



industriales
etsii

Escuela Técnica
Superior
de Ingeniería
Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Desarrollo de un sistema de control de posicionamiento cámara-pantalla en procesos de micromecanizado

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

Autor: Blas José Carrillo Díaz
Director: Horacio T. Sánchez Reinoso



Universidad
Politécnica
de Cartagena

Cartagena, Junio de 2016

ÍNDICE

Capítulo 1 Introducción	6
1.1. Objetivo	7
1.2. Antecedentes	7
1.3. Fases del trabajo	10
Capítulo 2 Sistemas de control	12
2.1. Definición de un sistema de control	12
2.2. Características del sistema de control	12
2.3. Requisitos para un buen sistema de control	13
2.4. Tipos de sistemas de control	13
2.4.1. Sistema de control de lazo abierto	14
2.4.2. Sistema de control de lazo cerrado	15
Capítulo 3 Desarrollo del hardware del sistema cámara-pantalla	17
3.1. Proceso de modelado	18
3.2. Motores paso a paso	26
3.3. Integrado L293B	27
3.4. Tarjeta de adquisición de datos NI USB 6001	28
Capítulo 4 Desarrollo experimental en lazo abierto	29
4.1. Funcionamiento del sistema	29
4.2. Concepto de píxel	31
4.3. Fundamento matemático del algoritmo de control de posicionamiento	33
4.3.1. Cálculo de los centros de masa	33
4.3.2. Cálculo de la distancia entre dos píxeles iluminados	33
4.3.3. Conversión de distancia en píxeles a distancia en milímetros	34
4.3.4. Avance real	34
4.4. Ensayo en lazo abierto	35
4.5. Estudio de repetibilidad	44
Capítulo 5 Desarrollo del algoritmo de control del hardware del sistema de control en lazo cerrado	46
5.1. Módulo de control de los motores paso a paso	47
5.2. Módulo de visión artificial	49
5.2.1. Módulo de control de la cámara web	49
5.2.2. Módulo de cálculos	51
5.3. Uso mediante formularios	54
5.3. Visual Basic para aplicaciones	56
Capítulo 6 Ensayo de posicionamiento en lazo cerrado	58
Capítulo 7 Conclusiones y líneas futuras	62
Bibliografía	64
Anexos	66

ÍNDICE DE FIGURAS

Fig. 1.1 Patrón de imagen junto con la imagen captada por la cámara	8
Fig. 1.2 Prueba de funcionamiento típica	9
Fig. 1.3 Esquema del prototipo usado en el ensayo	10
Fig. 2.1 Diagrama de bloques en lazo abierto	14
Fig. 2.2 Diagrama de bloques en lazo cerrado	15
Fig. 3.1 Esquema de los elementos del demostrador	18
Fig. 3.2 Vista explosionada del sistema de posicionamiento	19
Fig. 3.3 Proceso de extrusión de la base	20
Fig. 3.4 Soporte de la pantalla LCD.....	20
Fig. 3.5 Soporte del primer nivel	21
Fig. 3.6 Patín	21
Fig. 3.7 Detalle del corte para stepper.....	22
Fig. 3.8 Plataforma móvil. Eje X.....	22
Fig. 3.9 Motor paso a paso	23
Fig. 3.10 Soporte de la cámara web	23
Fig. 3.11 Soporte del tornillo guía	23
Fig. 3.12 Tornillo guía	23
Fig. 3.13 Relaciones de posición.....	24
Fig. 3.14 Menú para añadir piezas al ensamblaje	24
Fig. 3.15 Relación de posición. Coincidente	25
Fig. 3.16 Ensamblaje final del demostrador con los dispositivos electrónicos de control	25
Fig. 3.17 Vista de sección	26
Fig. 3.18 Motor paso a paso	27
Fig. 3.19 L293B	28
Fig. 3.20 NI USB 6001	28
Fig. 4.1 Imagen capturada por la cámara web	30
Fig. 4.2 Montaje de la tarjeta de adquisición de datos	30
Fig. 4.3 Diferentes tipos de información en píxeles	32
Fig. 4.4 Valores ponderados de intensidad de los píxeles de un punto de control de posición	32
Fig. 4.5 Máquina de medición por coordenadas.....	41
Fig. 4.6 Comparación gráfica de los resultados del ensayo. Ida	42
Fig. 4.7 Comparación gráfica de los resultados del ensayo. Vuelta	43
Fig. 4.8 Gráfica del estudio de repetibilidad	45
Fig. 5.1 Esquema de conexiones entre los diferentes elementos.....	46
Fig. 5.2 Flujograma del módulo de control de los motores paso a paso	48
Fig. 5.3 Flujograma del módulo de control de la cámara web	50
Fig. 5.4 Flujograma del módulo de cálculos	52
Fig. 5.5 Hoja de cálculo. Centros de masa y distancias	53
Fig. 5.6 Hoja de cálculo. Conversión de unidades	53
Fig. 5.7 Resumen funcionamiento del algoritmo	54
Fig. 5.8 Formulario inicio	54
Fig. 5.9 Formulario de movimiento de motores	55
Fig. 5.10 Interfaz gráfica de usuario en Visual Basic para aplicaciones.....	57
Fig. 6.1 Mensaje en pantalla final	59

ÍNDICE DE TABLAS

Tabla 3.1 Lista de materiales del sistema de posicionamiento.....	19
Tabla 4.1 Especificaciones de la pantalla	34
Tabla 4.2 Posición de los centros de masa de los píxeles de una imagen	36
Tabla 4.3 Posición de los centros de masa de los píxeles con los que se va a trabajar	37
Tabla 4.4 Desplazamiento medido en píxeles entre imágenes.....	38
Tabla 4.5 Avance real medido en milímetros.....	39
Tabla 4.6 Distancias entre píxeles iluminados	39
Tabla 4.7 Tabla de conversión píxeles-milímetros.....	40
Tabla 4.8 Resultados del ensayo en lazo abierto. Ida	40
Tabla 4.9 Error medio acumulativo del ensayo. Ida.....	41
Tabla 4.10 Resultados del ensayo en lazo abierto. Vuelta.....	42
Tabla 4.11 Error medio acumulativo del ensayo. Vuelta	43
Tabla 4.12 Error en posicionamiento Y	44
Tabla 6.1 Centros de masa de los píxeles utilizados	60
Tabla 6.2 Desplazamientos en píxeles	61
Tabla 6.3 Resultados ensayo en lazo cerrado	61

Capítulo 1

Introducción

Hoy en día, la miniaturización de los componentes se ha extendido a lo largo de las últimas décadas en diferentes campos de producción.

Las técnicas utilizadas en el control de posicionamiento de un micro-máquina-herramienta se basan generalmente en sistemas de lazo cerrado. Dentro de estos sistemas, el procedimiento de calibración puede representar un inconveniente importante, como por ejemplo, para garantizar la perpendicularidad entre ejes, donde se debe realizar una calibración que se lleva a cabo por medio de dispositivos auxiliares, tales como encoders o resolvers. Por otra parte, los gastos relacionados con la implementación de dichos dispositivos, así como el control del hardware asociado a ellos, elevan de manera notoria el coste del desarrollo de la máquina de micromecanizado. Otros métodos incluyen algoritmos de posicionamiento a partir de patrones de imágenes que se capturan con una cámara de fotos (D. Leviton, 2002). Dentro de este enfoque, Montes et al. ha desarrollado una mejora en el sistema de control, con lo que se utiliza una pantalla LCD como sistema de referencia.

El método presentado en este trabajo simplifica el proceso de posicionamiento mediante el desarrollo y uso de un algoritmo de detección y tratamiento de píxeles. Este algoritmo se basa en la detección de píxeles iluminados en una pantalla que superan el umbral intensidad de color establecido, su conversión a valores operables matemáticamente y los cálculos con dichos valores que resultarán en las órdenes dadas a los motores del sistema de posicionamiento.

El prototipo micromecanizado con un diseño robusto realizada específicamente y mejorado se presenta a fondo a lo largo de esta investigación. Además, se describen el procedimiento experimental y el análisis exactitud de los resultados.

