

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica



**Diseño e implementación del control de movimiento para el sistema
térmico del modulo experimental de *Burn-In***

**Informe de Proyecto de Graduación para optar por el título de Ingeniera en
Electrónica con el grado académico de Licenciatura/Bachillerato**

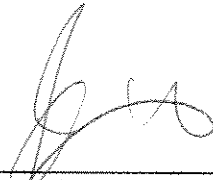
Mariel Monge Sequeira

Cartago, Junio de 2007

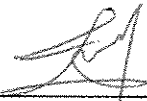
INSTITUTO TECNOLOGICO DE COSTA RICA
ESCUELA DE INGENIERIA ELECTRONICA
PROYECTO DE GRADUACIÓN
TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

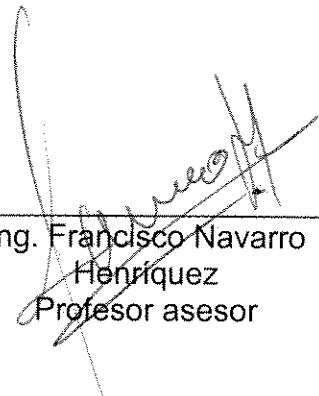
Miembros del Tribunal



Ing. Juan Carlos Jiménez Robles
Profesor lector



Ing. Marvin Hernández Cisneros
Profesor lector



Ing. Francisco Navarro
Henríquez
Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 11 junio de 2007
Lugar y fecha de la presentación

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Heredia, Junio de 2007



Mariel Monge Sequeira

Cédula: 6-336-993

Resumen

Actualmente en la planta de Intel en Costa Rica se realizan procesos de ensamblaje y prueba de los microprocesadores. Una de las pruebas que realizan es la de *Burn-In* la cual tiene propósito de acelerar las fallas de mortalidad infantil para que, en caso de que haya algún dispositivo defectuoso detectarlo en la fábrica y no llegue al cliente. Intel cuenta con una máquina a escala de producción que realiza este proceso y con otra a menor escala que es experimental, es decir, es la que usan los ingenieros para realizar pruebas de validación y, una vez depurado, se pasa a correr en la máquina de producción

Este proyecto es desarrollado en la máquina experimental, la cual es llamada *bench*. Esta máquina no cuenta con todas las características que la de producción, una de ellas es el de control de temperatura. El control de temperatura debe estar en capacidad de poner al microprocesador a una temperatura deseada. Actualmente esto se realiza en el *bench* por medio de dos cabezas térmicas que suministran o extraen calor según sea requerido. Estas cabezas deben hacer contacto con el microprocesador a probar. Este posicionamiento se realiza manualmente. Esto causa que para cada par de microprocesadores a probar se deba gastar en promedio 5 minutos solo colocando las cabezas. Evidentemente esta pérdida de tiempo se hace más significativa conforme aumenta la cantidad de unidades que deben probarse. A su vez conforme aumente el número de unidades que se prueben la confiabilidad se hace mayor puesto que la muestra es mayor.

Este problema se proyecta solucionar diseñando e implementando un sistema de posicionamiento para las cabezas térmicas, en el cual, mediante software el usuario pueda seleccionar las unidades que desea probar y el sistema por si mismo posicione las cabezas en el lugar correspondiente. Esto reducirá el tiempo que se debe emplear en este proceso. Además, de esto se va obtener otro beneficio de este proyecto que es que, como el posicionamiento se hace por software ya no se hace indispensable la presencia del ingeniero en el lugar en donde esta el *bench* para correr estas pruebas, se abre la posibilidad a que lo haga remotamente lo cual también implica un ahorro de tiempo debido a los procedimientos que deben seguirse para ingresar el piso de producción en donde esta la máquina

Abstract

Microprocessor's assembly and test processes are being developed by Intel Costa Rica. One of these tests is called Burn-In whose purpose is to accelerate Infant Mortality Failures in order to increase the Quality and Reliability of the product that is shipped to the customer. Test programs are run on each microprocessor to validate the Burn-In process. One machine at production scale is the one in charge of this process; moreover, Intel has another machine at experimental scale, this means that this machine is the one that is used by engineers on charge of this process to run validation tests and debug them before the test programs are run by the production scale machine.

This project is developed on the experimental machine which has all the capabilities as the production but one, thermal control which is the ability of getting the microprocessor to one predefined temperature. Currently this is being done by two thermal heads that take out/in heat from the microprocessors; in order to do this they should make physical contact with the units. This means that the thermal heads should be placed from one position to another each time a test is run on a unit. This movement is currently manually done and with an average of 5 minutes per pair of units the engineer in charge of running the test losses time. This waste of time gets worse if the amount of units is elevated. It is important to say that the more units tested the more reliable the process is.

A design and implementation of a movement system of the thermal heads which is going to enable the user to set the units he wants to test by software will be done to solve this problem. It's expected to decrease the time needed to make the movement and because the process will be automatic it can be done remotely.

Dedicatoria

*A todos los que de una u otra forma confiaron en mí,
aunque eso significara muchas veces su propio sacrificio.
Gracias por su apoyo incondicional durante esta, mi carrera*

Mariel

INDICE GENERAL

Capítulo 1: Introducción	1
1.1 Problema existente e importancia de su solución	2
1.1.1 Generalidades	2
1.1.2 Síntesis del problema	3
1.2 Solución seleccionada	3
Capítulo 2: Meta y Objetivos	6
2.1 Meta	6
Capítulo 3: Marco teórico	7
3.1 Descripción del proceso a mejorar	7
3.2 Antecedentes Bibliográficos	8
3.2.1 Proceso de <i>Burn-In</i>	8
3.2.2 Programas de prueba	8
3.2.3 <i>Burn-In Board (BIB)</i>	8
3.2.4 <i>Bench Application</i>	9
3.3 Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema	9
3.3.1 Dynamic Linking Library (DLL)	9
3.3.2 Protocolo RS232	10
3.3.3 EzVidCap Control	10
3.3.4 Visual Basic Script	10
3.3.5 Visual Basic 6.0	11
Capítulo 4: Procedimiento metodológico	12
4.1 Reconocimiento y definición del problema	12
4.2 Obtención y análisis de información	12
4.3 Evaluación de las alternativas y síntesis de una solución	12
4.3.1 Diseño de hardware	12
4.3.2 Diseño de software	13
4.3.3 Suplidores	13
4.4 Implementación de la solución	14
4.4.1 Implementación	14
4.4.2 Medios de difusión	14
4.5 Reevaluación y rediseño	14
Capítulo 5: Descripción detallada de la solución	16
5.3.1 <i>Bench Application</i>	16
5.3.2 VB6 Program	17
5.3.3 PLC Software	18
5.1 Análisis de soluciones y selección final	18
5.1.1 Software	18
5.1.2 Hardware	19
5.2 Descripción del hardware	22
5.2.1 Controlador Lógico Programable (Anexo B.1)	22
5.2.2 Driver posicionador (Anexo B.2)	24
5.2.3 Servo Motores (Anexo B.3)	24

5.2.4 Actuadores Lineales (Anexo B.4)	25
5.2.5 Guía Lineal	25
5.2.6 Webcam	25
5.3 Descripción del software	26
Capítulo 6: Análisis de Resultados	34
6.1 Resultados y análisis	34
6.1.1 Software	34
6.1.2 Hardware	42
7.1 Conclusiones	43
7.2 Recomendaciones	43
Bibliografía	44
Apéndices	45
A.1 Glosario, abreviaturas y simbología	45
A.2 Manual(es) de usuario	46
A.2.1 Instalación	46
A.2.2 Configuración de información para un nuevo arreglo de filas y columnas	46
A.2.3 Movimiento manual para determinar medidas de offset	48
A.2.4 Observar imagen proveniente de la webcam	50
A.2.5 Configurar unidades a probar e iniciar procedimiento	51
A.3 Información sobre la empresa/institución	54
Anexos	55
Anexo B.1 Hoja de datos del XLE OCS Model: HE-XE105	55
Anexo B.2 Hoja de datos del MR-J3-A/B	56
Anexo B.3 Hoja de datos del servomotor de la serie HPE/HC	57
Anexo B.4 Hoja de datos del actuador modelo GL-N	58
Anexo B.5 Hoja de datos de la guía modelo HSR	59

INDICE DE FIGURAS

Figura 1 Bench	1
Figura 2 Configuración actual del bench	2
Figura 3 Diagrama de bloques general de la solución seleccionada	5
Figura 4 Diagrama del proceso a mejorar	7
Figura 5 Diagrama de interrelación entre sistemas	16
Figura 6 Diseño final del sistema mecánico	20
Figura 7 Arreglo de filas y columnas para las BIB	21
Figura 8 Ejemplos de dispositivos que pueden ser utilizados con el XLe	22
Figura 9 Diagrama del PLC a utilizar	23
Figura 10 Driver posicionador a utilizar	24
Figura 11 Diagrama de flujo de precarga del formulario principal	27
Figura 12 Formulario principal de la aplicación VB6 (BIB.frm)	28
Figura 13 Diagrama de flujo de eventos del formulario principal	29
Figura 14 Diagrama de flujo de eventos del formulario para movimiento manual	29
Figura 15 Formulario para el movimiento manual (Manual.frm)	30
Figura 16 Diagrama de flujo para envío de comandos general	31
Figura 17 Diagrama de flujo para estados de espera basados en existencia de archivos	32
Figura 18 Inicialización del programa en Bench Application	33
Figura 19 Diagrama de flujo al correr los programas patrón en Bench Application	33
Figura 20 Ejemplo de archivo de variables iniciales (GetInfo.txt)	34
Figura 21 Archivo de comparación de arreglos del BIB	35
Figura 22 Fichero principal para el ejemplo	36
Figura 24 Archivo de información de mediciones del arreglo (3x6.txt)	36
Figura 25 Formulario para el movimiento manual para el ejemplo	40
Figura 26 Imagen del BIB desde la Webcam	42

Capítulo 1: Introducción

Una vez que cada microprocesador ha sido debidamente ensamblado es sometido a diferentes pruebas y procesos para así verificar su funcionalidad. Uno de estos procesos es el de Aceleración de Envejecimiento (el cual será llamado de aquí en adelante *Burn-In*¹), en el cual el objetivo principal es acelerar la aparición de las fallas de *mortalidad infantil* para que éstas sean capturadas en subsecuentes etapas antes de dejar la fábrica, y así incrementar la calidad y la confiabilidad del producto que se entrega al cliente.

La validación del proceso de *Burn-In* es realizada en Componentes Intel Costa Rica por el departamento llamado Calidad y Fiabilidad. Este proceso es realizado al 100% de la producción (para lo cual existe una máquina de producción). Sin embargo, además de esta máquina se cuenta con una a menor escala que es llamada *bench* en la cual se realiza el mismo proceso pero a menor escala. El objetivo del *bench* es permitir a los ingenieros realizar pruebas de forma experimental para así no afectar con ellas el proceso de producción. En la Figura 1 se muestra una foto de esta máquina.



Figura 1 Bench

El *bench*², a diferencia de su equivalente de producción, no cuenta con control térmico por lo que se han realizado proyectos con el fin de integrarle esa capacidad. Primero se añadieron dos cabezas térmicas controladas por medio de un *Chiller*³, (en la Figura 2 se muestra la configuración tal y como se encuentra actualmente). Sin embargo, las configuraciones referentes a temperatura se realizaban manualmente (por hardware) así como el posicionamiento de las cabezas térmicas.

¹ Glosario (Pág. 45)

² Glosario (Pág. 45)

³ Glosario (Pág. 45)

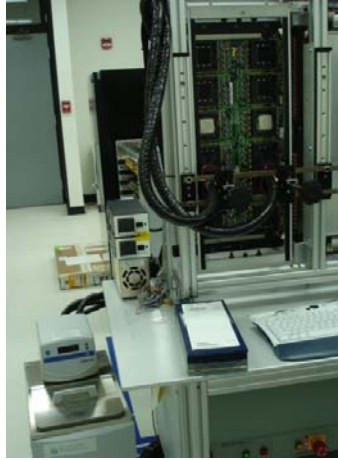


Figura 2 Configuración actual del bench

Es por esto que durante el año 2006 se desarrolló un proyecto con el fin de que las configuraciones de temperatura se pudieran realizar mediante software y no por hardware (manualmente)⁴

1.1 Problema existente e importancia de su solución

1.1.1 Generalidades

Intel actualmente cuenta con una máquina de producción que realiza el proceso de *Burn-In*. Además, Intel tiene otra a menor escala que es utilizada específicamente para realizar pruebas que luego pueden ser ejecutadas en la máquina de producción. Ésta cuenta con la mayoría de las características de la primera. Sin embargo, a diferencia de ella, por si sola no posee un control térmico. Es por esto que actualmente se utiliza un control de temperatura externo, éste posee dos cabezas térmicas las cuales son colocadas manualmente sobre los microprocesadores que se desean probar.

El hecho de que esto se haga manualmente crea la necesidad de la presencia de una persona en el lugar en donde está la máquina para colocar las cabezas sobre cada uno de los microprocesadores donde se realizan las pruebas.

Para cada prueba de *Burn-In* que se realiza por par de unidades una persona debe colocar manualmente las cabezas térmicas, esto toma alrededor de 5 minutos. El tiempo que tarda cada prueba ese en promedio de alrededor de 20 minutos. Esto significa que para cada par de unidades por prueba se tiene que emplear un 20% del tiempo sólo posicionando las cabezas.

⁴ Ver Bibliografía Zúñiga, Mauricio (Pág. 44)

Por otra parte, si se deben correr pruebas sobre todas las unidades del BIB (*Burn-In Board*)⁵ se debe esperar en el lugar a que terminen las pruebas en un par de unidades para poder posicionar las cabezas sobre el siguiente par y así sucesivamente hasta probar todas las unidades. Esto implica que, para el peor de los casos, un *BIB* de 18 unidades, el ingeniero encargado de correr las pruebas debe permanecer en el lugar 225 minutos (25 minutos por 9 pares de unidades) , esto es casi 4 horas de tiempo del ingeniero por *BIB* en el lugar corriendo las pruebas y posicionando las cabezas

Este procedimiento evidentemente se hace muy engorroso e implica a su vez una pérdida de tiempo para los ingenieros pues ellos deben estar corriendo pruebas (posicionando las cabezas térmicas) en lugar de emplear este tiempo para otras funciones

A continuación se explicará de forma general algunas particularidades de este proyecto:

- Los microprocesadores son colocados en placas llamadas *BIB* en forma matricial. La *BIB* es colocada y enganchada al bench mediante un sistema neumático. El arreglo de filas y columnas en el BIB varía dependiendo del producto. De esta manera existen diferentes configuraciones, entre ellas: 3x6, 2x6, 2x4. Es indispensable que el sistema por crear esté en la capacidad de funcionar con estas diferentes configuraciones así como de poder añadir nuevas configuraciones en caso de que las hubiesen
- Se debe tomar en cuenta factores como la orientación en que actualmente se encuentra el módulo donde se realizan pruebas, además del peso (cabeza térmica y mangueras) que debe poder manejar el sistema mecánico neumático
- Debe encontrarse la forma de poder integrar el software que controla el movimiento de las cabezas térmicas a la ya aplicación existente

1.1.2 Síntesis del problema

Pérdida de 20% del tiempo de realización de pruebas de *Burn-In* posicionando las cabezas térmicas sobre las unidades

1.2 Solución seleccionada

El problema existente se espera solucionar implementando un sistema mecánico-neumático que controle el posicionamiento de las cabezas el cual será a su vez manejado por la integración en el software actual de un módulo en el

⁵ Glosario (Pág. 45)

cual se hagan lecturas de información y a su vez que permita ajustar datos según se requiera

Los siguientes son aspectos que se deben tomar en cuenta en la solución de este problema:

- Es importante realizar una evaluación de ventajas y desventajas de la orientación en que podría ponerse la parte de la armazón del *bench* en que se coloca el *BIB*. Hay que tomar en cuenta factores como transferencia de calor al microprocesador, peso que deberían ser capaces de manejar los motores, impacto en la armazón por construir y repercusiones en cuanto a comodidad y seguridad para los que trabajan en esa máquina
- Se debe investigar y diseñar el sistema mecánico (armazón y piezas por utilizar) que mejor se ajuste a las necesidades actuales⁶
- Diseñar el sistema electrónico que cumpla con las características requeridas, esto es, que se ajuste al sistema ya existente. Este sistema electrónico incluye motores, sensores y dispositivos de control
- Se debe crear un algoritmo matemático inteligente que sea capaz de adaptar los movimientos de acuerdo a la *BIB* que se coloque para que no quede el proyecto limitado a una sola de éstas o a las existentes sino que quede abierta la posibilidad de configurarlo para nuevos *BIB*
- Investigar sobre la viabilidad de integrar el software por utilizar en el mismo lenguaje que se ha empleado con anterioridad ó en alguno que ofrezca más capacidades⁷. Considerando que de igual forma debe seguir en capacidad de utilizar los programas ya desarrollados
- No debe perderse de vista que lo que se pretende es facilitar un proceso, por lo tanto, lo que se desarrolle debe ser amigable con el usuario⁸ y debe estar debidamente documentado

En la Figura 3 se muestra un diagrama de bloques de la solución seleccionada

⁶ Para esto se contará con el apoyo de un ingeniero mecánico

⁷ Se evalúa la posibilidad de realizarlo en .net

⁸ Esto incluye validaciones en algoritmos

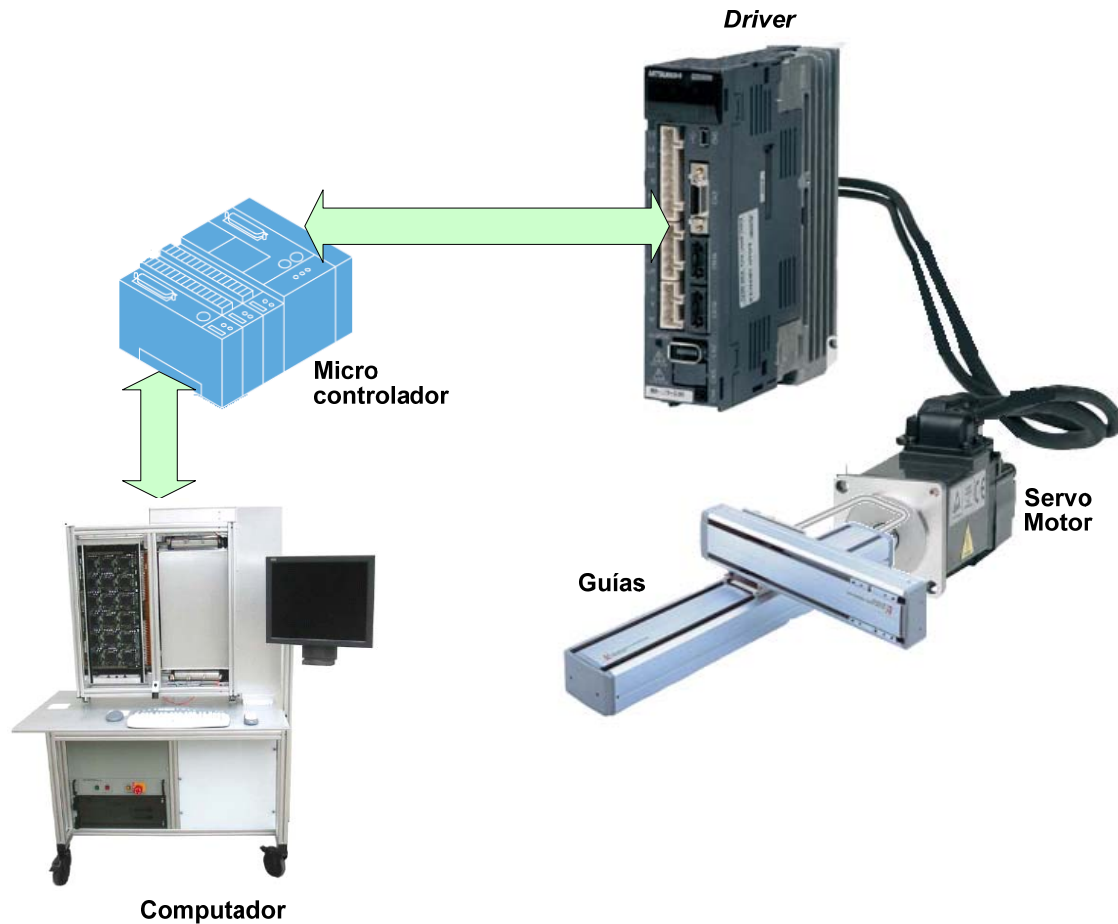


Figura 3 Diagrama de bloques general de la solución seleccionada

La computadora será utilizada para obtener información referente directamente a aspectos relacionados con la *BIB*, esto por medio de la aplicación específica de esta máquina. Estos aspectos incluyen: forma de arreglo de la *BIB*, presencia o ausencia de *DUTs* (*Device Under Test*⁹), entre otros. Es en la computadora donde también se realiza la interrelación entre ese programa con el que se encargará de controlar el posicionamiento de las cabezas térmicas por medio del envío de comandos de posicionamiento por el puerto serial al microcontrolador el cual es el encargado de traducir esos comandos en instrucciones de movimiento al *driver*¹⁰ que controla el motor y este a su vez las guías mecánicas que manejan las cabezas.

