V.
Guadalajara, Jalisco



CINVESTAV
BIBLIOTECA CENTRAL



SSIT000003852