1.1. Objetivo

El presente Trabajo de Fin de Grado tiene por objeto desarrollar un sistema de posicionamiento cámara-pantalla suficientemente preciso para ser aplicado a procesos de micromecanizado. Los principales objetivos a alcanzar en este Trabajo de Fin Grado son los siguientes:

- Estudio de los sistemas de posicionamiento en máquina-herramienta, comúnmente utilizados en la industria, y de las técnicas existentes para mejorar la precisión en el posicionamiento, y por lo tanto la calidad final de las piezas fabricadas.
- Implementación de nuevas tecnologías en los procesos de micromecanizado, tales como el uso de pantallas LCD correspondientes a Smartphone o dispositivos similares.
- Diseñar el modelo 3D de un sistema de posicionamiento cámara-pantalla, que mantenga una adecuada relación entre precisión de posicionamiento y coste, asegurando al mismo tiempo suficiente robustez mecánica de la micromáquina.
- Desarrollo de un algoritmo capaz de controlar el sistema de posicionamiento de la máquina herramienta con suficiente precisión para ser competitivo en la industria del micromecanizado.
- Estudiar los resultados obtenidos en los ensayos realizados con el hardware de control, fabricado a partir del diseño 3D de la máquina herramienta y sacar las conclusiones más importantes en relación a la precisión en el posicionamiento y el alcance de las tolerancias de fabricación en las piezas fabricadas.

1.2. Antecedentes

Los artículos y patentes relacionados con el estudio presentan, como se ha indicado en la introducción, métodos basados en el uso de encoders o resolvers y métodos basados en referencias en la imagen para el cálculo de la posición.

Se describen a continuación las investigaciones más recientes relacionadas con el estudio y se detallan los objetivos logrados.

En primer lugar se encuentra la investigación realizada por D. Chu *et al.*, 2006.

Este artículo trata de encontrar mediante un método matemático una forma rápida y precisa de ubicar una pequeña imagen (sub-imagen) dentro de una imagen mayor (imagen completa)

Para la solución, se basa en dos partes:

- Decodificación "gruesa" de la imagen: la imagen se localiza entre dos vectores principales adyacentes.

- Decodificación "fina" (Interpolación): la imagen se localiza de forma más precisa a nivel de una fracción de píxel mucho más pequeña.

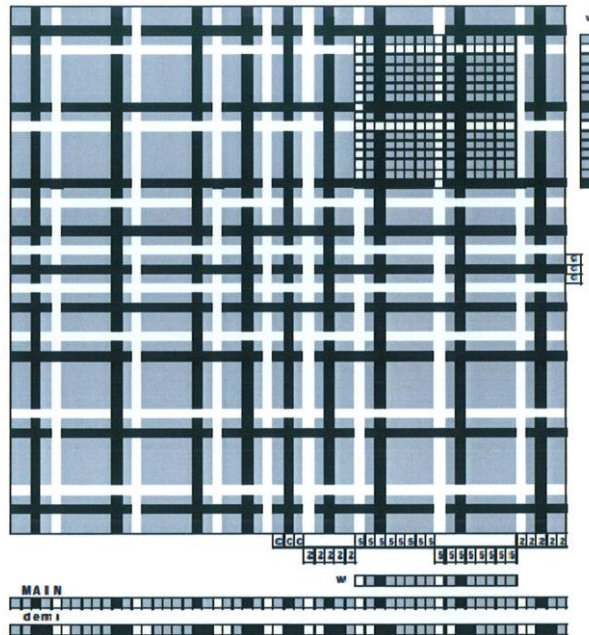


Fig. 1.1 Patrón de imagen junto con la imagen captada por la cámara

Para probar los algoritmos de codificación y de decodificación, realizan un objetivo de 7.11x7.11 mm basándose en un patrón de diseño 711x711 mostrado en la figura 1.1. Es un objetivo de 2 tonos con cada píxel gris aproximado por 8 subpíxeles negros y 8 subpíxeles blancos dispuestos en un tablero de ajedrez. El objetivo está montado en una base con el movimiento limitado a una sola dimensión. Se puede colocar manualmente en las dos dimensiones que abarcan toda la imagen global. La iluminación en el eje es proporcionada por un LED a través de un divisor de haz no polarizante (NPBS) de 550 nm.

La ampliación se ajusta a 1:1 entre el píxel de la imagen y el sensor del píxel, con ayuda de un patrón de "eye-chart" fuera del objetivo. Para comprobar la eficacia de los algoritmos de decodificación, la velocidad de toma de datos es de 99 fotogramas por segundo con decodificación de software.

Obtiene problemas en los límites de píxel ya que la distorsión y la iluminación hacen que los bordes no sean precisos. Utiliza una curva de corrección cúbica siguiendo la fórmula $C_1(-0.5) + C_3(-0.5)^3$, el error de esta corrección es de 130 nm sobre la imagen. Los resultados obtenidos son del orden de 0,5 micras/div.

En la figura 1.2 se representa una prueba de funcionamiento típica que muestra las travesías de destino sin problemas sobre 20 píxeles (0.2 mm) en 10 segundos.

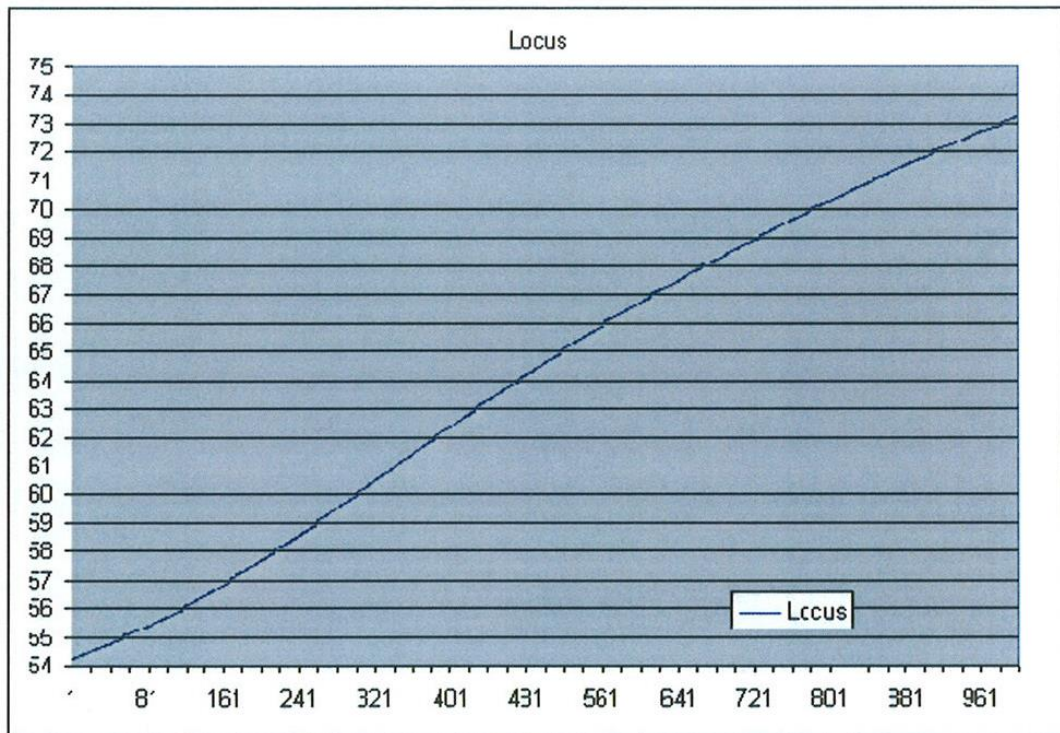


Fig. 1.2 Prueba de funcionamiento típica

En segundo lugar se estudia la investigación llevada a cabo por C. A. Montes *et al.*, 2010, que se comentaba en la introducción.

En este artículo se desarrolla un algoritmo de procesamiento de imágenes que es capaz de ubicar la posición en la imagen global. Este algoritmo permite una resolución de posicionamiento inferior a 1/100 parte del tamaño de píxel en la pantalla y está basado en representar el objetivo por la intersección de dos líneas perpendiculares que aparecen en una pantalla LCD, es decir, un objetivo en forma de cruz. Las pruebas que han realizado han demostrado que para una pantalla de 8 bits, con intensidades de los píxeles que varía de 0 (más bajo, negro) a 255 (el más alto: R, G o B), la configuración óptima se logra mediante la visualización de un objetivo blanco sobre fondo negro.

La baja velocidad, en frames por segundo, y el tiempo de procesamiento de imagen causan demoras en la adquisición de la posición calculada de la imagen destino. El retraso en el sistema de control basado en la visión se dirige a través de una versión modificada del esquema de predictor.

Los ensayos realizados presentan una desviación estándar de 0.253 μm , los resultados experimentales demuestran que las resoluciones del orden de 2 micras se pueden detectar con fiabilidad y los comandos de desplazamiento del mismo tamaño pueden ser emitidos a través de las variaciones de intensidad.