⁹ Glosario (Pág. 45)

¹⁰ Glosario (Pág. 45)

Capítulo 2: Meta y Objetivos

2.1 Meta

Disminuir a un 10% el tiempo empleado posicionando las cabezas térmicas sobre las unidades al realizar pruebas de *Burn-In*

2.2 Objetivo general

Diseñar e implementar el hardware y el software para el control del movimiento del sistema térmico del módulo experimental de *Burn-In*

2.3 Objetivos específicos

- Contactar y coordinar con el proveedor del sistema mecánico-neumático que permitirá el desplazamiento de las cabezas térmicas
- Diseñar e implementar el sistema electrónico para el control del sistema mecánico-neumático
- Integrar el software necesario al ya existente que permita el ingreso de datos y despliegue de la información referente al control de las cabezas térmicas
- Diseñar e implementar un sistema de monitoreo visual que pueda ser accedido remotamente mientras se ejecutan las pruebas

Capítulo 3: Marco teórico

3.1 Descripción del proceso a mejorar

En la Figura 4 se muestra un diagrama del proceso a mejorar con este proyecto

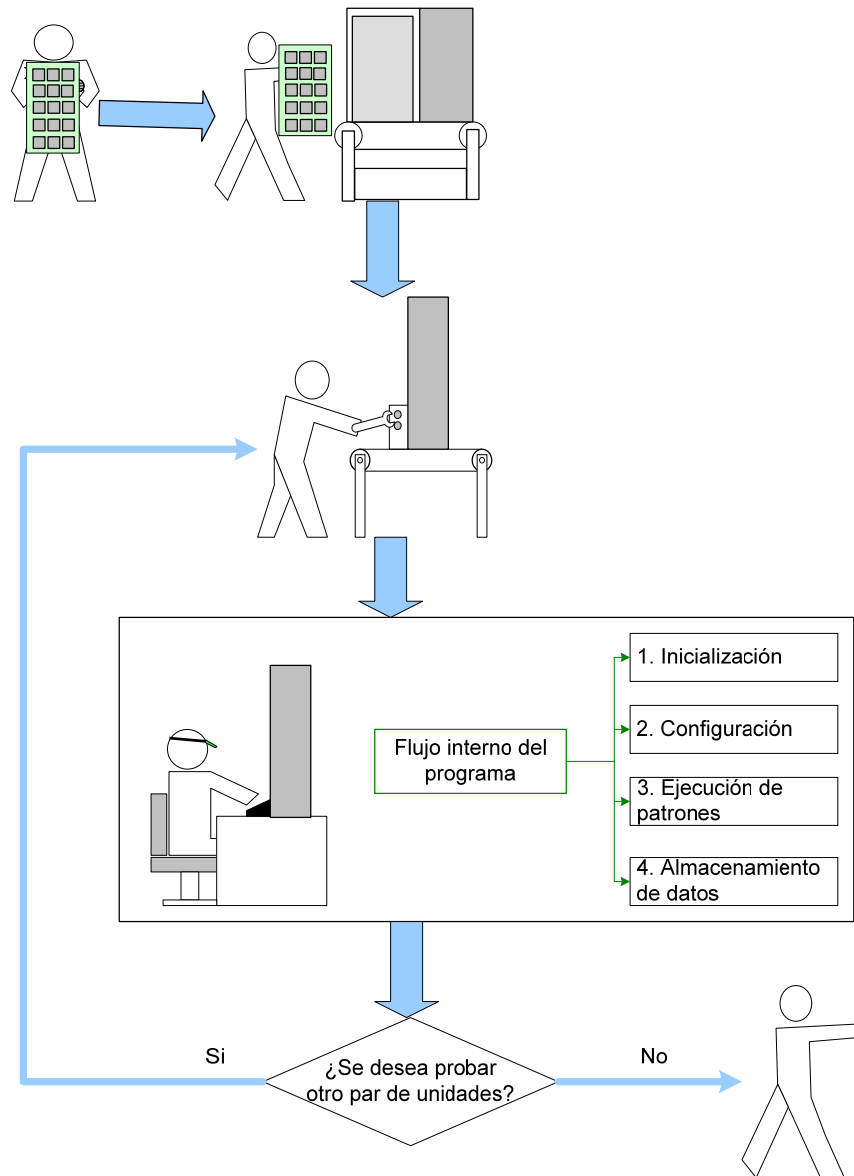


Figura 4 Diagrama del proceso a mejorar

Una vez que se han colocado las unidades en la *BIB* se procede a introducirla en el *bench*, cuando ésta se ha puesto correctamente y se ha realizado en enganche de ésta (por medio de *Bench Application*) se procede a colocar manualmente las cabezas térmicas sobre una o dos de las unidades.

Seguidamente se corren los correspondientes patrones. Para esto, en el paso de inicialización se cargan todos los datos en variables, luego los patrones son cargados. En el tercer paso los patrones son ejecutados al voltaje, tiempo y temperatura predeterminados. Por último toda la información es guardada; por lo general, en archivos de la computadora.

En este punto ya se ha terminado el proceso de *Burn-In* sobre dos unidades, pero si se quiere realizar este proceso sobre otras dos unidades se deben volver a colocar las cabezas térmicas sobre el otro par de unidades y así sucesivamente hasta que se hayan probado todas las unidades deseadas que es cuando termina el proceso

3.2 Antecedentes Bibliográficos

3.2.1 Proceso de *Burn-In*

El propósito del proceso de *Burn-In* es acelerar las fallas de mortalidad infantil para a su vez reducir el número de defectos por millón aumentando la calidad y confiabilidad del producto que es entregado al cliente.

El proceso de *Burn-In* es un ambiente donde las unidades son sometidas a condiciones más severas que a las que normalmente estaría sometido. Por ejemplo, temperaturas y voltajes más altos.

Una vez que las unidades son sometidas a esas condiciones son ejecutadas diversas pruebas con el fin de conmutar cada componente dentro de la unidad para que, en caso de que haya un defecto latente pueda ser detectado durante esta prueba

3.2.2 Programas de prueba¹¹

Los programas de prueba son ejecutados en las unidades con el fin de detectar posibles fallas en éstos.

Los patrones de prueba deben cubrir el 80% de los puertos y la unidad lógica, y el 100% de la memoria *RAM* de cada microprocesador.

El microprocesador se divide en tres áreas funcionales: memoria, lógica y buffers de entrada/salida. Cada área posee sus propias características físicas y de diseño por lo que existe una prueba de *BI* diferente para una de ellas.

3.2.3 *Burn-In Board (BIB)*

Las *Burn-In Board* constituyen el hardware utilizado con las capacidades eléctricas para ejecutar el proceso de *Burn-In* en las unidades

La *BIB* son diseñadas bajo el estándar de arquitectura de prueba definido por la *IEEE* en la norma 1149.1, el cual corresponde al denominado *JTAG*, que corresponde a un bus de comunicación serie con el microprocesador, por medio del cual se ejecutan los

¹¹ Glosario (Pág. 45)

patrones de prueba. Además la *BIB* cuenta con un puerto de acceso de pruebas denominado *TAP*¹².

3.2.4 Bench Application

Esta es la aplicación que maneja y a la vez monitorea lo que sucede durante el proceso de *Burn-In*. En el caso del proyecto la máquina que controla es una de tipo experimental, es decir, se utiliza para desarrollar pruebas de laboratorio en especial de *debugueo* de los patrones de prueba.

Esta aplicación controla valores de temperatura, voltaje y frecuencia entre otros

3.3 Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema

3.3.1 Dynamic Linking Library (DLL)¹³

DLL es el acrónimo de Dynamic Linking Library término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo. Esta denominación se refiere a los sistemas operativos Windows siendo la extensión con la que se identifican los ficheros, aunque el concepto existe en prácticamente todos los sistemas operativos modernos.

Las DLLs pueden verse como la evolución de las bibliotecas estáticas y de forma análoga contienen funcionalidad o recursos que utilizan otras aplicaciones. Su uso proporciona algunas ventajas:

- Reducen el tamaño de los archivos ejecutables: Gran parte del código puede estar almacenado en bibliotecas y no en el propio ejecutable lo que redundaría en una mejor modularización
- Pueden estar compartidas entre varias aplicaciones: Si el código es suficientemente genérico, puede resultar de utilidad para múltiples aplicaciones (por ejemplo, la MFC es una biblioteca dinámica con clases genéricas que recubren la API gráfica de Windows y que usan gran parte de las aplicaciones).
- Facilitan la gestión y aprovechamiento de la memoria del sistema: La carga dinámica permite al sistema operativo aplicar algoritmos que mejoren el rendimiento del sistema cuando se carguen estas bibliotecas. Además, al estar compartidas, basta con mantener una copia en memoria para todos los programas que la utilicen.

¹² Glosario (Pág. 45)

¹³ Tomado de <http://es.wikipedia.org/wiki/DLL> (Ver Bibliografía)

- Brindan mayor flexibilidad frente a cambios: Es posible mejorar el rendimiento o solucionar pequeños errores distribuyendo únicamente una nueva versión de la biblioteca dinámica.
Sin embargo, no todo son ventajas. En los sistemas Windows, las DLLs son muy comunes y muchos programas usan las mismas DLLs. Pero debido a la evolución, cada una de las DLLs evoluciona incorporándose mejoras pero modificándolas de tal forma que dejan de ser compatibles. Esto puede producir dos efectos no deseados: que la instalación de un programa reemplace una DLL con una nueva versión incompatible y que la desinstalación del programa borre una DLL compartida
Las versiones más modernas de Windows y los nuevos scripts de instalación MSI (sobre todo su característica de instalaciones residentes) abordan este problema. Sin embargo, el problema se mantiene cuando se utilizan otros instaladores (versiones antiguas) o se realizan modificaciones manuales.
Las DLL es una técnica de implementación de forma dinámica es decir que la computadora solo la carga cuando el programa que la usa la llama.

3.3.2 Protocolo RS232¹⁴

El puerto RS232 proporciona ventajas en cuanto al manejo de sistemas similares a este Proyecto debido a que proporciona una interfaz en que es posible enviar y recibir información fácilmente
El RS232 es un estándar de comunicaciones propuesto por la Asociación de Industrias Electrónicas (EIA)

3.3.3 EzVidCap Control

Este control es usado para acceder a la imagen de la webcam que será colocada con el fin de observar remotamente lo que sucede en la máquina.

Este es un control que necesita ser registrado usando *regsvr32.exe* y luego debe ser agregado como componente de la aplicación

3.3.4 Visual Basic Script¹⁵

Se debe investigar sobre este programa debido a que *Bench Application* esta basada en éste y es necesario desarrollar métodos en él para asegurar el correcto funcionamiento

¹⁴ Ver Bibliografía

¹⁵ Ver Bibliografía

3.3.5 Visual Basic 6.0¹⁶

Este es el programa en el que se desarrolló la aplicación para este proyecto, tiene las siguientes ventajas:

- La facilidad del lenguaje permite crear aplicaciones para Windows en muy poco tiempo. En otras palabras, permite un desarrollo eficaz y menor inversión en tiempo que con otros lenguajes.

- Permite **generar librerías dinámicas (DLL) ActiveX** de forma nativa y **Win32** (no ActiveX, sin interfaz COM) mediante una reconfiguración de su enlazador en el proceso de compilación.

- Permite la utilización de formularios (*Forms*) tanto a partir de recursos (como en otros lenguajes) como utilizando un IDE para diseñarlos

¹⁶ Ver Bibliografía

Capítulo 4: Procedimiento metodológico

4.1 Reconocimiento y definición del problema

Para identificar el problema se realizaron las siguientes actividades:

1. Reunión con el asesor de la empresa en el laboratorio correspondiente con el fin de que me diera una idea general del problema que deseaban solucionar
2. Estudio de información referente al proceso que se lleva a cabo en la máquina donde se realiza el proceso
3. Estudio de manuales de funcionamiento de la máquina en cuestión
4. Reunión con el asesor de la empresa y el equipo de trabajo con el fin de definir alcances y limitaciones

4.2 Obtención y análisis de información

La información preliminar necesaria para determinar el problema fue obtenida por medio de entrevistas tanto al asesor de la empresa como a la técnica *QRE (Quality and Reliability Engineer)* quien ha trabajado en esta máquina también.

Por otro lado, para el conocimiento de proceso que realiza esta máquina se hizo por medio de manuales de la compañía y de entrenamiento con los ingenieros del grupo

4.3 Evaluación de las alternativas y síntesis de una solución

La solución seleccionada se hizo tomando varios aspectos en cuenta. Lo más relevante; sin embargo, fue la consulta con el asesor de la empresa, conforme se fueron definiendo las necesidades se determinó también como se debía proyectar la solución. Los detalles a su vez fueron modificados según fue requerido

4.3.1 Diseño de hardware

En la parte de diseño del hardware, con objetivo de plantear las alternativas de solución, se contactó a varias empresas que ofrecen este tipo de servicios. A éstos se les presentó el proyecto y en conjunto con ellos se evaluaron diferentes alternativas.

La primer idea fue montar un sistema mecánico independiente para cada una de las cabezas térmicas, sin embargo, conforme se fue contactando a los proveedores esta idea se fue modificando de tal manera que no se saliera de los requerimientos de la empresa en cuanto a funcionamiento y costo.

Evidentemente, la idea de que las cabezas estuvieran mecánicamente separadas una de la otra tenía un costo muy alto, por lo que, se le consultó al asesor de la empresa si esto era estrictamente necesario ya que, aún si no fuesen independientes, el funcionamiento que se podría lograr no afectaría negativamente el rendimiento.

4.3.2 Diseño de software

En cuanto a software se refiere, el entrelazar las dos aplicaciones involucradas (la desarrollada para el control de movimiento y *Bench Application*) constituyó el reto más grande.

La primera idea que se evaluó como solución a esta parte fue tomando en cuenta un proyecto de graduación desarrollado por un estudiante de electrónica el año pasado en la misma aplicación¹⁷ el cual utilizaba dlls como medio de comunicación entre aplicaciones. Sin embargo, a la hora de realizar pruebas controladas de laboratorio hubo problemas en cuanto al funcionamiento de la interfaz con el usuario.

Dados estos resultados se procedió a investigar con desarrolladores de software y ellos expresaron que, por lo general, los dlls dan problemas al trabajar con esas interfaces.

Es por eso que se procedió a diseñar otra solución. Ésta estaba basada en el intercambio de la información por medio de archivos. Esta idea funcionó muy bien para información que se escribe y se lee una sola vez (por ejemplo el número de serie de la BIB, el número de filas y de columnas); sin embargo, no es factible para información que cambia dinámicamente y que, en el caso del proyecto, implica estados de espera de monitoreo hasta que el valor de esa información cambie a un valor deseado para poder proseguir debido a un conflicto que se da al abrir y cerrar el archivo para leerlo y para almacenar la información desde una aplicación y desde otra simultáneamente. Por lo cual, se mantuvo la idea de la escritura y lectura de archivos sólo para variables estáticas. Para las variables dinámicas (que en el caso del proyecto son de sólo dos estados: *true* ó *false*) se recurrió a una tercer idea: escribir y/o borrar un archivo determinado de acuerdo al cambio de la variable; de esta manera, lo que se monitorea no es lo que contenga el archivo sino la existencia de éste o no, lo cual a su vez mejora el rendimiento del algoritmo debido a que se evita abrirlo, leerlo y cerrarlo para monitorear la variable. Se probó el funcionamiento de esta opción y se obtuvieron resultados positivos

4.3.3 Suplidores

También se tuvo que determinar cuál era la opción más factible entre proveedores basado en lo que cada uno de ellos ofrecía, en cómo eso se ajustaba a nuestras necesidades y en el presupuesto y costo de esos servicios.

¹⁷ Mauricio Zúñiga Vásquez

El proveedor escogido al final fue el que más se acercaba a los requerimientos, en especial de costo y tiempo de implementación.

El procedimiento por medio del cual éste fue escogido fue comparándolo con los demás, se presentaron las opciones al gerente del equipo cada una con sus ventajas y desventajas y, basado en los diferentes aspectos, fue seleccionado como el apropiado para el proyecto

4.4 Implementación de la solución

4.4.1 Implementación

Una vez que se definió el problema existente y las generalidades del proyecto se procedió a diseñar diagramas de bloques de la solución y a determinar las funcionalidades de cada uno de esos bloques. Se hicieron análisis de consistencia en ese diseño y conforme se avanzaba en el proyecto se fue detallando y materializando cada uno de esos bloques.