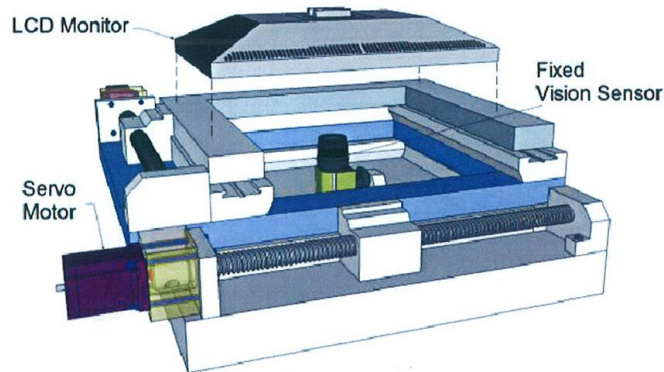


Fig. 1.3 Esquema del prototipo usado en el ensayo

En la Figura 1.4 se muestra el objetivo pixelado dinámico en la pantalla LCD con cuatro elementos de referencia y en la Figura. 1.5 el esquema del modelo.

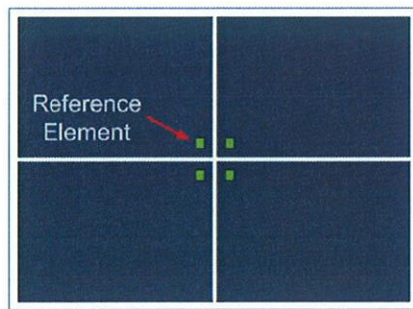


Fig. 1.4 Objetivo pixelado dinámico

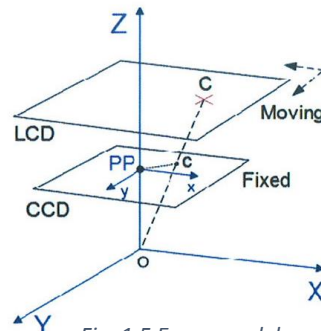


Fig. 1.5 Esquema del modelo

Con los datos experimentales que presenta usando imágenes BMP se obtiene una desviación estándar de $0.253 \mu\text{m}$. En la Figura. 1.6 se representa el desplazamiento horizontal y en la Figura 1.7 el desplazamiento vertical.

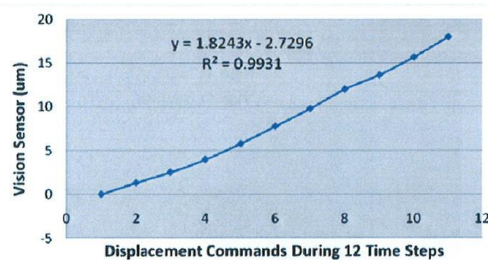


Fig. 1.6 Desplazamiento horizontal

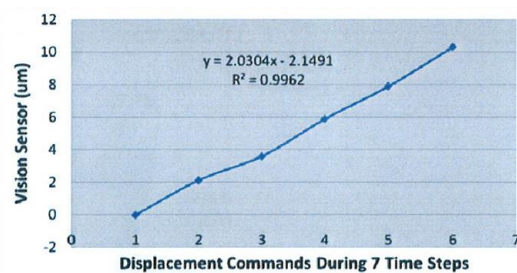


Fig. 1.7 Desplazamiento vertical

1.3. Fases del trabajo

- Análisis del problema a abordar y estudio del material relacionado disponible. El estudio abarca la comprensión de las técnicas utilizadas actualmente para el posicionamiento y el funcionamiento de los sistemas máquina herramienta para micromecanizado.
- Identificación de los objetivos concretos del trabajo y desarrollo de plan de acción.

- Diseño del modelo 3D de la máquina herramienta. Para la realización de dicho modelo será necesario la familiarización y el aprendizaje del software adecuado, en este caso, se trata del programa “SolidWorks”.

- Conexión de los motores paso a paso de la máquina herramienta a un PC y posterior desarrollo del algoritmo necesario para el control de la máquina. En esta fase será necesario el aprendizaje del lenguaje de programación Visual Basic.

- Puesta en marcha en taller del sistema de posicionamiento desarrollado:

- Relación de ensayos de posicionamiento con motor paso a paso y sistema de control cámara-pantalla.

- Análisis de los errores de precisión de la máquina de micromecanizado y representación de los mismos.

- Recopilación de todo el material bibliográfico, tales como artículos web, manuales, etc. utilizados durante la realización del TFG.

- Presentación del resultado final y conclusiones.

- Redacción y revisión final del TFG.

- Preparación de la defensa final del TFG.

Capítulo 2

Sistemas de control

2.1. Definición de un sistema de control

Un sistema de control es un sistema de dispositivos o conjunto de dispositivos, que gestiona, ordena, dirige y regula el comportamiento de otro/s dispositivo/s o sistema/s para lograr unos resultados deseados. En otras palabras, un sistema de control es un sistema, que controla otro sistema.

A medida que la civilización humana se está modernizando, la demanda de la automatización aumenta en consecuencia. Una alta automatización requiere un control de los dispositivos. En los últimos años, los sistemas de control juegan un papel principal en el desarrollo y avance de la tecnología moderna y la civilización. Prácticamente todos los aspectos de nuestra vida cotidiana se ven afectados en mayor o menor medida por algún sistema de control. La cisterna del inodoro en el cuarto de baño, un frigorífico, un aire acondicionado, un automóvil, todos ellos utilizan sistemas de control. Estos sistemas también se utilizan en procesos industriales para conseguir mayores rendimientos. Nos encontramos con el sistema de control en el control de calidad de los productos, sistema de armas, sistemas de transporte, sistema de energía, tecnología espacial, la robótica y muchos más. Los principios de la teoría de control son aplicables tanto a los campos relacionados con la ingeniería como a los que no lo están.

2.2. Características del sistema de control

La característica principal del sistema de control es que debe haber una clara relación matemática entre la entrada y la salida del sistema. Cuando la relación entre la entrada y la salida del

sistema puede ser representada por una proporcionalidad lineal, el sistema se denomina sistema de control lineal. Por otro lado, cuando la relación entre la entrada y la salida no puede ser representada por una sola proporcionalidad lineal y, sin embargo, están relacionadas por alguna relación no lineal, el sistema se conoce como sistema de control no lineal.

2.3. Requisitos para un buen sistema de control

- **Precisión:** La precisión es la tolerancia de medición del instrumento y define los límites del error cometido cuando el instrumento se utiliza en condiciones normales de funcionamiento. La precisión puede ser mejorada mediante el uso de elementos de retroalimentación. Para aumentar la precisión de cualquier sistema de control, el detector de error debe estar presente en el sistema de control.

- **Sensibilidad:** Los parámetros del sistema de control siempre están cambiando debido a cambios de las condiciones del entorno, a modificaciones internas o cualquier otro parámetro. Este cambio se puede expresar en términos de sensibilidad. Cualquier sistema de control debe ser insensible, en la medida de lo posible, a tales parámetros y sensible sólo a señales de entrada.

- **Ruido:** Una señal de entrada no deseada se conoce como ruido. Un buen sistema de control debe ser capaz de reducir el efecto del ruido para un mejor rendimiento.

- **Estabilidad:** Es una característica importante del sistema de control. Para una señal de entrada concreta, la salida debe ser también concreta, y si la entrada es cero, entonces la salida debe ser cero. De esta manera dicho sistema de control se considera que es un sistema estable.

- **Ancho de banda:** Un rango de frecuencia operativo determina el ancho de banda del sistema de control. El ancho de banda debe ser tan grande como sea posible para que la respuesta de frecuencia sea la de un buen sistema de control.

- **Velocidad:** Es el tiempo que tarda el sistema de control en lograr una salida estable. Un buen sistema de control posee una alta velocidad. El período transitorio para dicho sistema es muy pequeño.

- **Oscilación:** Un pequeño número de oscilaciones o una pequeña oscilación constante a la salida suponen que el sistema tienda a ser estable.

2.4. Tipos de sistemas de control

Como hemos visto en muchas máquinas e instalaciones, las magnitudes físicas (temperatura, presión, fuerza, desplazamiento, etc.) deben alcanzar un valor determinado (por ejemplo, la posición del carro de una máquina-herramienta) independientemente de influencias exteriores perturbadoras o ruidos. Para este asunto, es necesario realizar dos operaciones unidas la una a la otra: la comparación y el ajuste. El ciclo de operaciones necesarias se realiza en un bucle denominado de regulación.

Hay varios tipos de sistema de control, pero todos ellos son creados para controlar las salidas.

Tomaremos como ejemplo un sencillo controlador de temperatura de una habitación, para ver los distintos tipos. Supongamos que hay un elemento de calefacción simple, que se calienta mientras la alimentación eléctrica está conectada. Mientras la fuente de alimentación de la calefacción esté encendida la temperatura de la habitación sube hasta alcanzar la temperatura deseada, en ese momento se apaga la alimentación eléctrica. A continuación, debido a la temperatura ambiente, la temperatura de la habitación desciende y, manualmente, se enciende la alimentación de la calefacción para alcanzar la temperatura deseada de la habitación de nuevo. De esta manera se puede controlar manualmente la temperatura ambiente en el nivel deseado. Este es un ejemplo de **sistema de control manual**.

Sin embargo, este sistema puede mejorarse aún más mediante el uso de un sistema de conmutación de la fuente de alimentación donde el suministro de electricidad de la calefacción se encienda y se apague en un intervalo de tiempo predeterminado para lograr la temperatura deseada en la habitación. Pero aún se puede mejorar más este sistema para controlar la temperatura de una habitación. Por ejemplo, un sensor mide la diferencia entre la temperatura real y la temperatura deseada. Si hay diferencia entre estos valores, la calefacción comienza a funcionar para reducir la diferencia y cuando esta diferencia se hace menor que un nivel predeterminado, la calefacción deja de funcionar. Estas dos últimas formas del sistema son ejemplos de **sistema de control automático**.

En la primera de las dos soluciones anteriores la entrada del sistema es completamente independiente de la salida del sistema, es decir, la temperatura de la habitación (la salida) aumenta siempre y cuando la fuente de alimentación de la calefacción (la entrada) se mantenga encendida. Eso significa que la calefacción está encendida siempre y cuando la fuente de alimentación esté también encendida, de manera que la temperatura final de la habitación no tiene ningún control sobre la fuente de alimentación, que como dijimos, era la entrada del sistema. Este sistema se conoce como **sistema de control de lazo abierto**. En el segundo caso, la calefacción del sistema depende de la diferencia entre la temperatura real y la temperatura deseada. Esta diferencia se denomina error del sistema. Esta señal de error alimenta de nuevo al sistema para controlar la entrada. Como los caminos de la entrada y la salida, y el camino de realimentación del error crean un bucle cerrado, este tipo de sistema de control se conoce como **sistema de control de lazo cerrado**.

Por lo tanto, hay dos tipos principales de sistemas de control. Son los siguientes:

2.4.1. Sistema de control de lazo abierto

Se trata de un sistema de control en el que la acción de control es totalmente independiente de la salida del sistema, sin bucle de control entre la entrada y la salida del sistema, ni retorno de información. El sistema de control manual sería, por tanto, un sistema de control de lazo abierto. La figura 2.1 muestra el diagrama de bloques del sistema de control de lazo abierto en el que la salida del proceso es totalmente independiente de la acción del controlador.



Fig. 2.1 Diagrama de bloques en lazo abierto

Ventajas del sistema de control de lazo abierto:

- Simple en la construcción y el diseño.
- Económico.
- Fácil de mantener.
- Generalmente estable.

Desventajas del sistema de control de lazo abierto:

- Es inexacto.
- No es completamente fiable.
- Cualquier cambio en la producción no se puede corregir automáticamente.

2.4.2. Sistema de control de lazo cerrado

El sistema de control en el que la salida tiene un efecto sobre la cantidad de entrada de tal manera que la cantidad de entrada se ajustará a sí misma en función de la salida generada se denomina sistema de control de lazo cerrado. Cualquier sistema de control de lazo abierto se puede convertir en un sistema de control de lazo cerrado proporcionándole una retroalimentación. Esta retroalimentación automáticamente hace los cambios adecuados en la salida provocados por las perturbaciones externas. Cuando trabaja de la forma descrita, este sistema de control de lazo cerrado se denomina sistema de control automático. La figura 2.2 muestra el diagrama de bloques del sistema de control de lazo cerrado en el que la retroalimentación se toma de la salida y se alimenta a la entrada.

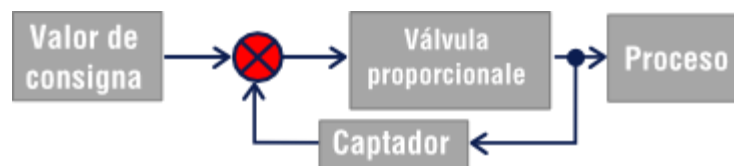


Fig. 2.2 Diagrama de bloques en lazo cerrado

Ventajas del sistema de control de lazo cerrado:

- Este sistema de control es más preciso incluso cuando se presentan relaciones no lineales.
- Alta precisión puesto que cualquier error es corregido mediante la señal de realimentación.
- Gran rango de ancho de banda.

- Facilita la automatización.
- La sensibilidad del sistema es suficientemente pequeña consiguiendo un sistema más estable.

- Este sistema se ve menos afectado por el ruido.

Desventajas del sistema de control de lazo cerrado:

- Es más costoso.
- Es más complicado de diseñar.
- Se requiere más mantenimiento.
- La retroalimentación conduce a una salida oscilatoria.
- La ganancia total se reduce debido a la presencia de la retroalimentación.
- La estabilidad es el principal problema y se necesita más cuidado para diseñar un sistema de lazo cerrado estable.

Capítulo 3

Desarrollo del hardware del sistema cámara-pantalla

La máquina de micromecanizado ha sido diseñada teniendo en cuenta que debe cumplir una serie de parámetros necesarios para asegurar la precisión y robustez del sistema que se requieren. En nuestro caso concreto, además, se buscaba obtener el mejor resultado posible con el menor coste económico, por lo que la primera decisión fue utilizar aluminio para la construcción de la estructura del demostrador.

Se escogió este material por encima de otros que pudiesen ofrecer mayores prestaciones, como puede ser el acero, debido a que la fabricación final a partir de los planos resultaba ser la opción de menor coste económico.

A continuación se consideraron los requisitos principales que debía reunir la máquina, los más relevantes se recogen en la lista que sigue:

- Cantidad de ejes: 2 perpendiculares entre sí, correspondientes a los planos X e Y.
- Actuadores: Motores paso a paso modelo ST28.
- Transmisión: Husillo convencional de paso 0.5 mm.
- Guía para los planos de desplazamiento: Patín modelo BSR2080SL.

Con todos estos requisitos y características de la máquina ya conocidos, se continúa con la explicación sobre la configuración de los elementos en el prototipo y su función en el mismo.

El sistema de posicionamiento consta de dos componentes fundamentales que son las dos plataformas móviles que se mueven perpendiculares entre sí, y que corresponden con los planos X e Y.

La plataforma superior se encarga de situar la pieza de trabajo sobre la superficie superior, mientras que la cámara web está montada en la superficie inferior. La plataforma superior, además, controla el movimiento a lo largo del eje Y por medio de un motor paso a paso, una guía de precisión y un tornillo guía que se fija en la segunda plataforma. La plataforma de movimiento inferior, que controla el movimiento en el eje X, utiliza el mismo sistema, pero se encuentra perpendicular a la guía de la primera plataforma. En ambas plataformas móviles se ha empleado, como se recoge en las características principales vistas anteriormente, unas guías lineales cuya función es asegurar que los desplazamientos de las plataformas se realicen con precisión sobre los ejes.

Por otro lado, la parte fija de la estructura, ubicada en la parte más inferior de la misma, contiene una pantalla LCD de un dispositivo móvil que puede representar determinados patrones de movimiento mediante la iluminación de LEDs. Los movimientos de la pieza de trabajo son capturados por la cámara web, que viene instalada debajo de la plataforma del eje X de forma que queda enfrentada a la pantalla LCD.

Las plataformas móviles están dotadas de movimiento gracias a unos motores paso a paso controlados por un integrado L293B, que hará de driver entre los propios motores paso a paso y la tarjeta de adquisición de datos NI USB 6001.

3.1. Proceso de modelado

Para realizar los distintos diseños, ensamblajes y planos que se incluyen en el anexo, se ha empleado el software SolidWorks 2014, del cual también se incluye un breve manual de sus características en el anexo correspondiente.