Una vez implementado, éste fue evaluado de acuerdo al desempeño obtenido respecto del esperado con resultados satisfactorios.

4.4.2 Medios de difusión

El proyecto será expuesto ante el grupo de trabajo que será el usuario final de éste, con el fin de darles una explicación general de cómo fue desarrollado, de entrenarlos y de obtener retroalimentación de su parte para futuras mejoras.

Es importante además mencionar que en la empresa se desarrollan *Technical Forums* que son reuniones con un tema o proyecto específico que tiene como objetivo difundir los aspectos técnicos más relevantes. El jefe del departamento en donde fue desarrollado este proyecto ha planeado que se lleve a cabo uno de éstos para el proyecto

Como parte de los requerimientos de la empresa se espera que ciertos materiales sean traducidos al inglés, entre ellos, la guía de usuario, manual de instalación y diagramas de funcionamiento. Asimismo, se espera poder implementar este proyecto a otras plantas de Intel que cuentan con la misma máquina con el fin de proveerles de las ventajas que éste ofrece.

4.5 Reevaluación y rediseño

Quizá una de las mejoras más importantes que se pueden realizar a este proyecto es integrarlo a la *ethernet* de la empresa. Esto con el fin de mejorar la accesibilidad a éste.

Migrar la aplicación a una plataforma más reciente como lo son .Net de Microsoft o Java de Sun Micro Systems con el fin de aportar nuevas capacidades

Capítulo 5: Descripción detallada de la solución

La Figura 5 muestra el diagrama de interrelación entre los sistemas involucrados en este proyecto.

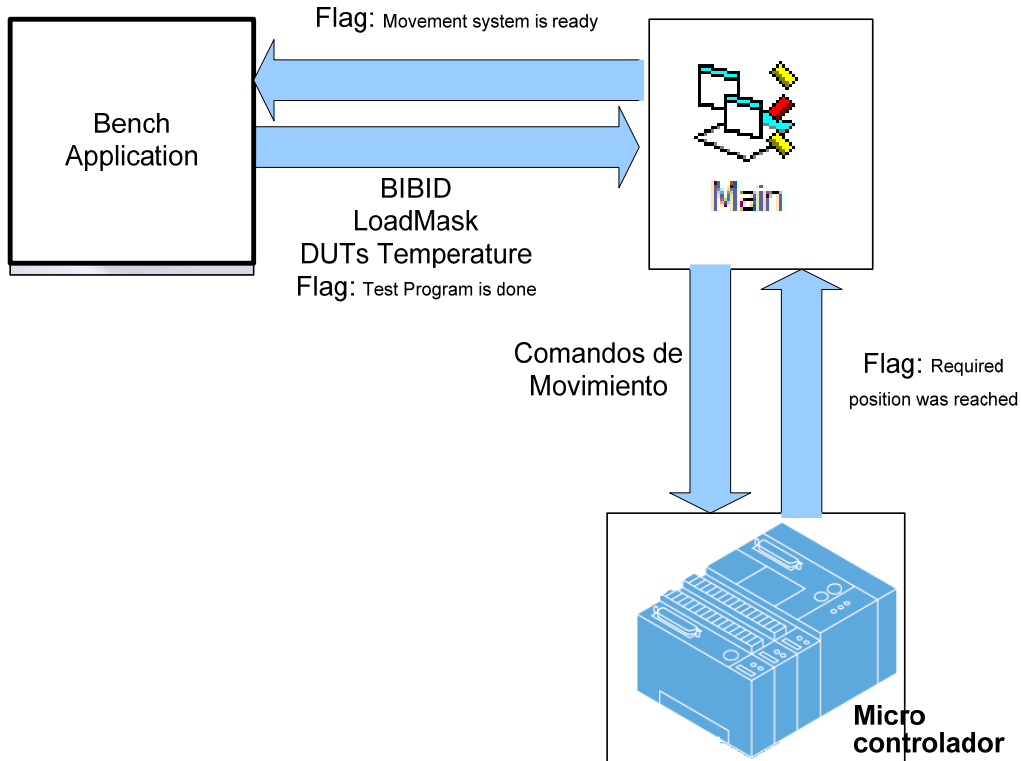


Figura 5 Diagrama de interrelación entre sistemas

A continuación se explican sus funcionalidades detalladamente:

5.3.1 Bench Application

Bench Application es una aplicación que fue diseñada específicamente para el control de esta máquina. *Bench Application* controla directamente todo lo que ocurre en el *Bench*. Desde esta aplicación es posible desde encender las fuentes de los microprocesadores hasta correr patrones de pruebas de *Burn-In*

En este proyecto se aprovechan las funcionalidades de esta aplicación para obtener la siguiente información

- tipo de *BIB*: esta se encuentra almacenada en una EEPROM interna de cada una de las *boards* como un número de serie

- número de filas y columnas: se encuentra almacenada al igual que el número de serie de la *BIB* en una EEPROM interna
- máscara de unidades: por medio de esta máscara (que es en forma de arreglo) es posible determinar cuáles unidades están presentes en los *sockets* y cuáles no

Ésta aplicación debe monitorear de alguna forma el estado del sistema de movimiento para asegurarse del momento cuando puede empezar a correr un patrón; asimismo, esta aplicación debe enviar una señal a la VB6 aplicación para informarle que se ha terminado de correr un patrón y que, por lo tanto, se puede mover al siguiente par de unidades

5.3.2 VB6 Program

En este programa es donde se desarrolla toda la lógica mediante la cual debería funcionar el posicionamiento de las cabezas térmicas

- toma el número de serie de la *BIB* y en conjunto con los valores leídos de números de filas y columnas determina el arreglo que debe tener la *BIB* y de acuerdo a eso carga dinámicamente los *checkboxs* que dan opciones al usuario para elegir cuales unidades probar
- con la máscara de unidades habilita o deshabilita los *checkboxs* que permiten al usuario elegir unidades a probar con el fin de prevenir errores por ejemplo posicionar alguna de las cabezas térmicas sobre un *socket* sin unidad.
- lee y guarda archivos con información referente a distancias para cada *BIB ID*, con el fin de no perder esta información cada vez que se cierra el programa
- lee archivos de relación entre el número de serie de la *BIB* con el arreglo correspondiente. Con el propósito de dejarlo a la posibilidad de expansión en caso de que sea requerido
- hace posible el movimiento manual dado la necesidad de medir los niveles de *offset* para cada una de los *BIB*
- determina cuál o cuales cabezas térmicas mover utilizando el método más efectivo en cuanto a movimientos se refiere
- envía comandos que incluyen distancias absolutas en x y y, no posiciones, esto con el fin de facilitar el proceso en que los comandos se ven envueltos
- envía comandos de posicionamiento por medio del puerto serial (protocolo RS232)
- espera por un comando de retroalimentación para asegurarse de que se ha alcanzado la posición deseada

- envía comando del siguiente posicionamiento sólo cuando se ha recibido notificación de que el patrón ha terminado de correr
- los comandos de posicionamiento incluyen el desplazamiento en y, en x e información que indica cuáles cabezas deben utilizarse para ese movimiento
- posee modulo de verificación de información antes de enviar cada comando, para evitar errores en cuanto a límites específicamente

5.3.3 PLC Software

- maneja los *drivers* de los motores, mediante los comandos que son enviados por el puerto serie
- se posiciona en el lugar indicado por el comando
- mueve las cabezas según lo indiquen los comandos
- envía un comando de retroalimentación cuando se ha alcanzado una posición

5.1 Análisis de soluciones y selección final

5.1.1 Software

Hay dos tipos de variables que deben intercambiarse entre aplicaciones: estáticas y dinámicas de dos estados.

Las estáticas son valores que se leen una vez, se cargan se toman en cuenta para la toma de decisiones al inicio y no son evaluadas más (número de serie del *BIB*, número de columnas, número de filas y máscara de *DUTs*).

Las dinámicas de dos estados son valores de tipo *boolean* que deben ser monitoreadas dinámicamente tanto en la aplicación de VB6 como en Bench Aplicación. Éstas son indicadores que generan estados de espera en ambas aplicaciones para asegurarse que se puede o no continuar un determinado procedimiento. Por ejemplo, VB6 genera una de estas variables para informar que el sistema de movimiento ha sido correctamente posicionado. Por su parte Bench Application monitorea esta variable cada vez que esta listo para correr un nuevo patrón y éste debe esperar hasta que esa variable tome el valor de *true* para poder iniciar, si la variable es *false* debe mantenerse monitoreándola

La primera idea para interconectar las dos aplicaciones principales involucradas en el proyecto fue utilizando dlls, (las cuales permiten el intercambio dinámico de información) en ésta las interfaces con el usuario estaban encapsuladas dentro del dll. De esta forma, lo que se hacía era: desde *Bench Application* se

creaba un objeto de ese dll y se intercambiaban valores de variables de un programa a otro utilizando las funciones de los módulos de clase definidas previamente en el dll. Esta opción fue modificada debido a que el manejo de interfaces con el usuario presentó problemas al correr la aplicación. Estos problemas incluyeron:

- la aplicación se congelaba al tratar de correrla
- los eventos no respondían
- no se seguían los procedimientos esperados

Por este motivo se decidió utilizar un ejecutable que era el que contenía los interfaces con el usuario y, por medio de grupos de proyectos, se podían hacer llamados a las rutinas de los dlls por medio de los cuales se pretendía intercambiar la información dinámicamente; sin embargo, el problema en este caso fue que, para cargar las variables dentro de esos dlls se crea un objeto desde *Bench Application* y para poder llamar esas funciones se debe crear otro objeto, aún si las variables del dll están definidas como globales no es posible intercambiar esa información puesto que se habla de dos objetos totalmente independientes uno de otro. Por esto se decidió cambiar la idea de utilización de dlls por archivos.

No se encontró ningún problema al intercambiar las variables estáticas entre aplicaciones; sin embargo, no sucede lo mismo con las dinámicas esto debido a que el monitoreo requiere abrir, leer y cerrar el archivo en un ciclo lo que, eventualmente, cuando se cambie el valor de esa variable desde la otra aplicación y se necesite escribir en el archivo habría un conflicto entre las aplicaciones.

Este conflicto es porque las dos aplicaciones requerirían abrir el archivo al mismo tiempo, una para monitorearla y la otra para escribirlo.

A este problema se le dio solución al no monitorear lo que había en el archivo sino la existencia del archivo; esto, aprovechando la ventaja de que las variables dinámicas son solo de dos estados. De esta manera en uno de los estados el archivo no existe y en otro se crea el archivo.

Esta solución cumple con los requerimientos para el intercambio de la información entre aplicaciones

5.1.2 Hardware

El diseño de la parte mecánica-neumática se hizo en conjunto con los diferentes suplidores, basados en las necesidades del proyecto y en sus ideas y modificaciones que se fueron realizando.

La primera idea era que las dos cabezas térmicas tuvieran movimiento completamente independiente una de la otra, esta idea; sin embargo, fue descartada debido a lo complejo que sería el diseño mecánico y por ende al alto precio que involucraría.

Se cuestionó entonces la necesidad de la independencia de esas cabezas. Si se analiza con detenimiento se encontrará que lo que se requiere es posicionamiento de ellas sobre las unidades seleccionadas por el usuario, si se diseña un sistema lo suficientemente eficiente se podría aprovechar al máximo el tiempo al tratar de colocar las 2 cabezas al mismo tiempo, en caso de que ellas dependieran en un eje implicaría que estarían en el mismo eje, pero que diferencia hay en esto a que puedan estar en un eje diferente si por lo general se requerirían mover a todos los lugares? Claro está, que dependiendo de la configuración podrían haber casos en que la independencia de las cabezas agilizaría el proceso, pero por esos casos aislados no vale la pena el costo económico que involucraría.

El sistema mecánico que mejor se ajustó a las necesidades es el mostrado en la Figura 6

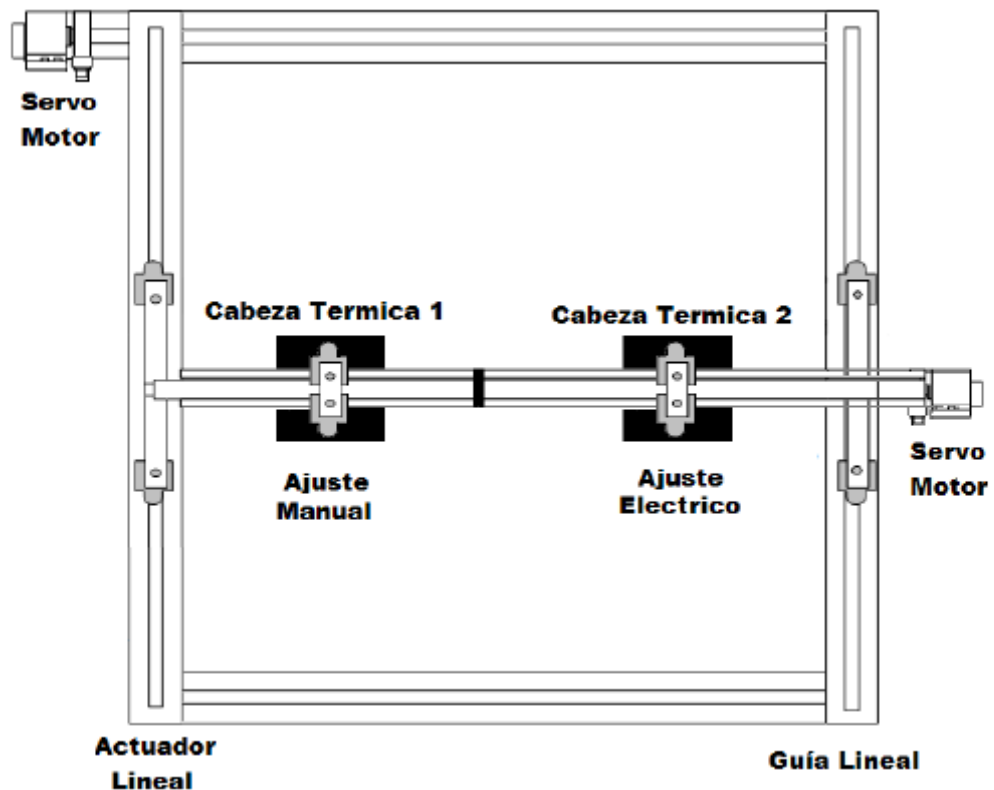


Figura 6 Diseño final del sistema mecánico

Como se observa, en el diseño final las cabezas son dependientes en el eje y, por otro lado en el eje x (se pueden manejar un máximo de 4 filas; sin embargo, habrán 3 máximo por limitantes de hardware) quedó diseñado de la siguiente forma: la cabeza térmica 1 siempre deberá posicionarse sobre la última fila (ver Figura 7), contará sólo con ajuste (en x) manual, la cabeza térmica 2 debe estar en capacidad de posicionarse sobre la fila 1 ó la fila 2 según sea el caso por lo que se hizo necesario que el ajuste de ésta sea eléctrico y que contara con un actuador lineal en ese eje (se recuerda que las posiciones pueden variar de acuerdo al *BIB*)

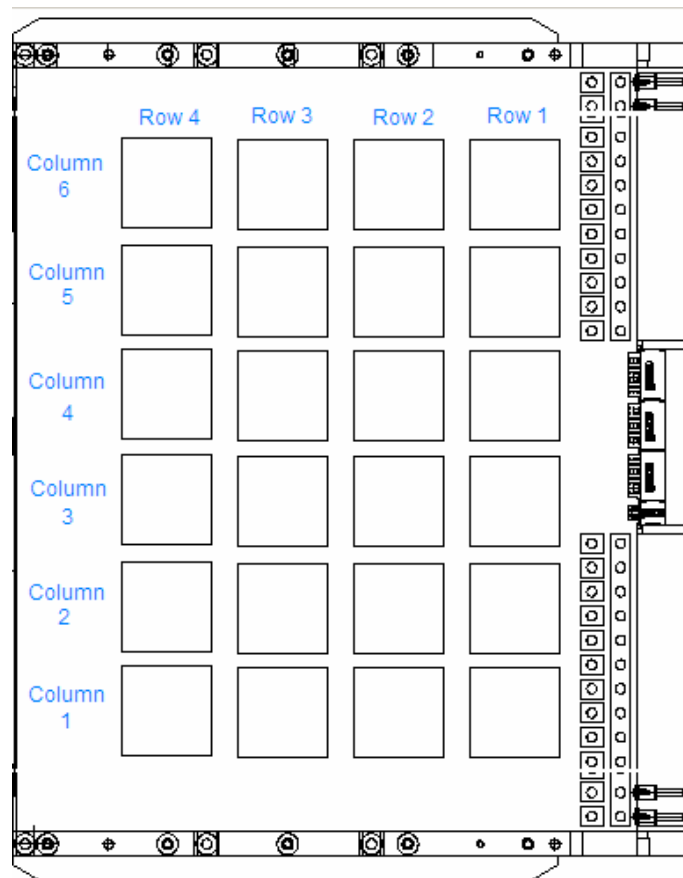


Figura 7 Arreglo de filas y columnas para las BIB

5.2 Descripción del hardware

5.2.1 Controlador Lógico Programable (Anexo B.1)

El XLe (el PLC a utilizar) tiene capacidad para conectarse a gran variedad de dispositivos. La Figura 8 muestra ejemplos de los dispositivos que pueden ser usados con el XLe

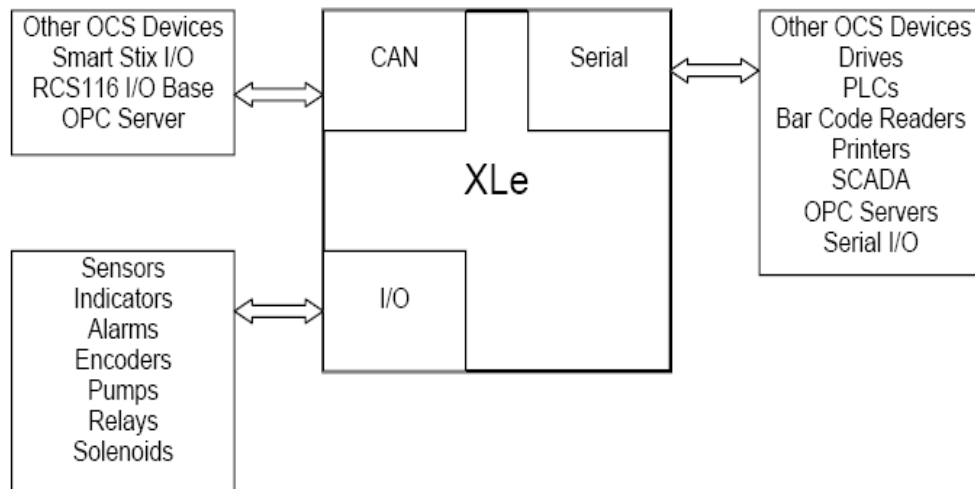


Figura 8 Ejemplos de dispositivos que pueden ser utilizados con el XLe

En el caso del proyecto son de gran relevancia las características de los I/O pues permitirán conectar las salidas de sensores de fin de carrera¹⁸ y los encoders de los servomotores. Por otra parte, las Serial I/O permitirán la comunicación con la PC para recibir comandos y para enviar señal de retroalimentación así como para comunicarse con el driver posicionador