Antes de iniciar las explicaciones de como se ha llevado a cabo el trabajo en SolidWorks, se hará una pequeña introducción señalando las piezas más destacables que conforman la máquina de micromecanizado sobre una fotografía del modelo final:

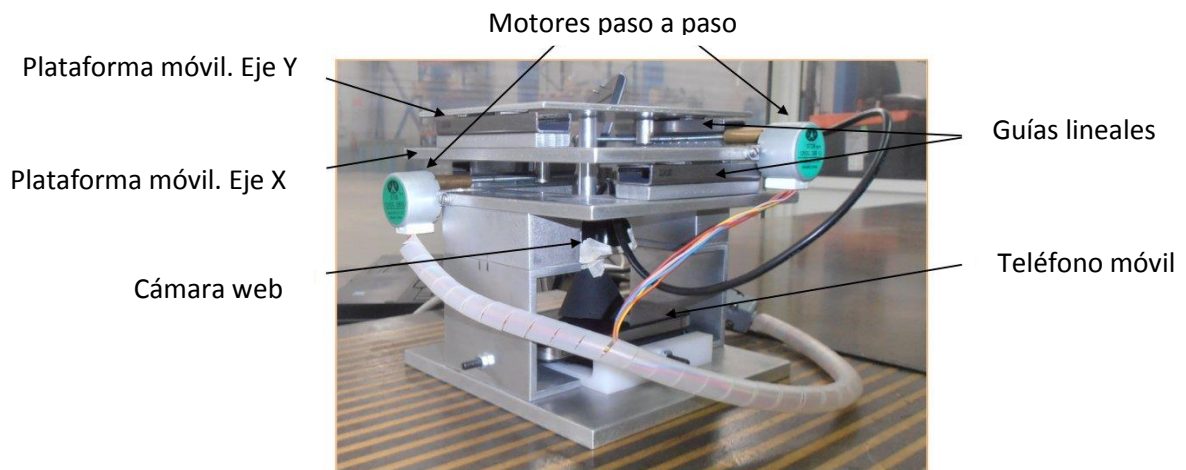


Fig. 3.1 Esquema de los elementos del demostrador

En la siguiente vista explosionada del sistema (la Figura 3.2) se localizan todos los diferentes elementos que lo componen. Se puede encontrar una versión de mayor tamaño de este mismo desglose, incluyendo la tornillería en la lista de materiales del anexo:

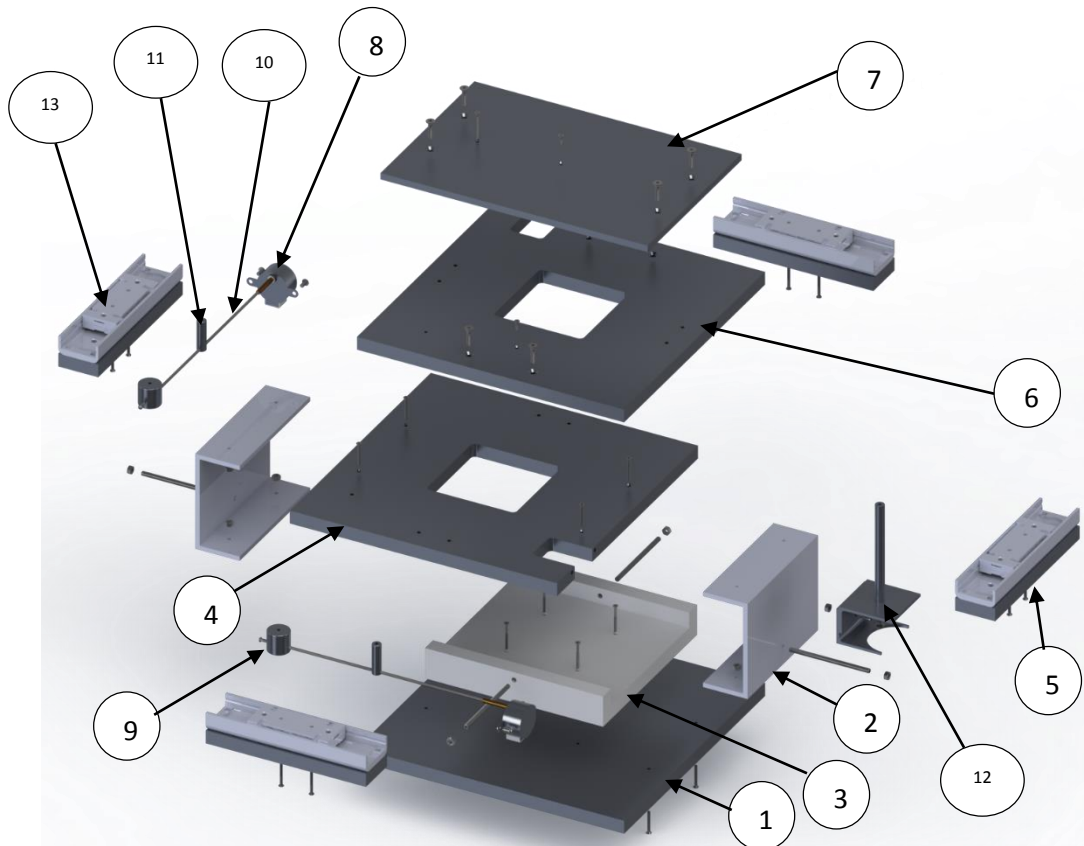


Fig. 3.2 Vista explosionada del sistema de posicionamiento

La Tabla 3.1, que acompaña a la vista explosionada anterior, detalla los componentes del hardware de posicionamiento.

Nº. de elemento	Nombre de pieza
1	Base
2	Soporte del nivel 1
3	Soporte de pantalla LCD
4	Nivel 1
5	Soporte de patín
6	Plataforma móvil. Eje X
7	Plataforma móvil. Eje Y
8	Motor paso a paso
9	Soporte del husillo. Tipo 1
10	Husillo
11	Soporte del husillo. Tipo 2
12	Soporte de cámara web
13	Patín

Tabla 3.1 Lista de materiales del sistema de posicionamiento

Como se puede comprobar en el manual incluido en el anexo para el software que nos ocupa en este trabajo, SolidWorks trabaja a partir de ensamblajes de pequeños sub-ensamblajes, por lo que previamente antes de realizar esos pequeños sub-ensamblajes, deberemos modelar las piezas que conformarán dicho subgrupo. En

En primer lugar para obtener el modelo 3D del hardware de control en lazo cerrado comenzamos por la base, la pieza que dará la estabilidad necesaria que requiere un buen sistema de control. Para generarla se trata de realizar un simple croquis rectangular, al cual se le aplicarán las medidas deseadas mediante la herramienta de cota inteligente, extruido al que se le aplican una serie de taladros que tendrán como objetivo hacer posible la unión de esta parte con las siguientes. El material elegido para la estructura de la máquina es aluminio, tal y como se indicó al comienzo del capítulo.

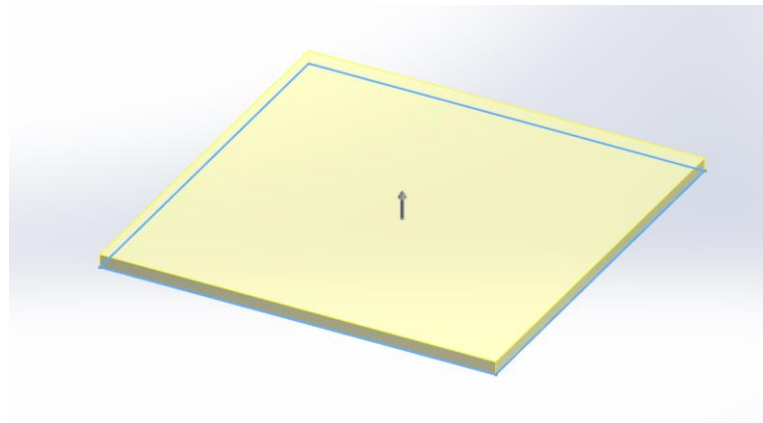


Fig. 3.3 Proceso de extrusión de la base

A continuación se modela la pieza sobre la que se apoyará el dispositivo que utilizaremos como pantalla. En este caso se ha elegido como material de construcción un plástico duro, que confiere una mayor adherencia en su superficie, lo que evitará posibles desplazamientos del dispositivo con la pantalla LCD. El croquis que define este elemento tendrá perfil en U con espesor variable, y una serie de taladros que permitirán la colocación de unos tornillos de plástico, con el fin de mantener el dispositivo pantalla en posición sin que se produzcan movimientos indeseados del mismo, así como el acoplamiento con la base anteriormente modelada. El aspecto final de este elemento se muestra en la Figura 3.4.