A continuación se listan las características de interés de este PLC:

- Pantalla LCD (mostrar posición actual): esta pantalla en un futuro se puede utilizar por ejemplo para movimiento local
- Compensador PID: necesario para el lazo de control del sistema de posicionamiento
- Puerto de red CsCAN para comunicación remota de I/O, otros PLCs o PCs
- Protocolos seriales configurables para comunicación con drivers, PLCs y otros periféricos seriales
- Módulos expandibles opcionales que permiten capacidades adicionales como ethernet¹⁹ o módems

¹⁸ Es importante destacar que esta restricción también será controlada por medio del software diseñado; sin embargo, se hace esta otra verificación para hacer del sistema lo más seguro posible

¹⁹ Referirse a las recomendaciones

La Figura 9 muestra el PLC y sus partes principales

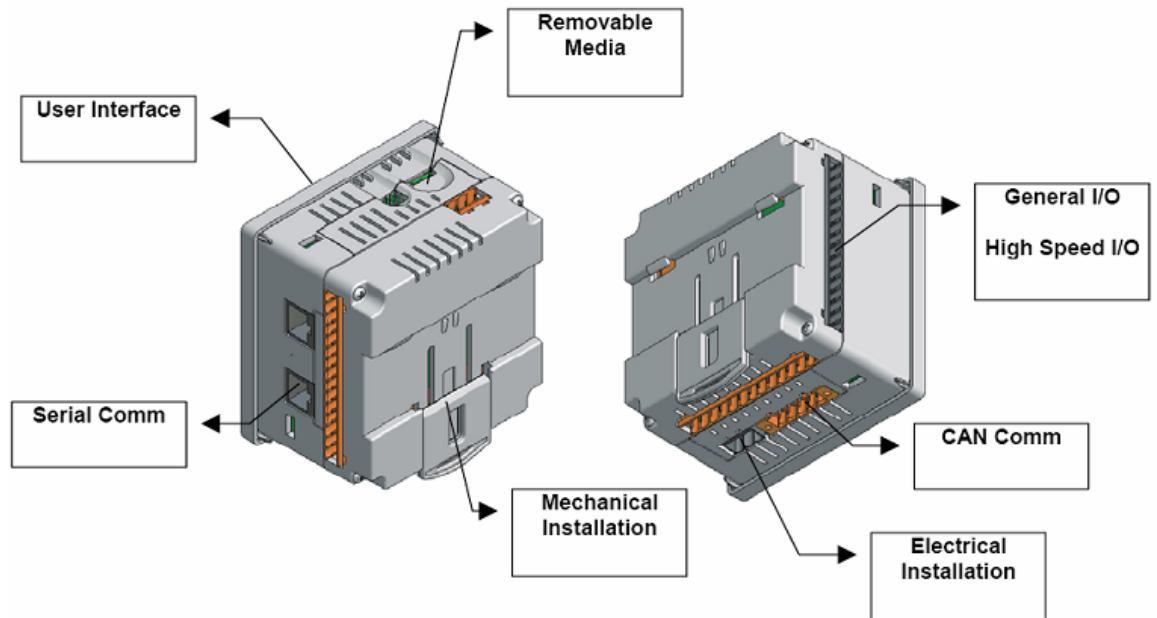


Figura 9 Diagrama del PLC a utilizar

Todos los modelos de XLe poseen 2 puertos seriales los cuales son implementados con módulos de conectores RJ45 de 8 pines, éstos son nombrados MJ1 y MJ2

El MJ1 contiene las interfaces half-duplex RS-485 y la RS-232 con handshaking mientras que el MJ2 contiene RS485 full-duplex y la RS-232 sin handshaking. Para efecto de este proyecto se pretende utilizar el MJ2 con protocolo RS-232 sin handshaking

Existen 6 tipos de I/O que pueden o no tener los PLCs. A continuación se detallan las diferencias entre ellas y se señala las que posee el modelo de XLe escogido:

- salidas digitales de estado sólido: activar dispositivos de baja corriente y voltaje. Característica del modelo HE-XE105. Puede ser utilizado como fuente de sensores
- salidas relay: manejar dispositivos de alta corriente y voltaje. No es característica del modelo HE-XE105, no es necesario
- Entradas digitales: entradas DC de bajo voltaje (12-24VDC). Característica del modelo HE-XE105
- Entradas analógicas: permite mediciones de corriente y voltaje de gran cantidad de dispositivos (posee filtros para eliminar el ruido). No es característica del modelo HE-XE105, no es necesario

- Entradas analógicas universales: interfaz para dispositivos analógicos: voltaje, corriente, temperatura. Característica del modelo HE-XE105
- Salidas analógicas: salidas de alta resolución de voltaje y corriente. Característica del modelo HE-XE105

5.2.2 Driver posicionador (Anexo B.2)

En la Figura 10 se muestra este driver posicionador



Figura 10 Driver posicionador a utilizar

Algunas de las características de este driver son:

- Elimina exceso de vibración en la carga aumentando rendimiento y respuesta del sistema
- Encoder de alta resolución
- Modos de control de: posición, torque, velocidad. Modo de control de posición es el de interés a nivel de este proyecto
- Sistema de detección de posición absoluta
- Interfaz de comunicación RS422 (cable de conversión requerido)

5.2.3 Servo Motores (Anexo B.3)

El Anexo B.3 muestra la hoja de datos de los servomotores que se usarán en este proyecto.

El modelo específico de servomotores a utilizar será escogido de acuerdo al recomendado por el fabricante del driver a utilizar. Esto aún no se ha realizado debido a que no se ha determinado aún las guías y actuadores lineales a utilizar por lo cual no se puede estimar el torque requerido

5.2.4 Actuadores Lineales (Anexo B.4)

Los actuadores lineales a utilizar son del modelo GL-N. Aún no se está trabajando en determinar cuál es que mejor se ajusta a las necesidades del proyecto

5.2.5 Guía Lineal

Las guías a utilizar son del modelo HSR debido a que se ajusta a requerimientos de distancias

5.2.6 Webcam

En cuanto a la webcam a utilizar no se tienen especificaciones detalladas, puede ser cualquier modelo que funcione en Windows XP.

Las pruebas durante el desarrollo del proyecto se realizaron utilizando una HP Digital Webcam

5.3 Descripción del software

A continuación se mostrarán diagramas detallados del funcionamiento del software desarrollado en VB6 en este proyecto. Además, para facilitar la comprensión del mismo se muestran dibujos de las pantallas involucradas.

En la Figura 11 se muestra lo que sucede antes de cargar el primer formulario. Esto sucede de esta forma debido a que lo que se muestre en él depende de esas variables.

Las variables que se leen en la precarga con valores que han sido almacenados en la inicialización del programa en *Bench Application*. De acuerdo a esas variables se carga el formulario, por ejemplo, con el número de *checkbox* correspondiente, es importante mencionar que el archivo de *ProductName.txt* ha sido creado con el fin de que en caso de que hubiera una nueva configuración el usuario pueda fácilmente añadirla por medio de ese archivo, esto para proporcionar capacidad de expansión.

Si no se encuentra coincidencia entre lo que había sido inicialmente leído proveniente de *Bench Application* y lo que hay almacenado en ese archivo el sistema dará un mensaje de error informando esta situación. Si encuentra la coincidencia en el arreglo pero no encuentra el archivo de información referente a ese arreglo el sistema dará opción al usuario de configurarla de una vez cargando el formulario con los espacios de información vacíos.

Si encuentra el archivo con la información referente al arreglo, lee esa información, añade los *checkbox* al formulario y muestra toda la información en el formulario (ver figura 12). Nótese que los *checkbox* son habilitados o deshabilitados, esto de acuerdo a la máscara dependiendo de si hay o no unidad en el *socket*

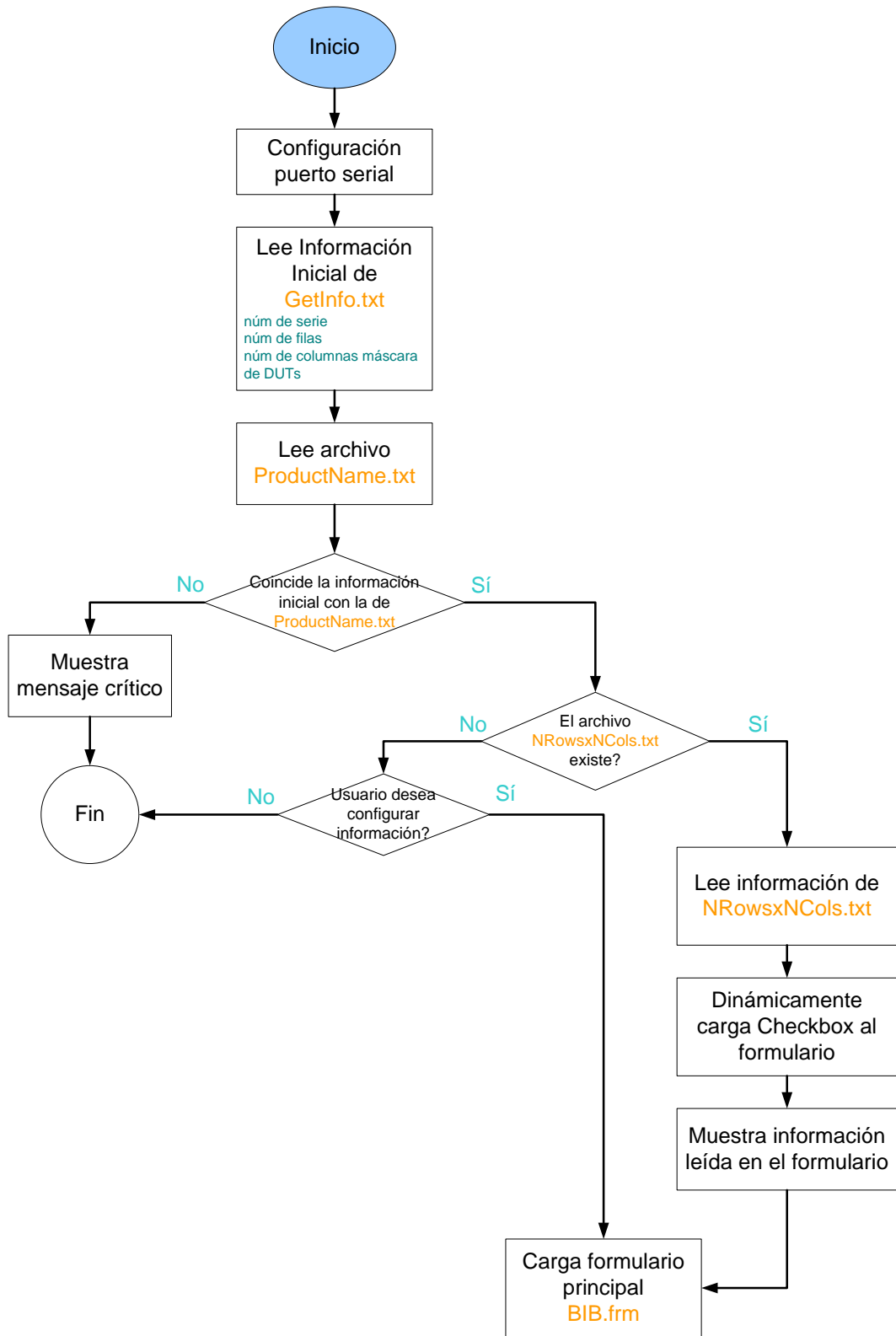


Figura 11 Diagrama de flujo de precarga del formulario principal

BIB Set Up

BIB ID: LCD00133

BIB Dimensions

BIB Socket Pitch (mm)

(x): 12

(y): 15

BIB Socket Offset (mm)

(x): 45

(y): 87

Get measures

View BIB Image Save Information

Select unids to be tested:

Continue...

Figura 12 Formulario principal de la aplicación VB6 (BIB.frm)

En el formulario principal (Figura 12) se pueden hacer varias tareas, tal y como se muestra en la Figura 13: se puede cargar el formulario para el movimiento manual (el cual será detallado más adelante), se puede acceder a la imagen de video que da la webcam, se puede guardar la información de las casillas de texto y además se puede iniciar con lo que es la ejecución de programas patrones y el envío de comandos

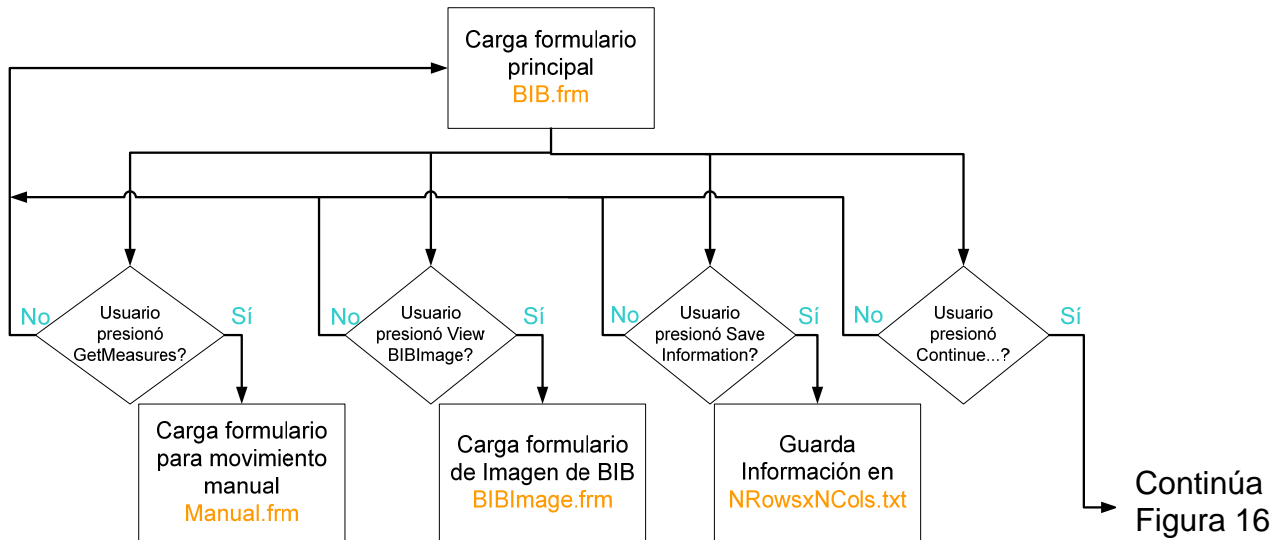


Figura 13 Diagrama de flujo de eventos del formulario principal

El movimiento manual fue diseñado básicamente con el fin de facilitar las mediciones del *offset* una vez montado el sistema mecánico. La idea es ir presionando los botones direccionales y que el sistema se vaya moviendo de acuerdo a eso hasta alcanzar la primera unidad, punto en el cual tenemos las medidas del *offset*. Como se observa en la Figura 14, se cuenta con varias funcionalidades en este formulario: con el botón *Back to Inicial Position* el sistema envía un comando para regresar a la posición inicial, con el botón *OK* las mediciones obtenidas son transferidas al formulario principal, con *Cancel* se cierra el formulario y con los direccionales se envían comandos de posicionamiento por el puerto serie. Es importante también señalar que las posiciones son evaluadas antes de enviar cada comando esto con el fin de evitar que el sistema trate de salirse de los límites

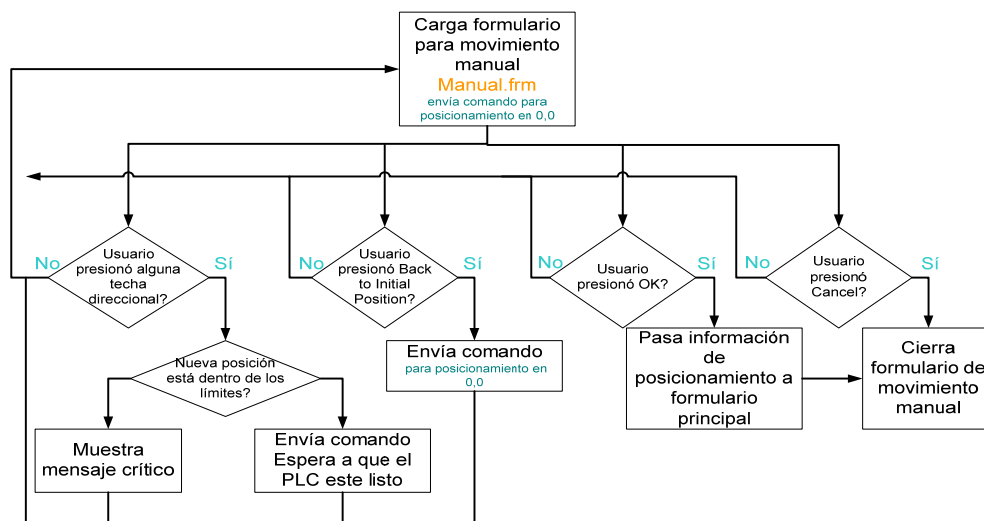


Figura 14 Diagrama de flujo de eventos del formulario para movimiento manual

En la Figura 15 se muestra el formulario para el movimiento manual

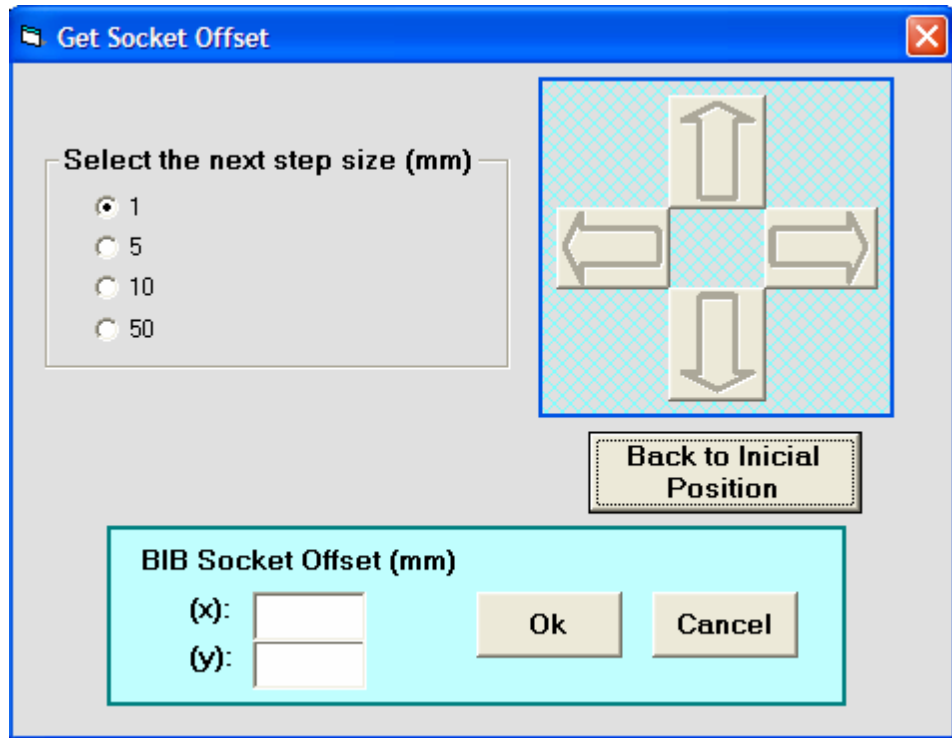


Figura 15 Formulario para el movimiento manual (Manual.frm)

Como se observa, también se cuenta con botones de opción con los cuales se puede escoger el tamaño del paso de los movimientos

En la Figura 16 se muestra todo lo que ocurre cuando el botón *Continue...* es presionado.