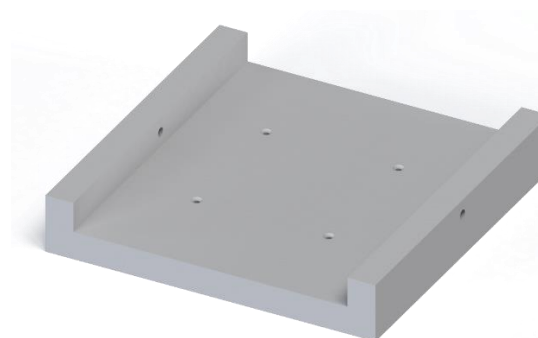


Fig. 3.4 Soporte de la pantalla LCD

El siguiente elemento es el soporte que sujeta el primer nivel (Figura 3.5), que servirá de base a las plataformas móviles. De nuevo un croquis y extrusión simples formarán una pieza con perfil en U y de material estructural aluminio, que contará con una serie de taladros que permitan el ensamblaje posterior. Para llevar a cabo este sistema de sujeción, empleamos un único modelo 3D que será utilizado dos veces en el ensamblaje posterior.

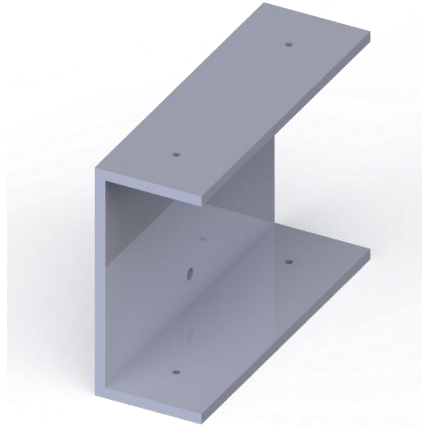


Fig. 3.5 Soporte del primer nivel

Las siguientes piezas, como sucede en el soporte para el primer nivel, son modeladas una vez, pero utilizadas varias veces en el ensamblaje final. Esto mismo ocurre para la base del patín y para el patín en sí mismo, aunque el caso concreto del patín no es distinto. Para la inclusión del patín en el modelo final se recurrió a un diseño descargado del comercial con referencia BSR-SL2080S y, junto al motor paso a paso, son las excepciones respecto al resto de elementos que componen el sistema de posicionamiento, puesto que ninguno de estos dos elementos ha sido modelado durante la realización de este trabajo.

El modelado de la base para el patín se inicia con un croquis rectangular al que se dan medidas a través de la herramienta cota inteligente, se continúa con un extruido y finaliza con la realización de unos taladros para hacer posible su posterior ensamblaje.

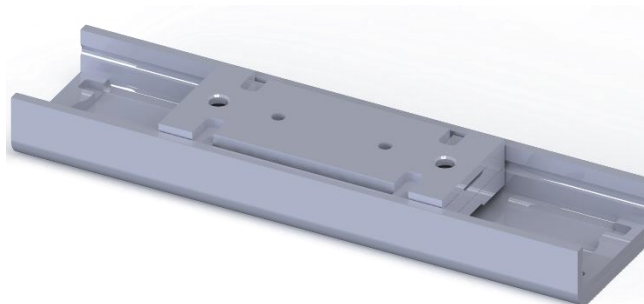


Fig. 3.6 Patín

Una vez los componentes de los niveles inferiores han sido diseñados, se procede con los niveles superiores. En esta ocasión se construyen tres distintas piezas que, debido a la ausencia de simetría en las piezas a la hora de encajar los motores paso a paso que dotarán de movimiento a la cámara y a la herramienta solidaria a esta, serán similares entre ellas pero no iguales y, en consecuencia, no se puede recurrir a un único modelado utilizado varias veces en el ensamblaje como se indicó para algunos soportes anteriormente.

Se comienza a dar forma a estos elementos, nuevamente, con un croquis rectangular con extrusión, esto conforma la estructura general de la pieza en los tres casos, pero ahora, para los niveles en los que se incluyen los motores paso a paso, además de los taladros para ensamblaje posterior, se necesitan unos cortes con bordes redondeados para el buen acoplamiento de estos motores a la máquina. Se muestra un detalle en la Figura 3.7 de estos cortes.

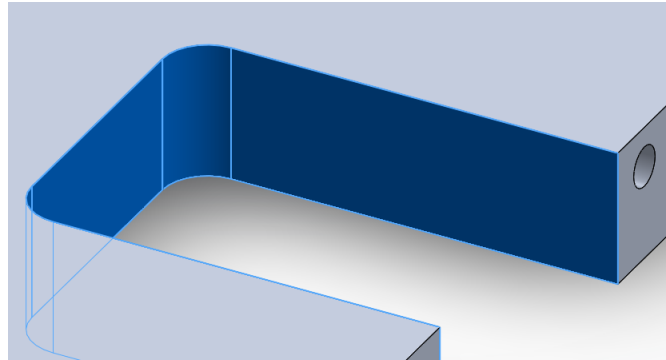


Fig. 3.7 Detalle del corte para stepper

En todas las piezas, excepto en el superior correspondiente al plano Y, que es el que además ejerce de soporte de la cámara, se incluye también un corte rectangular en el centro para que la cámara tenga acceso hasta el dispositivo pantalla. El material escogido para la fabricación de los tres niveles es aluminio, como se indicó al comienzo del proceso de modelado.

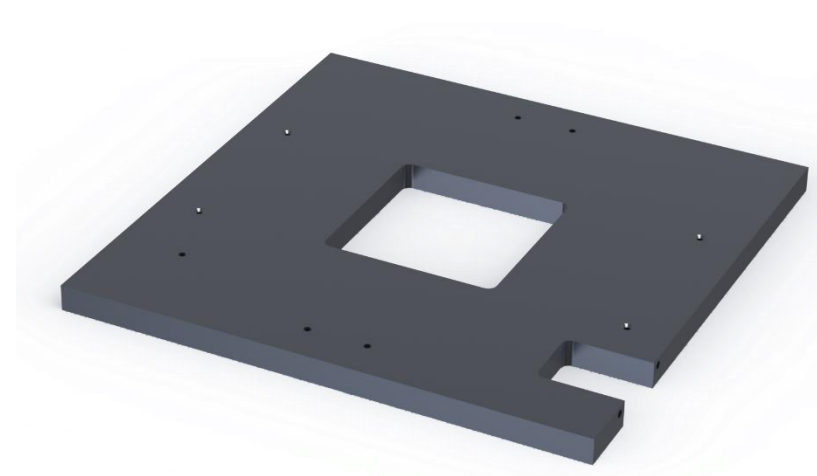


Fig. 3.8 Plataforma móvil. Eje X

Para lograr que la cámara esté a una distancia suficiente de la pantalla, se ha diseñado también un soporte que es uno de los sub-ensamblajes que componen el ensamblaje final, tal y como se hacía referencia al inicio de este apartado. Este sub-ensamblaje está compuesto por un tubo cilíndrico que, por un extremo está fijado al nivel superior, y por el otro sostiene la cámara con la ayuda de dos piezas que la atrapan, de tal forma que quede libre el objetivo de la cámara. Otro sub-ensamblaje es el que define el diseño del motor paso a paso, y que como fue mencionado previamente, es la otra excepción ya que su diseño fue descargado de la página web del comercial.

Como se observa en la Figura 3.9 el material estructural sigue siendo aluminio, como se ha venido utilizando para la mayor parte de las piezas que conforman la máquina, tal y como se indicó al comienzo del capítulo.

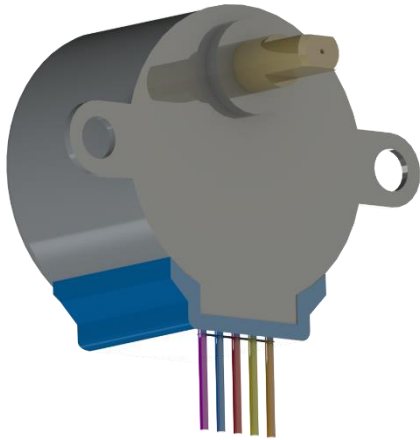


Fig. 3.9 Motor paso a paso



Fig. 3.10 Soporte de la cámara

Finalmente, los últimos modelos 3D diseñados son el tornillo guía y las sujeciones del mismo. El primer elemento se consigue a partir de una espiral de corte a lo largo de un tubo cilíndrico, donde a la espiral se le aplica un punto de inicio, que será el final del tornillo, y un paso que corresponderá con el que se indicó al comienzo del capítulo en las características de la máquina.

El segundo elemento, la sujeción del husillo, al igual que el tornillo también se utilizará el modelado en varias ocasiones al ensamblar debido a la existencia de los dos planos de movimiento, X e Y. Para su diseño lo primero es realizar un croquis rectangular extruido, al que se le aplica un taladro, que será donde se apoye el tornillo guía, y por último se redondean los bordes hasta conseguir la forma final de un pequeño tubo de aluminio con un taladrado.



Fig. 3.12 Tornillo guía

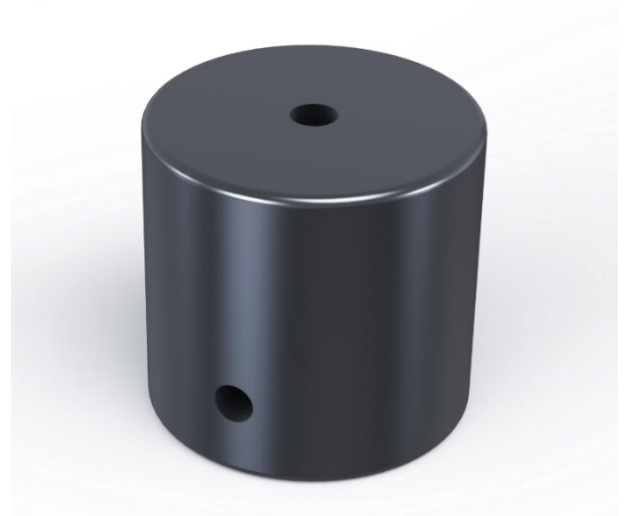


Fig. 3.11 Soporte del tornillo

Con todos los elementos necesarios ya modelados se procede al ensamblaje del sistema completo. Para realizar esta operación es preciso abrir un nuevo tipo de archivo, en esta ocasión un archivo de ensamblaje donde se irán añadiendo todas las distintas piezas cuyo modelaje se ha ido detallando durante el presente capítulo. El tipo de archivo y funcionamiento del mismo puede encontrarse explicado en mayor profundidad en el anexo correspondiente al software SolidWorks.

Para conseguir que cada elemento ocupe su posición se ha utilizado la herramienta relación de posición, de forma que se indica para cada pieza que posición tiene con respecto a otra. De esta manera se van restringiendo grados de libertad de la pieza que se está ensamblando, hasta conseguir su posición final, tal y como tendrá en su fabricación final.

En la siguiente imagen (Figura 3.13) se puede observar un ejemplo de algunas de estas relaciones de posición y como se identifican en pantalla mediante colores, tras haber añadido gran cantidad de ellas para los distintos elementos que componen toda la estructura:

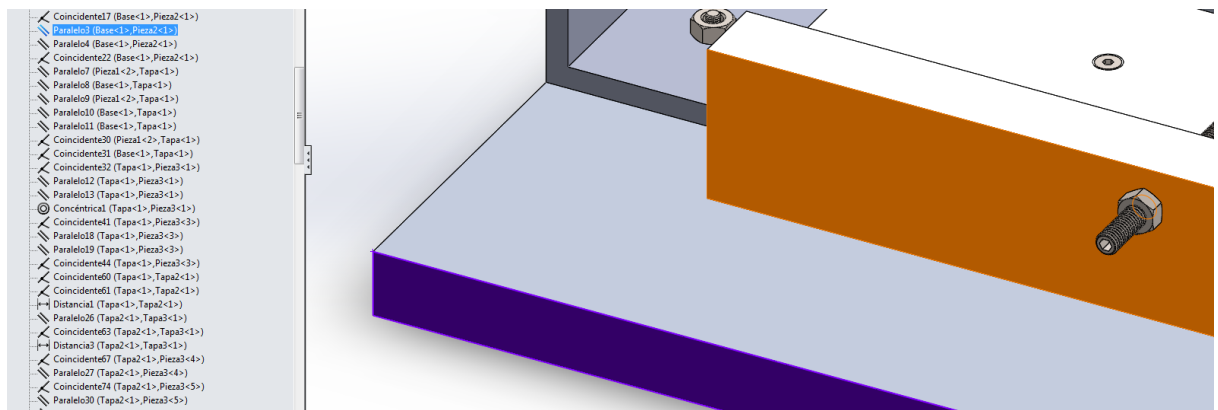


Fig. 3.13 Relaciones de posición

Un ejemplo del procedimiento seguido para ir acoplando los elementos en el ensamblaje se explica a continuación, tomando como caso el ensamblaje de la base de la estructura y la base donde se colocará el dispositivo móvil con la pantalla LCD.

En primer lugar se añaden los elementos que se quieren incluir en el ensamblaje a través del menú que se puede observar en la Figura 3.14:

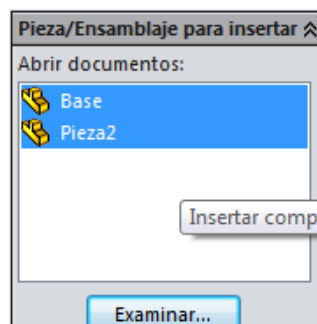


Fig. 3.14 Menú para añadir piezas al ensamblaje

Una vez ambos elementos están colocados en el área de trabajo, se procede a utilizar la mencionada herramienta relación de posición. Para ello se clica en su icono y a continuación se seleccionan las dos superficies que deben coincidir, la superior de la base de la estructura y la inferior de la base para la pantalla. Aparece un nuevo menú como el de la Figura 3.14, donde se escogerá la opción “coincidente”.

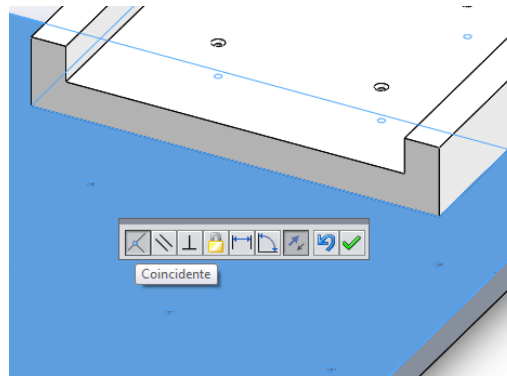


Fig. 3.15 Relación de posición. Coincidente

Los siguientes pasos consisten en un proceso prácticamente idéntico al anterior, pero ahora con los lados que se requieren paralelos y con uno de los taladros para tornillo, de manera que se restringen los dos grados de libertad que quedaban entre ambas piezas.

El aspecto final del ensamblaje, tras colocar cada pieza en su lugar y añadir todos los tornillos y tuercas correspondientes es el que se aprecia en la siguiente imagen renderizada:

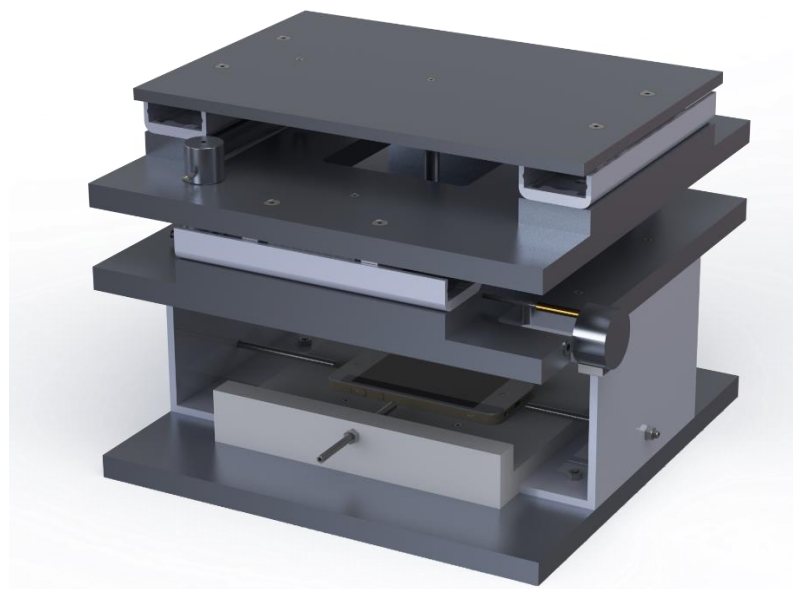


Fig. 3.16 Ensamblaje final del demostrador con los dispositivos electrónicos de control

Se incluye también una vista de sección del ensamblaje, para aquellas partes que quedan ocultas en la vista general de la máquina, como puede ser el soporte de la cámara web.