Se programó una función que determinara de la manera más efectiva los movimientos a realizar. Esta manera es explicada a continuación:

- siempre la cabeza 1 se posicionará sobre la última fila, su posición en x será manual y en y estará en función al de la otra cabeza (están en el mismo eje unidas)
- la cabeza 2 debe poder posicionarse sobre la primera o segunda fila por lo cual su posición en x es incluida en cada comando junto con el de y
- el barrido se realiza siempre de abajo hacia arriba y de derecha a izquierda
- si en una determinada fila ha sido seleccionado para ser probado el último *DUT* la cabeza se coloca ahí y si además alguno de los

- otros de la misma fila han sido seleccionados las dos cabezas harán contacto con el *DUT* en el mismo comando
- si los 3 *DUTs* han sido seleccionados de una misma fila primero se atenderán dos y luego uno solo

Esto indica lo que se debe incluir en cada comando: cuál (es) cabezas se utilizarán en ese movimiento, distancia en y y distancia en x

Viene de
Figura 13

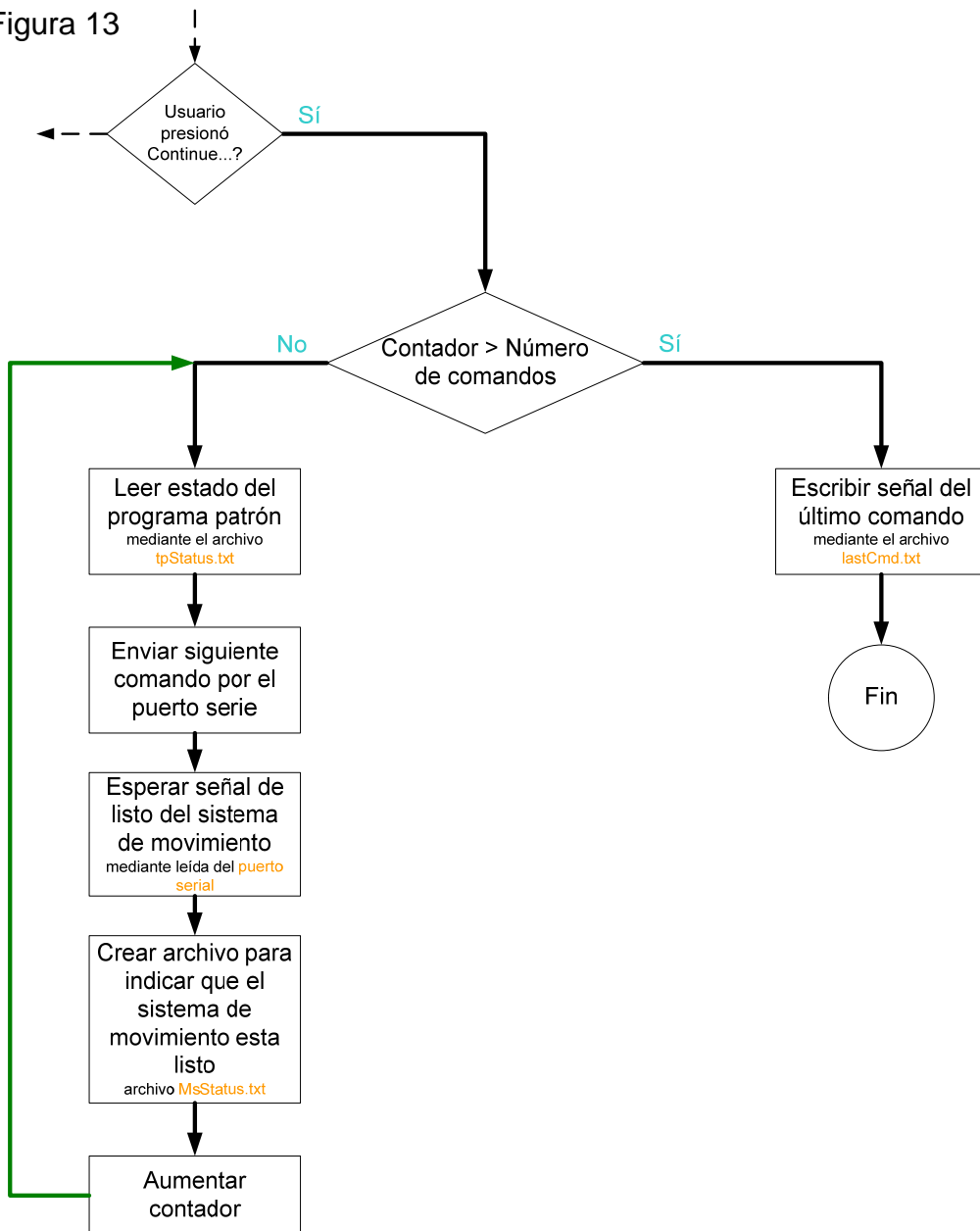


Figura 16 Diagrama de flujo para envío de comandos general

Volviendo a la Figura 16, se desarrolló un método que determinara y además contara los comandos a enviar, este valor determina el número de veces que se ejecuta todo el procedimiento de envío de comandos.

Como se verá más adelante cada vez que el programa de pruebas está listo para iniciar creará un archivo llamado **tpStatus.txt** de modo que lo primero que debe hacerse es esperar a que ese archivo exista, de no ser así implica que por alguna razón aun el programa de pruebas no puede ser ejecutado. En la Figura 17 se muestra el diagrama de flujo para estados de espera por existencia de un archivo. Una vez que se encontró el archivo es borrado

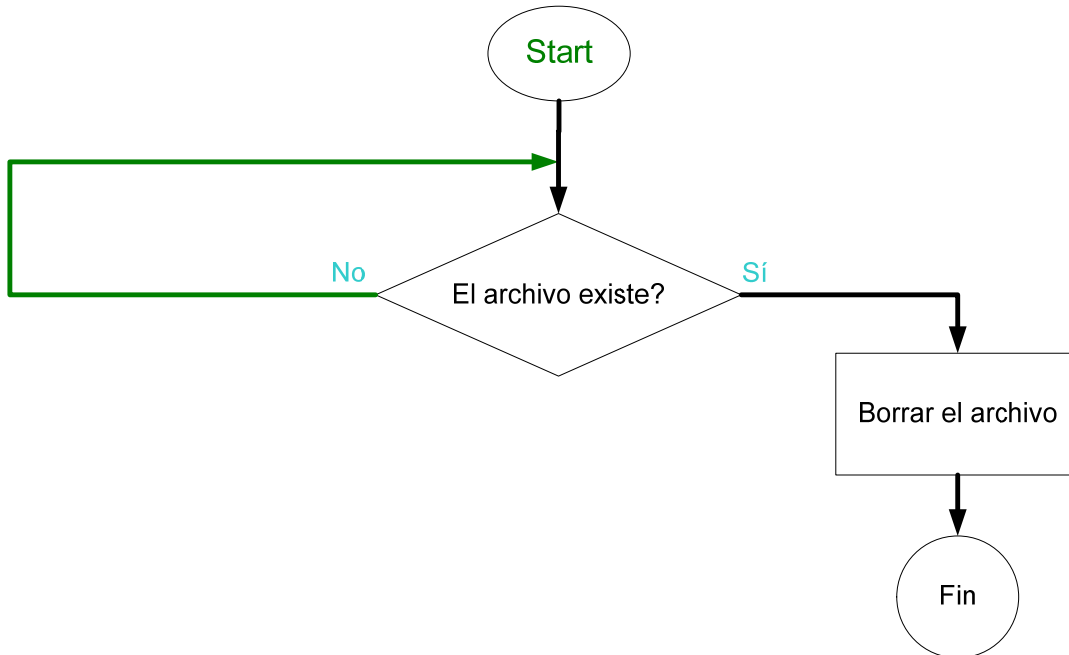


Figura 17 Diagrama de flujo para estados de espera basados en existencia de archivos

Cuando ya fue encontrado ese archivo se procede a enviar el comando por el puerto serie y de inmediato se cae en un estado en el que espera por recibir una señal del PLC que indica que el se ha posicionado y que por lo tanto esta listo. Una vez recibida esta señal se crea un archivo que indique al *Bench Application* que puede *iniciar* a ejecutar el programa de pruebas. Una vez terminado el procedimiento, guante el último comando se escribe otro archivo con el fin de que el programa de prueba pueda salir del ciclo y continuar con el resto de sus funciones

Por otra parte en *Bench Application* fue necesario agregar algún código para alcanzar lo requerimientos.
Primero en la inicialización se escribe un archivo con variables iniciales (Figura 18)

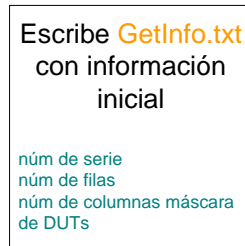


Figura 18 Inicialización del programa en Bench Application

En la Figura 19 se muestra el flujo de lo que se realiza para poder ejecutar los programas de pruebas dentro de un ciclo para movilizarse por todas las unidades. Básicamente monitorea que el sistema de movimiento esta correctamente posicionado, entonces y solo entonces ejecuta el programa, una vez terminado crea un archivo informado que está listo para iniciar otra vez. Además, se debe monitorear si ese fue el último comando para de una vez salirse del ciclo

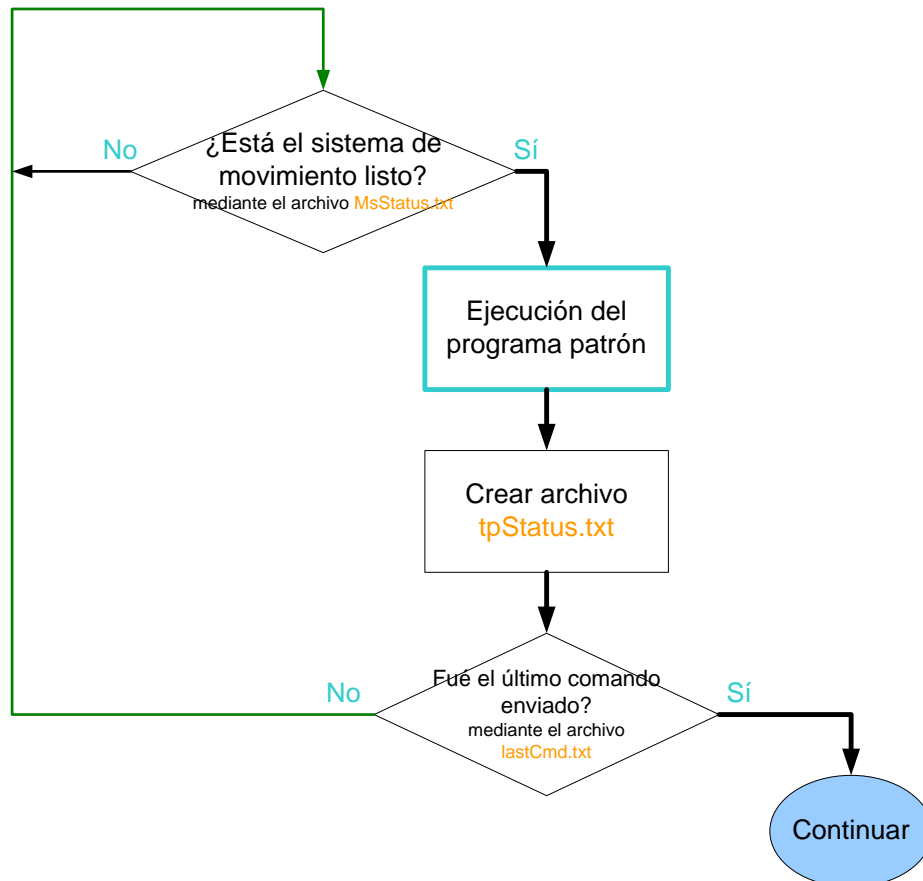


Figura 19 Diagrama de flujo al correr los programas patrón en Bench Application

Capítulo 6: Análisis de Resultados

6.1 Resultados y análisis

6.1.1 Software

A continuación se explicará paso a paso un ejemplo de funcionamiento.

El archivo de variables iniciales se muestra en la Figura 20, este archivo es creado por *Bench Application* durante la inicialización de la ejecución de las pruebas. El significado de cada línea está especificado en la misma figura. Es importante mencionar que el número de serie es necesario porque, de acuerdo a él (tercera letra) se puede determinar el arreglo de filas y columnas que corresponde. El leer éstos valores de la memoria estática de la *BIB* no es suficiente debido a que existe una excepción: en la *BIB* de 2x6 el número de filas está grabado como 3 y esto debe corregirse

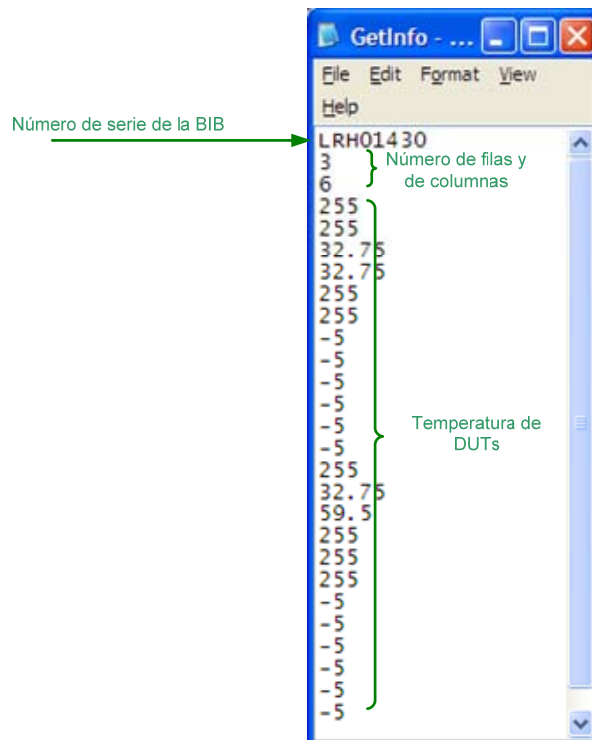


Figura 20 Ejemplo de archivo de variables iniciales (GetInfo.txt)

Una vez leído ese archivo se procede a comparar los valores leídos con los arreglos que se deberían obtener, esto es, comparar con el archivo ProductName.txt (Figura 21)

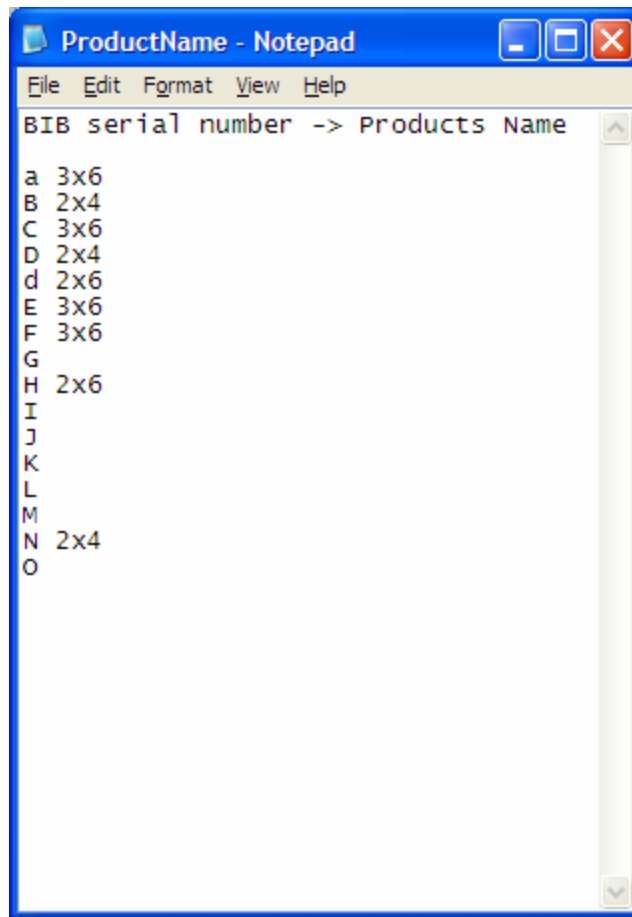


Figura 21 Archivo de comparación de arreglos del BIB²⁰

Como se observa lo que debe escribirse aquí es la tercera letra del número de serie de la *BIB* (mayúscula o minúscula) seguido por el arreglo de filas y columnas que corresponde. Este archivo se creó con el fin de permitir crecimiento en caso de que fuese requerido en el futuro.

Una vez procesada toda esa información se procede finalmente a cargar el fichero correspondiente. En este caso Figura 22. Nótese que los *checkboxs* son deshabilitados (según información del archivo GetInfo.txt) dependiendo de si hay o no *DUT* en el *socket* esto con el fin de evitar que el usuario por error seleccione unidades por probar que no están presentes.

²⁰ Archivo de acuerdo a nomenclatura de la empresa

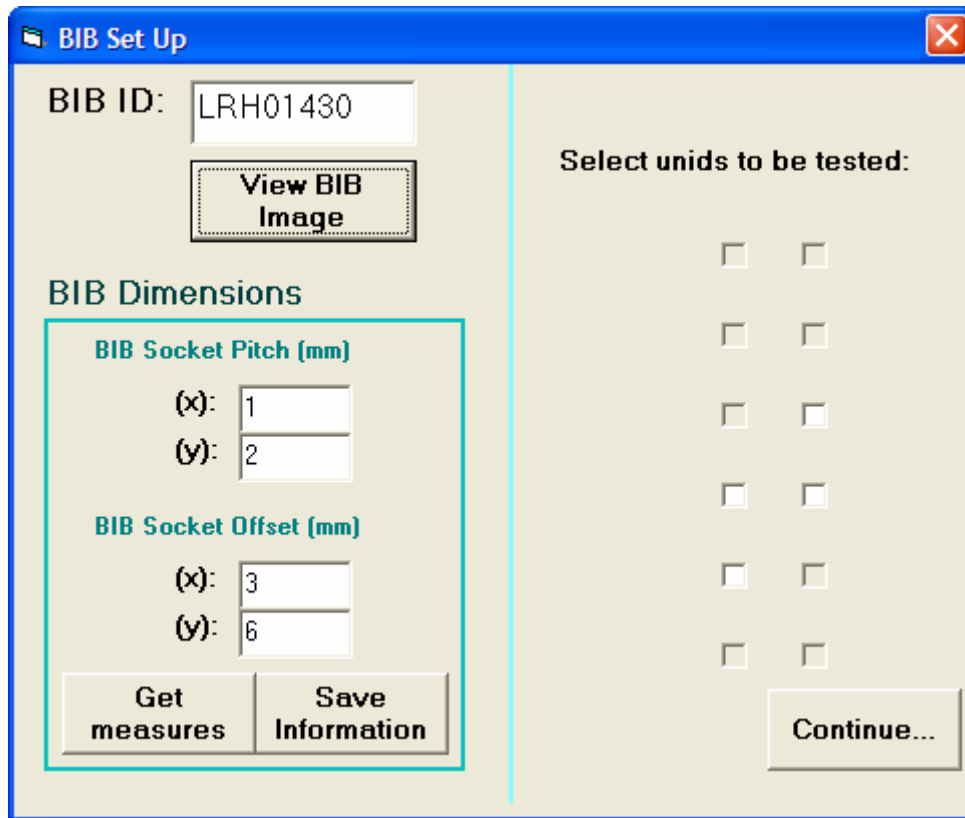


Figura 22 Fichero principal para el ejemplo

Por otra parte las medidas mostradas a la izquierda son descargadas del archivo con el arreglo correspondiente, en este caso 26.txt (Figura 24).

- BIB socket Pitch corresponde a la separación entre unidades.
- BIB socket Offset corresponde a las distancias de compensado necesarias para alcanzar desde el sistema mecánico el primer socket

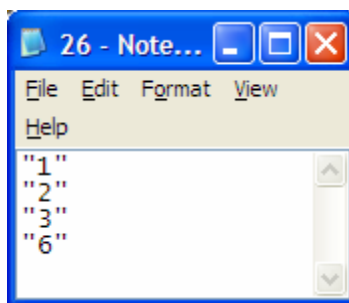


Figura 23 Archivo de información de mediciones del arreglo (3x6.txt)

En la Figura 22, al lado derecho, el usuario selecciona las unidades que desea probar y presiona *Continue...* para iniciar el algoritmo.

Para efectos del ejemplo, se unieron los pines de transmisión y recepción del puerto serial para simular la retroalimentación recibida por el PLC (en este caso el comando de retroalimentación que envía el PLC que indica que está listo es el "99").

Se utiliza en este experimento además un monitor de puerto serie con el fin de mostrar lo que sucede ahí.

A continuación se muestran los resultados para el experimento

Port opened by process "vb6.exe" (PID: 2080)

4D 4F 56 20 30 30 20 30 30 30 20 30 30 30 39 39 **MOV 00 000 000**

Port closed

Port opened by process "vb6.exe" (PID: 2080)

39 39 **99**

Port closed

Port opened by process "vb6.exe" (PID: 2080)

4D 4F 56 20 31 30 20 30 30 38 20 30 30 30 **MOV 10 008 000**

Port closed

Port opened by process "vb6.exe" (PID: 2080)

39 39 **99**

Port closed

Port opened by process "vb6.exe" (PID: 2080)

4D 4F 56 20 31 31 20 30 31 30 20 30 30 33 **MOV 11 010 003**

Port closed

Port opened by process "vb6.exe" (PID: 2080)

39 39 **99**

Port closed

Port opened by process "vb6.exe" (PID: 2080)

4D 4F 56 20 30 31 20 30 31 32 20 30 30 33 **MOV 01 012 003**

Port closed

Port opened by process "vb6.exe" (PID: 2080)

39 39 **99**

Port closed

Lo primero que sucede es que el sistema envía un comando que indica que debe irse a la posición 0,0 esto con el fin de estar seguro de que el sistema esta en *home* al iniciar y espera a que el PLC envíe un comando (99) informando que esta listo para iniciar, una vez recibido ese comando internamente el programa espera a que *Bench Application* escriba el archivo tpStatus.done informando que esta listo para iniciar una vez encontrado ese archivo el programa envía el primer comando y nuevamente espera a que el sistema de movimiento informe que está listo (esto implica que se ha posicionado donde se requiere) esto constituye la señal que *Bench Application* monitorea cada vez que va a iniciar a ejecutar un programa, con el fin de asegurarse que las cabezas están listas. Dada esta señal se empieza a ejecutar el programa de prueba en *Bench Application*. Al finalizar, éste crea un archivo informando que acaba de finalizar la ejecución y que, por lo tanto, esta listo para volver a ejecutarlo si fuera necesario. Esta señal es monitoreada por la aplicación de Visual Basic y el proceso inicia nuevamente.

Los comandos involucran distancias en *mm* de posicionamiento, cada valor ya es un valor final, esto significa que ya se han tomado en cuenta las medidas correspondientes de *pitch* y de *offset*. Son mostrados en valores ASCII para facilitar la comprensión. Todos los comandos son ajustados para cumplir con un estándar: el comando "MOV" seguido por dos dígitos que indican las cabezas térmicas involucradas en ese comando, luego 3 dígitos que corresponden al desplazamiento en y y 3 dígitos que corresponden al desplazamiento en x

El programa está diseñado de tal forma que la secuencia de comandos es la más eficiente. Primero se prueban tratando de usar las 2 cabezas térmicas al mismo tiempo, como el primer caso luego se prueba la unidad del centro por la cabeza térmica 2 si así se requiriera y luego pasa a la siguiente columna. Esta metodología de probar las unidades ayuda a tener un buen rendimiento en cuanto a tiempo ya que se promueve que las cabezas se posicionen al mismo tiempo sobre las unidades siempre que se pueda

Por otra parte es indispensable que las mediciones involucradas sean lo más acertadas posibles, por esta razón fue que se incluyó la posibilidad de movimiento manual. La idea es mover poco a poco las cabezas hasta alcanzar con la cabeza número 2 la primera unidad, esas mediciones corresponderán a los valores de *offset*. A continuación se muestran los resultados de esa sección.

En la Figura 25 se muestra el formulario correspondiente para el ejemplo

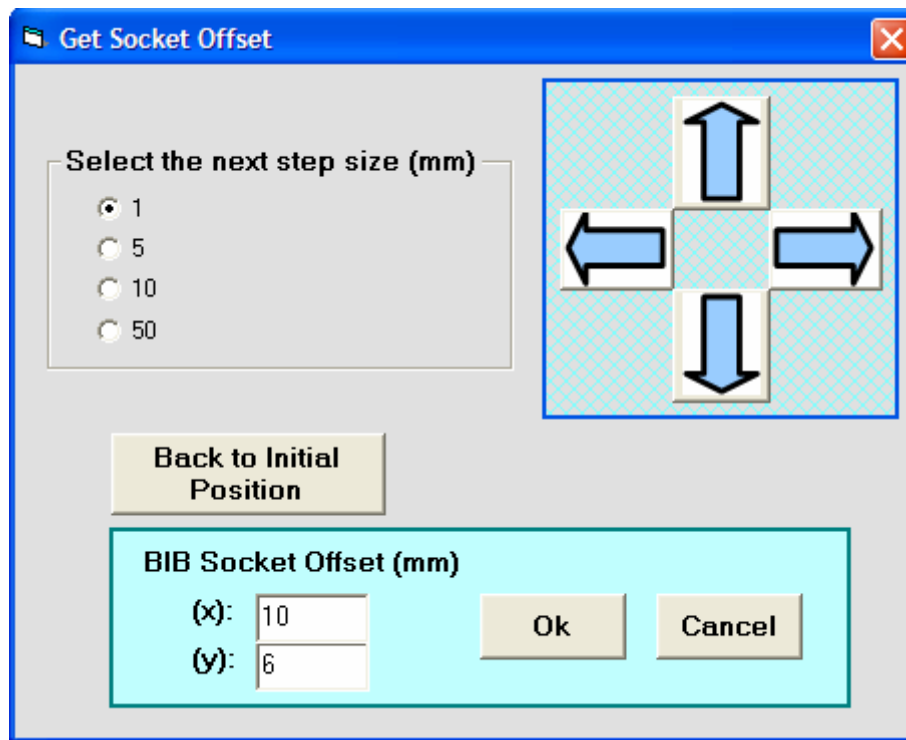


Figura 24 Formulario para el movimiento manual para el ejemplo

A continuación se muestran los resultados obtenidos

```
Port opened by process "MovSystem.exe" (PID: 3288)
4D 4F 56 20 30 30 20 30 30 30 20 30 30 30 39 39  MOV 00 000 000

Port closed

Port opened by process "MovSystem.exe" (PID: 3288)
39 39 99

Port closed

Port opened by process "MovSystem.exe" (PID: 3288)
4D 4F 56 20 30 30 20 30 30 35 20 30 30 30  MOV 00 005 000

Port closed

Port opened by process "MovSystem.exe" (PID: 3288)
39 39 99

Port closed

Port opened by process "MovSystem.exe" (PID: 3288)
4D 4F 56 20 30 30 20 30 30 35 20 30 31 30  MOV 00 005 010

Port closed

Port opened by process "MovSystem.exe" (PID: 3288)
39 39 99

Port closed

Port opened by process "MovSystem.exe" (PID: 3288)
4D 4F 56 20 30 30 20 30 30 36 20 30 31 30  MOV 00 006 010

Port closed

Port opened by process "MovSystem.exe" (PID: 3288)
39 39 99

Port closed
```

Al acceder a esta modalidad lo primero que se pide es que el sistema de movimiento se coloque en la posición 0,0. Se puede seleccionar por medio de los botones de opción la magnitud en *mm* de los pasos a dar. Cada vez que se toca uno de los botones direccionales se envía el comando correspondiente y se espera que el PLC de la señal de retroalimentación, esta espera se asegura deshabilitando los botones direccionales hasta que se reciba esa retroalimentación. Nunca las cabezas harán contacto con las unidades en esta modalidad.

Por último, en la Figura 26 se muestra una fotografía del *BIB* tomada desde la webcam a cual ha sido accesada desde la aplicación de VB6 desarrollado, cumpliendo con el último objetivo específico de este proyecto

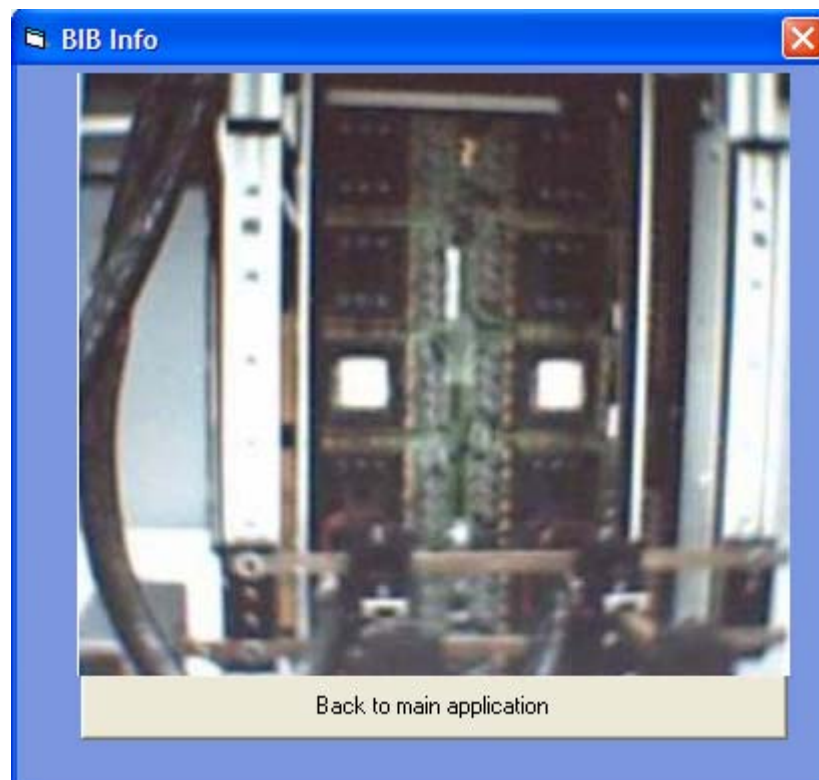


Figura 25 Imagen del BIB desde la Webcam

6.1.2 Hardware

Actualmente se están llevando a cabo los trámites correspondientes para contratar a un proveedor con el fin de que éste implemente el sistema mecánico involucrado en este proyecto.

En la Figura 6 se muestra el diagrama del sistema mecánico a implementar.

Capítulo 7: Conclusiones y recomendaciones

7.1 Conclusiones

1. Se logró el intercambio de información entre *Bench Aplicacion* y la aplicación desarrolla en VB6
2. Los tiempos de espera son controlados mediante el monitoreo de la existencia o no de archivos que funcionan como banderas de estados
3. Se implementó el algoritmo de posicionamiento que mejor rendimiento alcanza
4. El movimiento manual facilita la obtención de las medidas de *offset* necesarias
5. El control *exVidCap.ocx* permitió manejar la imagen proveniente de la webcam
6. Se cuenta con algoritmos de verificación de mediciones para evitar posicionamientos fuera de los límites
7. La herramienta desarrollada permite añadir nuevas configuraciones fácilmente
8. El sistema de software esta en capacidad de manejar cualquier arreglo de filas y columnas (tomando el de 3x6 como máximo)
9. El programa está diseñado de tal forma que la información de medidas no se pierden y son cargadas fácilmente cada vez que se requiera
10. Se logró entrelazar la aplicación con la ya existente sin afectar negativamente el funcionamiento de ambas

7.2 Recomendaciones

1. Quizá una de las mejoras más importantes que se pueden realizar a este proyecto es integrarlo a la ethernet de la empresa. Esto con el fin de mejorar la accesibilidad a éste.
2. Migrar la aplicación a una plataforma más reciente como lo son .Net de Microsoft o Java de Sun Micro Systems con el fin de aportar nuevas capacidades (bases de datos, facilitar una posible migración a web de la aplicación y otros)

Bibliografía

- Cornell, Gary. Learn Microsoft Visual Basic scripting edition now. 4a Edition U.S.A, McGrawhill, 2000.
- Hillier, Scot. Inside MS Visual Basic Scripting Edition. 2a Edition U.S.A, McPearson, 2002.
- Microsoft. Microsoft Visual Basic 6.0 Developers Workshop. 2a Edition U.S.A, Pearson Education, 2002
- Zúñiga, Mauricio. Diseño e integración de una herramienta de software multiproducto capaz de modificar la temperatura en la superficie de los microprocesadores. Tesis. Instituto Tecnológico de Costa Rica, Enero de 2007.
- <http://www.intel.com/costarica/encostarica.htm?iid=CRHPAGE>
 - Información de la empresa en que se desarrolla el proyecto
- http://www.elquille.info/VB/cursos_vb/BASICO/indice.htm
 - Tutorial de VB6
- <http://www.depeca.uah.es/alcabot/seminario2006/Trabajos/JoseManuelMurciaBarba.pdf>
 - Información referente al puerto serial y al protocolo RS232
- http://www.martin2k.co.uk/vb6/tips/vb_20.php
 - Información de VB6
- http://es.wikipedia.org/wiki/Visual_Basic
 - Información de VB6

Apéndices

A.1 Glosario, abreviaturas y simbología

BI: burn in, aceleración de envejecimiento

BIB: burn in board

DUT: device under test

Bench: máquina experimental para el proceso de Burn-In

Chiller: controlador de temperatura de las cabezas térmicas

Driver: controlador de los servomotores

Patrones de prueba: programas ejecutados en las unidades con el fin de detectar posibles fallas en éstos

TAP: Puerto de acceso de pruebas

DLL: Dynamic Linking Library

RS232: protocolo de transmisión de datos

EzVidCap: control para utilización de webcam en visual Basic 6.0

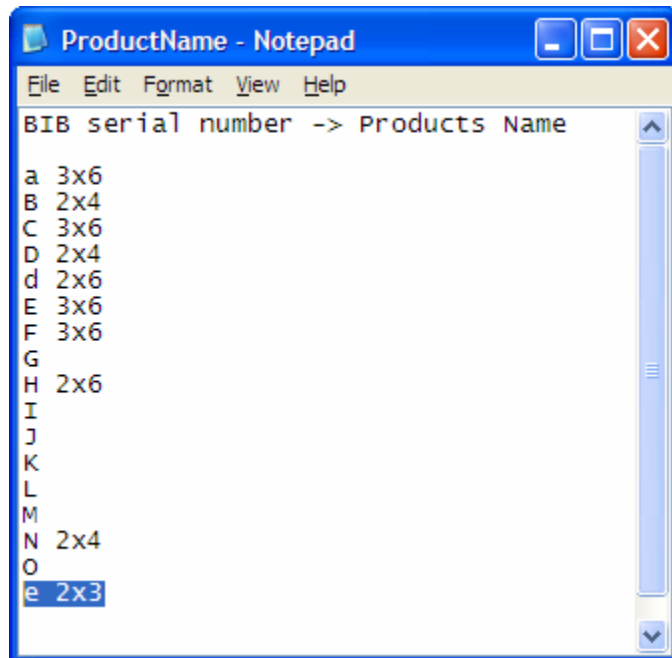
A.2 Manual(es) de usuario

A.2.1 Instalación

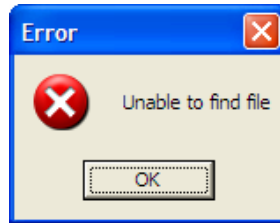
1. Copiar el archivo `exVidCap.oxc` en el directorio `c:\WINDOWS\system`
2. Registrar el control `exVidCap`
 - 2.1 Presione `Stara/run`
 - 2.2 Escriba `"cmd"`
 - 2.3 En el `command prompt` escriba `"cd c:\WINDOWS\system"`
 - 2.4 Escriba `"regsvr32 exVidCap.oxc"`
 - 2.5 Recibirá un mensaje confirmando que fue bien registrado.
 - 2.6 Escriba `"exit"` para salir del `command prompt`

A.2.2 Configuración de información para un nuevo arreglo de filas y columnas

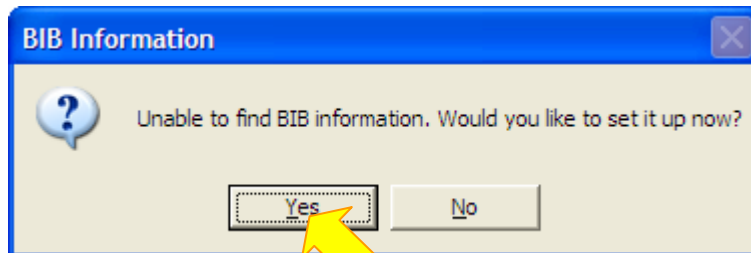
1. Primero debe modificarse el archivo `ProductName.txt` agregándole una línea con la tercer letra del número de serie de la BIB con la nueva configuración seguida del arreglo de filas y columnas que prosee



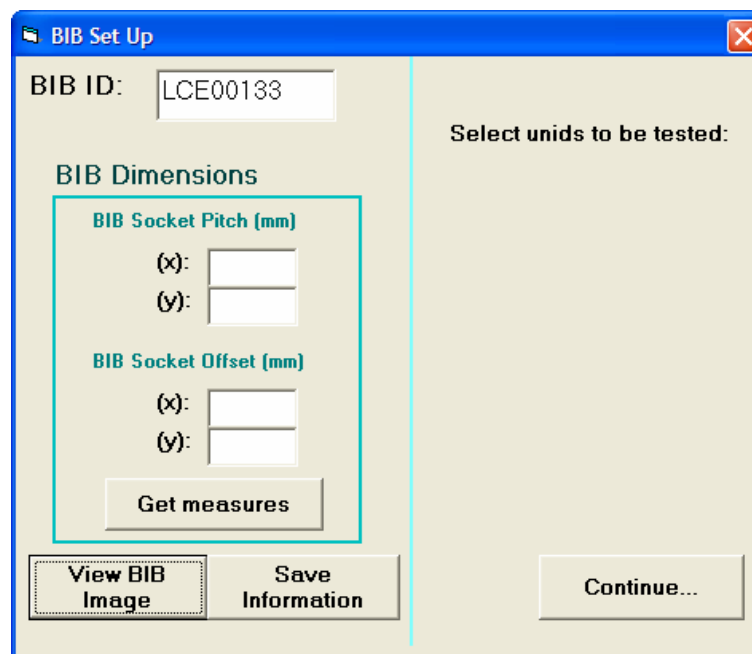
2. Cuando se intenta correr el ejecutable este mostrará el siguiente mensaje



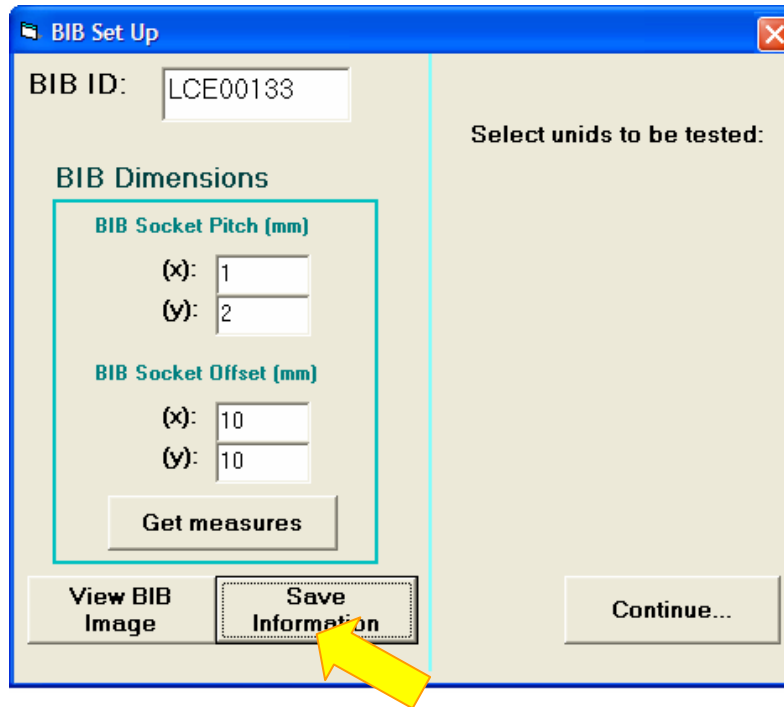
Seguido de



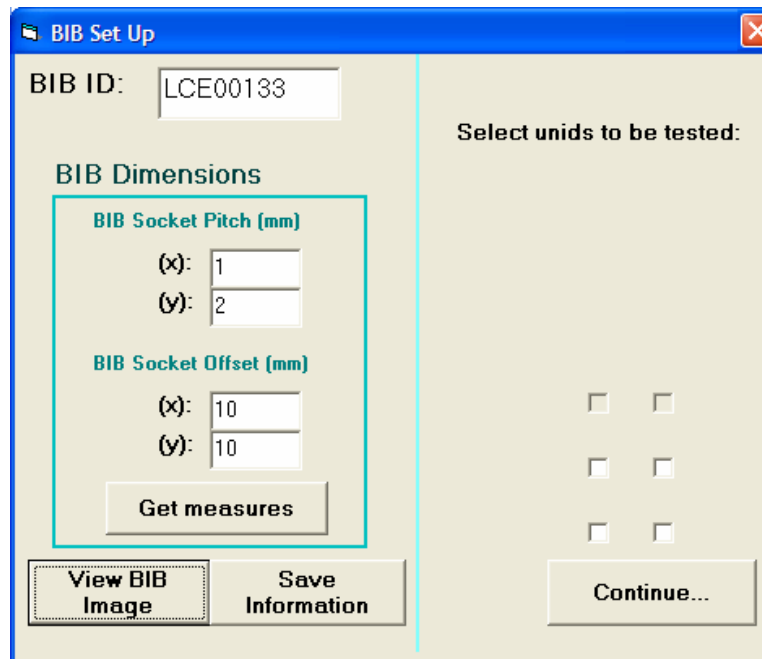
Si desea configurar la información clic on yes y la siguiente ventana aparecerá



3. Digite la información correspondiente el las casillas. Puede utilizar el botón de *Get measures* para obtener fácilmente las medidas de *offset*. Presione *Save Information* para guardar la información

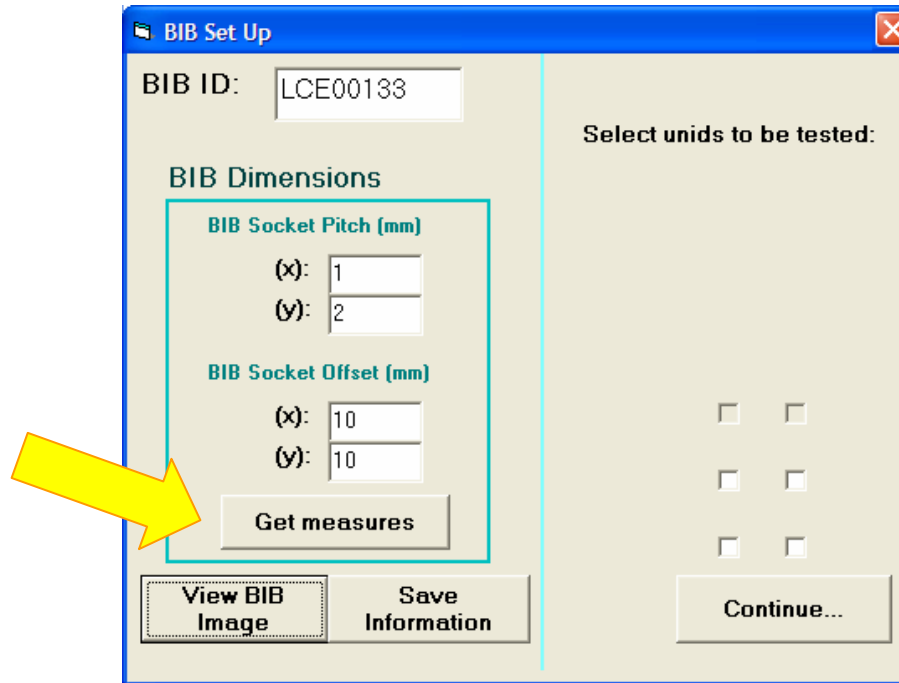


4. Una vez guardada esta información, abra la aplicación nuevamente y podrá confirmar que la información fue guardada satisfactoriamente

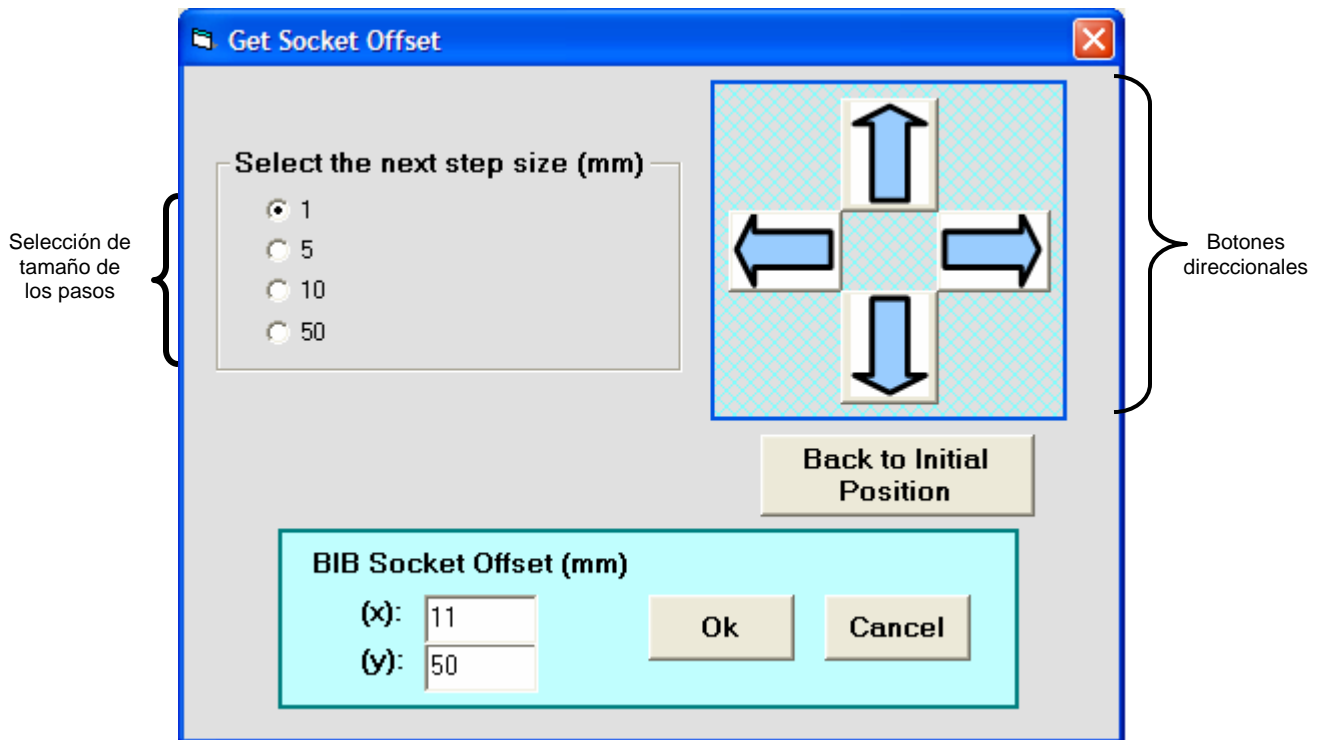


A.2.3 Movimiento manual para determinar medidas de offset

1. Ejecute la aplicación y presione el botón *Get measures*



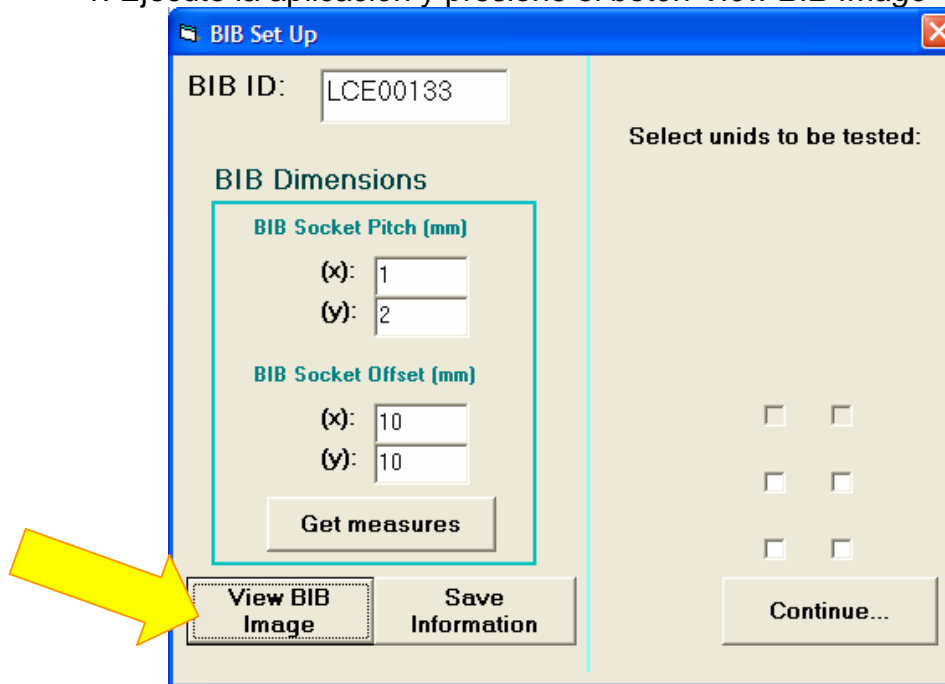
2. En el nuevo formulario presione las teclas direccionales (con flechas) hasta lograr que la cabeza térmica 2 se posicione sobre la primera unidad del *BIB*. Nótese que es posible seleccionar el tamaño de los pasos usando los botones de opción de la izquierda del formulario



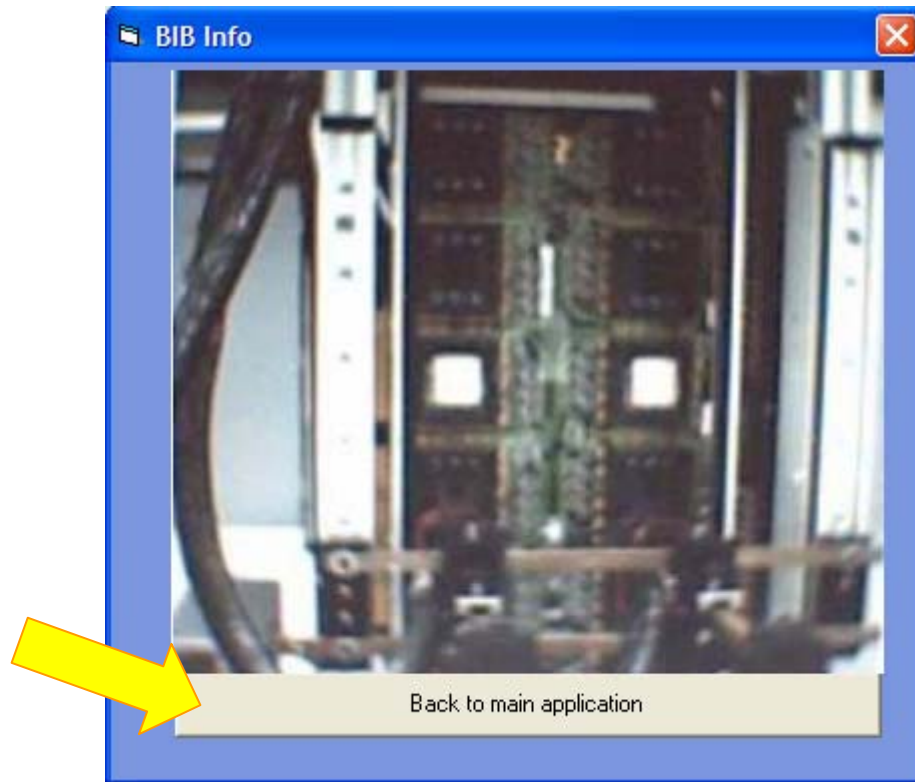
3. En el recuadro de la parte inferior se muestra la distancia acumulada hasta el momento, la cual una vez colocada la cabeza sobre la primera unidad será la distancia de *offset*. Si se requiere volver a la posición inicial presione el botón *Back to Inicial Position*
4. Una vez que el proceso sea concluido presione *Ok* para que las medidas de *offset* sean transferidas al formulario principal
5. En el formulario principal presione *Save Information* para guardar los cambios

A.2.4 Observar imagen proveniente de la webcam

1. Ejecute la aplicación y presione el botón *View BIB Image*



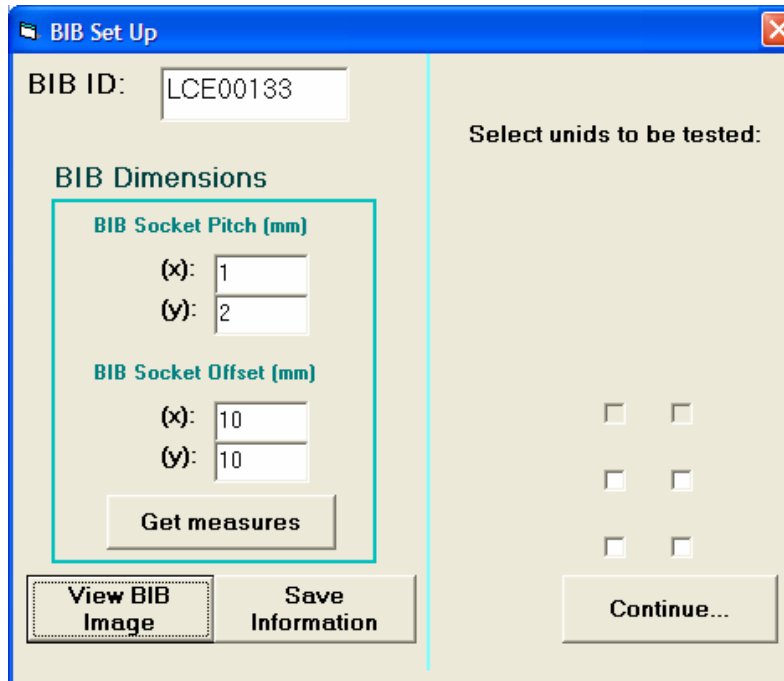
2. Podrá observar la imagen en el formulario que se abre. Presione *Back to Main Application* cuando desee cerrar esa ventana



A.2.5 Configurar unidades a probar e iniciar procedimiento

1. Ejecute la aplicación²¹

²¹ Debe ser ejecutada después de iniciado el proceso de la aplicación del *bench* debido a que utiliza información de precarga



2. Al lado derecho se desplegara un arreglo de *checkboxes* que corresponde al del *BIB* que se está usando. Los *checkboxes* correspondientes a *sockets* vacíos estarán deshabilitados. Seleccione los *checkboxes* de las unidades que se desean probar

3. Finalmente presione *Continue...* para iniciar el proceso

BIB Set Up ✖

BIB ID:

BIB Dimensions

BIB Socket Pitch (mm)

(x):

(y):


BIB Socket Offset (mm)

(x):

(y):

Select units to be tested:

<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>



A.3 Información sobre la empresa/institución

HOJA DE INFORMACIÓN ACTUALIZADA PROYECTO DE GRADUACIÓN

Información del estudiante:

Nombre: Mariel Monge Sequeira

Cédula: 6 -336 -993 **Carné ITCR:** 200219137

Dirección de su residencia en época lectiva: 50 oeste KFC Barrio la California, apartamentos amarillos portón verde, San José

Dirección de su residencia en época no lectiva: 50 oeste KFC Barrio la California, apartamentos amarillos portón verde, San José

Teléfono en época lectiva: 3970360 **Teléfono época no lectiva:** 3970360

Email: mmonge80@yahoo.com.mx, mariel.monge@intel.com

Información del Proyecto

Duración en meses: 6

Nombre del Proyecto: Diseño e implementación del control de movimiento para el sistema térmico del modulo experimental de Burn-In

Area del Proyecto: automatización, electrónica de potencia, sistemas digitales, programación

Profesor Asesor: Francisco Navarro

Información de la Empresa

Nombre: Componentes Intel de Costa Rica

Zona: Heredia, Belén, La Rivera

Dirección en la cual se ubica Ud. dentro de la empresa : Cubículo 18 CR1

Su teléfono en la empresa 298-7292

Actividad Principal de la empresa: Ensamble y pruebas de microprocesadores y chipsets

Información del encargado y asesor en la empresa:

Nombre: David González Mayorga

Puesto que ocupa: Ingeniero de producto y desarrollo en el área de calidad

Departamento: Calidad y Fiabilidad

Profesión: Ingeniero Electrónico

Grado académico: Bachiller

Teléfono: 2988208

Email: david.m.gonzalez.mayorga@intel.com

Anexos

Anexo B.1 Hoja de datos del XLE OCS Model: HE-XE105

MAN0810-04
Specifications / Installation

15 JUL 2006

PAGE 1
HE-XE105



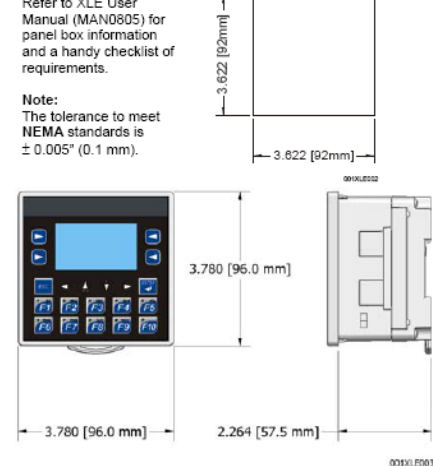
XLE OCS Model: HE-XE105
12 Digital DC Inputs / 12 Digital DC Outputs
2 Analog Inputs (High Resolution)
2 Analog Outputs

Want More Information?
 To download the XLE User Manual (MAN0805), refer to *Technical Support* in this document.

1 Specifications			
HE-XE105 Specifications			
Digital DC Inputs		Digital DC Outputs	
Inputs per Module	12 including 4 configurable HSC inputs	Outputs per Module	12 including 2 configurable PWM outputs
Commons per Module	1	Commons per Module	1
Input Voltage Range	12 VDC / 24 VDC	Output Type	Sourcing / 10 K Pull-Down
Absolute Max. Voltage	35 VDC Max.	Absolute Max. Voltage	28 VDC Max.
Input Impedance	10 kΩ	Output Protection	Short Circuit
Input Current	Positive Logic Negative Logic	Max. Output Current per point	0.5 A
Upper Threshold	0.8 mA	Max. Total Current	4 A Continuous
Lower Threshold	0.3 mA	Max. Output Supply Voltage	30 VDC
Max Upper Threshold	8 VDC	Minimum Output Supply Voltage	10 VDC
Min Lower Threshold	3 VDC	Max. Voltage Drop at Rated Current	0.25 VDC
OFF to ON Response	1 ms	Max. Inrush Current	650 mA per channel
ON to OFF Response	1 ms	Min. Load	None
HSC Max. Switching Rate	10 kHz	OFF to ON Response	1 ms
		ON to OFF Response	1 ms
Analog Inputs, High Resolution		Analog Outputs	
Number of Channels	2	Number of Channels	2
Input Ranges (Selectable)	0 - 10 VDC 0 - 20 mA 4 - 20 mA PT100 RTD, and J, K, N, T, E, R, S, B Thermocouples	Output Ranges	0-10 VDC, 0-20 mA
Safe input voltage range	10 VDC: -0.5 V to +15 V 20 mA: -0.5 V to +6 V RTD / T/C: ±24 VDC	Nominal Resolution	12 Bits
Nominal Resolution	14 Bits	Update rate	Once per PLC scan
Input Impedance (Clamped @ -0.5 VDC to 12 VDC)	Current Mode: 100 Ω, 35mA Max. Continuous Voltage Mode: 500 kΩ, 35mA Max. Continuous	Minimum 10 V load	5 kΩ
%AI full scale	10 V, 20 mA, 100 mV: 32,000 counts full scale. RTD / T/C: 20 counts / °C	Maximum 20 mA load	500 Ω
Max. Over-Current	35 mA	Analog Outputs; Output Points Required	2
Open Thermocouple Detect Current	50 nA	Maximum Error at 25°C (excluding zero)	0.1%
Conversion Speed	All channels converted once per ladder scan	Additional error for temperatures other than 25°C	0.01% / °C
RTD Excitation Current	250 μA	General Specifications	
Thermocouple Common Mode Range	±10V	Required Power (Steady State)	130 mA @ 24 VDC
Converter Type	Successive Approximation	Required Power (Inrush)	30 A for 1 ms @ 24 VDC
Max. Error at 25°C (*excluding zero)	*4-20 mA 0.10%* *0-20 mA 0.10%* *0-10 VDC 0.10%* RTD (PT100) ± 1.0 °C 0-100 mV 0.05% T/Cs To Be Determined	Primary Power Range	10 - 30 VDC
Additional error for temperatures other than 25°C	TBD	Operating Temperature	0° to 50° Celsius
		Relative Humidity	5 to 95% Non-condensing
		CE	See Compliance Table at http://www.heag.com/Support/compliance.htm
		UL	
		Filtering	160Hz hash (noise) filter 1-128 scan digital running average filter
		Terminal Type	Screw Type, 5 mm Removable
		Weight	12.5 oz. (354.36)

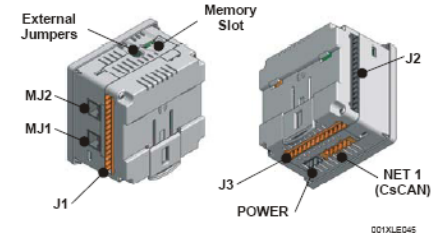
2 Panel Cut-Out and Dimensions

Note: Max. panel thickness: 5 mm.



3 Ports / Connectors / Cables

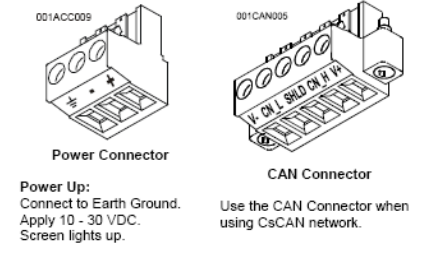
Note: The case of the XLE is black, but for clarity, it is shown in a lighter gray color.



To Remove Back Cover: Unscrew 4 screws located on the back of the unit. Lift lid.

CAUTION: Do not overtighten screws when screwing the lid back on.

I/O Jumpers: (Not Shown)
 I/O Jumpers (JP) are located internally. To access, remove back cover of unit.
 The I/O Jumpers, External Jumpers and Connectors (J1 - J3) are described in the *Wiring and Jumpers* section of this document.



Anexo B.2 Hoja de datos del MR-J3-A/B

MR-J3

Servo and Motion Control

Simple, easy, dynamic Ultimate positioning control

Common specifications MR-J3-A/B			10A 10B	20A 20B	40A 40B	60A 60B	70A 70B	100A 100B	200A 200B	350A 350B	500A 500B	700A 700B	
Power supply	Voltage / frequency [®]	A type	3-phase 200 – 230 V AC, 50 / 60 Hz; 1-phase 200 – 230 V AC, 50 / 60 Hz; 1-phase 200 – 240 V AC, 50 / 60 Hz					3-phase 200 – 230 V AC, 50 / 60 Hz					
		B type	3-phase 200 – 230 V AC, 50 / 60 Hz; 1-phase 230 V AC, 50 / 60 Hz										
	Permissible voltage fluctuation	A type	3-phase 200 – 230 V AC: 170 – 253 V AC, 1-phase 200 – 230 V AC: 170 – 253 V AC					3-phase 170 – 253 V AC					
		B type	3-phase 200 – 230 V AC: 170 – 253 V AC, 1-phase 230 V AC: 207 – 253 V AC										
	Permissible frequency fluctuation	± 5 % maximum											
Control circuit power supply	Voltage / frequency	1-phase 170 – 253 V AC, 50 / 60 Hz (± 5 % max.)											
	Power consumption (W)	30									45		
Interface power supply	A type	24 V DC ± 10 % (required current capacity: 300 mA)											
	B type	24 V DC ± 10 % (required current capacity: 150 mA)											
Control system	Sinusoidal PWM control / current control system												
Dynamic brake	Built-in												
Protective functions	Overcurrent shutdown, regeneration overvoltage shutdown, overload shutdown (electronic thermal), servo-motor overheat protection, encoder fault protection, regeneration fault protection, undervoltage / sudden power outage protection, overspeed protection, excess error protection.												
Structure	Self-cooling, open (IP00)						Fan-cooling, open (IP00)						
Environment	Ambient temperature	Operation: 0 – 55 °C (no freezing). Storage: -20 – 65 °C (no freezing)											
	Ambient humidity	Operation: 90 % RH max. (no condensation). Storage: 90 % RH max. (no condensation)											
	Elevation	1000 m or less above sea level											
	Oscillation	5.9 m/s ² (0.6 G) max.											
Weight [kg]		0.8	0.8	1.0	1.0	1.4	1.4	2.3	2.3	4.6	4.6	6.2	

[®] Rated output capacity and rated rotation speed of the servo motor used in combination with the servo amplifier are as indicated when using the power voltage and frequency listed. Output and speed cannot be guaranteed when the power supply voltage is less than specified.

Note: See technical catalogue or manual for specific model details.

Anexo B.3 Hoja de datos del servomotor de la serie HPE/HC

Planetary geared brushless servo motors

HPE/HC series

- High peak torque provides high acceleration & deceleration capability
- Capable of withstanding factor 2 emergency shock loads without damage
- Low backlash construction provides accurate positioning and long life
- High torsional rigidity for stable operation in high speed positioning.
- Low torque ripple for smooth operation in velocity servo applications
- Planetary construction for high torque to volume ratio
- High reliability and serviceability

Planetary geared servo motor dimensions:

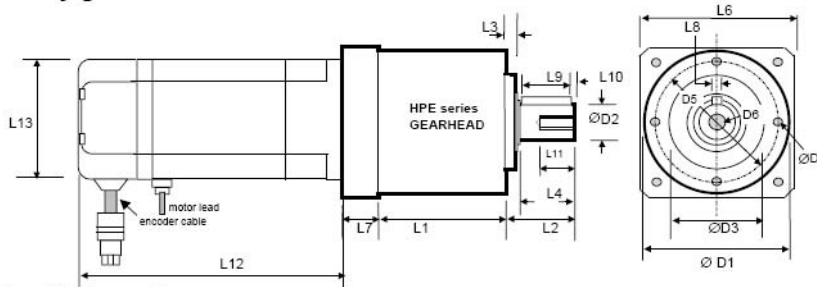


Table of Dimensions: mm

Gearhead		HPE50		HPE70		HPE90		
Motor Types		HC-PQ13	HC-PQ053	HC-PQ43	HC-PQ23	HC-PQ43		
Gearbox Version		HPE50S	HPE50D	HPE70S	HPE70D	HPE90D		
Gearbox Diameter	D1	50	50	70	70	90		
Output shaft diameter	D2	12 k6	12 k6	16 k6	16 k6	22 k6		
Gearbox mounting register	D3	35 h6	35 h6	52 h6	52 h6	68 h6		
Gearbox mounting holes	D4	4 x M48 deep		4 x M5 10 deep		4 x M8 12 deep		
mounting hole PCD	D5	44	44	62	62	80		
Shaft fixing bolt diameter	D6	tapped	M4	tapped	M5	M8		
Gearbox Length	L1	53	74.5	69	91.5	109		
Overall Output Shaft length	L2	24.5	24.5	36	36	46		
Gearbox register length	L3	4	4	5	5	5		
Free shaft length	L4	18	18	28	28	36		
Gearhead flange size	L6	50	50	70	70	90		
Adaptor length	L7	19	19	22	22	32		
Output shaft key width	L8	4 h9	4 h9	5 h9	5 h9	6 h9		
Output shaft Key length	L9	14	14	25	25	32		
Key distance to shaft end	L10	2	2	2	2	2		
Shaft fixing tapped length	L11	8	8	10	10	13		
Motor body length	L12	See motor data						
Motor body square	L13	40	40	60	60	60		

Performance of geared servo motors

Gearbox Model	Gear Ratio n:1	Motor Model	Rated Speed (rpm)	Rated Torque (Nm)	Peak Torque (Nm)	Maximum Emergency Torque (Nm)	Max. Backlash (arc. min.)	Assembly Mass (Kg)
HPE 50S	5	HC-PQ13	600	1.5	4.5	4.5	≤ 12	1.3
	10		300	3.0	9.1	9.1		
HPE 50D	25	HC-PQ13	120	5.7	11.5	22	≤ 15	1.5
	50		60	5.7	11.5	22		
	100		30	5.2	10.5	26		
HPE 70S	5	HC-PQ43	600	6.1	18	18	≤ 12	3.4
	10		300	12.4	29	36		
HPE 70D	25	HC-PQ23	120	15	32	44	≤ 15	3.2
	50		60	16	32	75		
	100		30	15	29	75		
HPE 90D	25	HC-PQ43	120	21	80	170	≤ 15	8.1
	50		60	40	80	170		

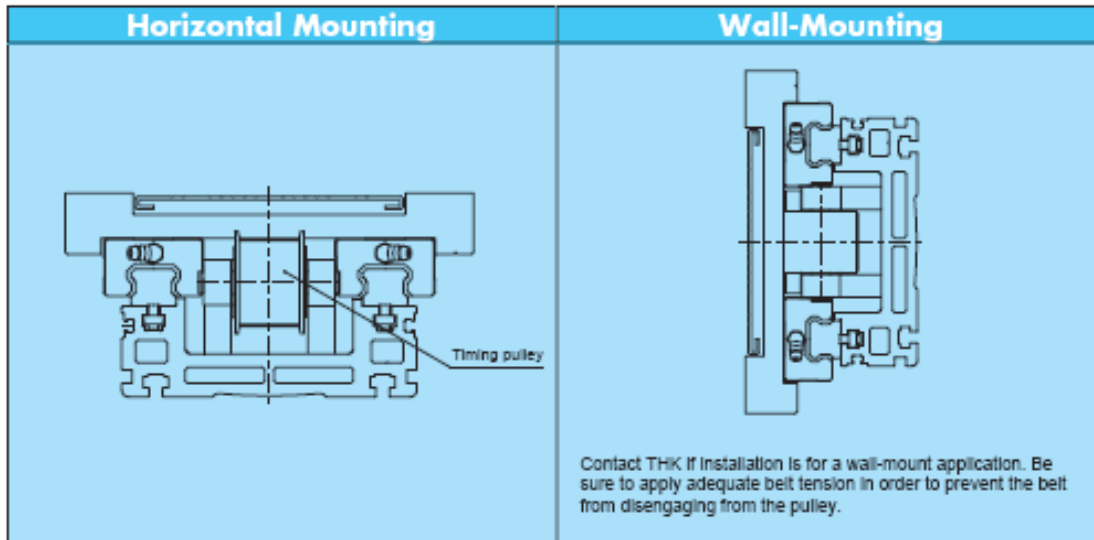
For ultra low backlash specify Omagear HP series gearheads.

Mclennan Servo Supplies Ltd. Tel: +44 (0)8707 700 700 www.mclennan.co.uk



Anexo B.4 Hoja de datos del actuador modelo GL-N

[Timing Belt Type]



Note 1: When mounting the GL-N to a vertical surface in a horizontal mode, the timing belt may sag or bow, this must be taken into consideration as the belt may tend to rub in the flange/pulley area.

Note 2: When using a horizontal model for a wall-mount application, caution must be observed due to the firm grip of the belt to the timing pulley, which is from the timing belt's weight. It will cause damage to the timing pulley. Proper belt tension must be maintained to prevent the timing belt from coming off of the pulley.

● Specification

Model number		GL15N						GL20N					
		Ball Screw			Belt			Ball Screw			Belt		
Screw lead	(mm)	5, 10, 16, 20, 30						—					
Pulley pitch diameter	(mm)	—						35.01					
Repeatability	(mm)	±0.02						±0.08					
Effective stroke	(mm)	100 to 1200						50 to 1700					
Applicable LM Guide		SSR15V	SSR15W	SHS15V	SSR15V	SSR15W	SHS15V	SSR20V	SSR20W	SHS20V	SSR20V	SSR20W	SHS20V
Rated moment load	MA	84.3	98.2	157.5	112.1	130.7	210.4	165.9	242.5	336.8	150.2	238.1	329.4
	Mb (N·m)	64.5	77.5	154.4	86.4	103.8	206.2	122.5	179.5	327.8	110.9	176.7	318.6
	Mc	75.7	113.1	204.1	108.0	164.8	296.2	128.3	187.5	371.2	123.8	199.3	346.0

Note 1: This repeatability is ensured at an ambient temperature of 20°C.

Note 2: This load moment is a value for life distance of 5000 km.

Note 3: The effective stroke decreases depending on the table length. See the dimensional drawings for stroke details.