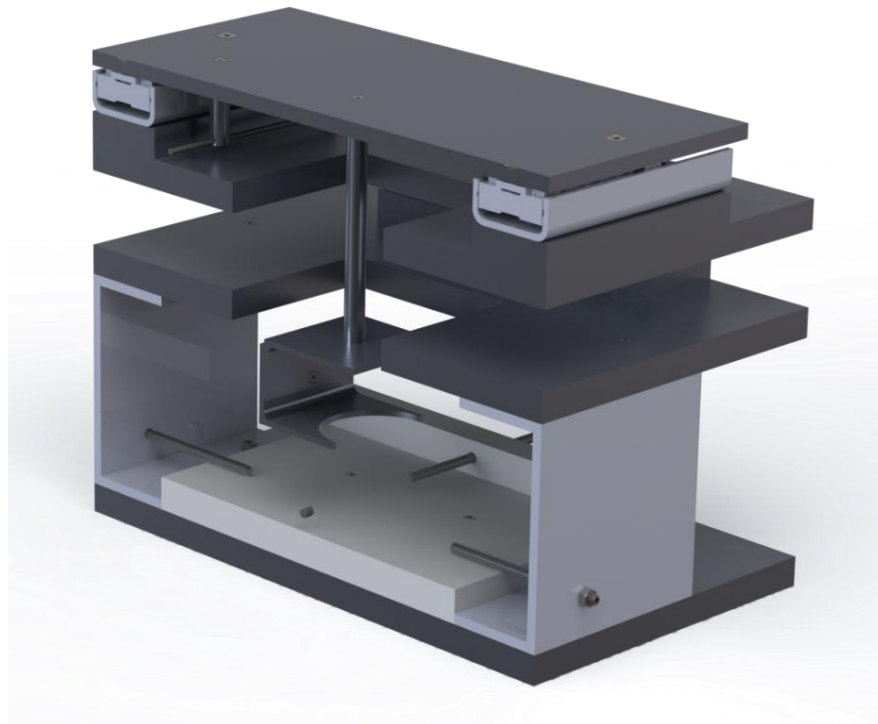


Fig. 3.17 Vista de sección

3.2. Motores paso a paso

Uno de los elementos característicos de esta máquina es el uso de los motores paso a paso, que permiten desplazamientos muy pequeños, necesarios para alcanzar la precisión buscada en este trabajo, y por lo tanto, se debe conocer cuál es su funcionamiento básico.

Los motores paso a paso, son motores en los que podemos controlar el desplazamiento del rotor en función de tensiones que se aplican a las bobinas. La señal eléctrica de alimentación puede ser de corriente continua (c.c), o corriente alterna (c.a), pero lo importante es recibir un tren de pulsos que se suceden con una secuencia, previamente definida, a cada una de las bobinas que componen el estator. Por lo que podemos conseguir controles de desplazamiento hacia delante y hacia detrás, determinado un número de pasos por vuelta.

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos.

Básicamente estos motores están constituidos normalmente por un rotor sobre el que van aplicados distintos imanes permanentes y por un cierto número de bobinas excitadoras bobinadas en su estator. Las bobinas son parte del estator y el rotor es un imán permanente. Toda la conmutación (o excitación de las bobinas) deber ser externamente manejada por un controlador.

La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde 90° hasta pequeños movimientos de

tan sólo 1.8° , es decir, que se necesitarán 4 pasos en el primer caso (90°) y 200 para el segundo caso (1.8°), para completar un giro completo de 360° .

Estos motores poseen la habilidad de poder quedar enclavados en una posición o bien totalmente libres. Si una o más de sus bobinas están energizadas, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas.

Por lo tanto, si somos capaces de mover el motor en pequeños pasos, esto nos va a permitir controlar su posición, con mayor o menor precisión dependiendo del avance de cada paso.

Además, variando la frecuencia con la que se aplican los pulsos, también estaremos variando la velocidad con que se mueve el motor, lo que nos permite realizar un control de velocidad.

Por último, si invertimos la secuencia de los pulsos de alimentación aplicados a las bobinas, estaremos realizando una inversión en el sentido de giro del motor.

Resumiendo, definimos el motor paso a paso, como aquel motor cuyas bobinas del estator son alimentadas mediante trenes de pulsos, con una determinada frecuencia, y que permite:

- Controlar posición.
- Controlar velocidad.
- Controlar sentido de giro.



Fig. 3.18 Motor paso a paso

3.3. Integrado L293B

Los motores paso a paso necesitan una corriente para funcionar que no se aporta desde los puertos USB del PC que dirige sus acciones, para ello es necesario el empleo de amplificadores de potencia. En este trabajo se ha optado por utilizar una solución barata y común de la que se exponen a continuación sus principales características.

El L293B es un circuito integrado que incorpora dos drivers denominados "puentes H". Es capaz de aportar una corriente de salida de hasta 1 A por canal. Cada canal es controlado por señales de entradas que son compatibles con TTL y cada pareja de canales tiene una señal de habilitación que desconecta la salida de los mismos. Mediante ellos es posible activar un motor de corriente continua, así como establecer su sentido de giro y, con un pequeño circuito adicional, controlar también su velocidad. Estas son algunas de sus características principales:

- Corriente de salida de 1A (amperios) por canal.
- Corriente de salida de pico de 2A por canal.
- Señal para la habilitación de las salidas del integrado.
- Gran inmunidad a los ruidos.
- Alimentación para las cargas separada de la alimentación de control.
- Protección contra elevadas temperaturas.

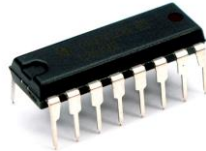


Fig. 3.19 L293B

3.4. Tarjeta de adquisición de datos NI USB 6001

Las órdenes que procedan del algoritmo ejecutado en el PC deben llegar a los motores paso a paso y para conseguirlo, se requiere la intervención de una tarjeta de adquisición de datos que transforme dichas órdenes en señales lógicas de salida que puedan accionar los motores.

Los dispositivos de adquisición de datos (DAQ) de National Instruments, tienen la ventaja de su simplicidad y portabilidad gracias a la conexión USB y ofrecen medidas reales de alto rendimiento.

Para el cometido que ocupa este trabajo estas han sido las características y ventajas que se han tenido en cuenta para su elección por encima de otros dispositivos, como podría ser una tarjeta Arduino.

Sin embargo, las aplicaciones de estos dispositivos de adquisición de datos de National Instruments van desde dispositivos de bajo coste y de una sola función, como es el caso de este proyecto, a sistemas modulares de alto rendimiento ideales para una gran variedad de aplicaciones, desde registro de datos simples hasta sistemas embebidos OEM.

Las características completas del modelo concreto utilizado en los ensayos, National Instruments USB 6001, pueden ser encontradas en las hojas de especificaciones del producto, incluidas en el anexo.



Fig. 3.20 NI USB 6001

Capítulo 4

Desarrollo experimental en lazo abierto

Antes de desarrollar el software que funcionará en lazo cerrado es preciso que conozcamos algunos conceptos, así como comprobar si la precisión del método propuesto es suficiente, y por lo tanto es susceptible de proponer un sistema en lazo cerrado.

En primer lugar, se realizará una descripción del funcionamiento del sistema, a continuación se expondrán ciertos conceptos precisos para entender ciertas partes del procedimiento, así como el fundamento matemático tras el algoritmo y finalmente una explicación sobre en qué consiste el ensayo.

4.1. Funcionamiento del sistema

Para obtener la posición (x, y) de la herramienta con respecto al sistema de coordenadas de la pieza, se definen diferentes planos, cada uno de los cuales tiene su sistema de referencia:

- El plano que contiene la pantalla permanece inmóvil con respecto a la herramienta. En este plano se definen los puntos a los que el centro de la cámara web debe moverse para posicionar la pieza en la herramienta. Un punto de destino en la pantalla tendrá coordenadas (x_p, y_p) . La intensidad de los LEDs se ha reducido de forma que la imagen capturada por la cámara web pueda ver estos LEDs evitando reflejos y falta de nitidez, obteniendo así los colores que no sean el rojo en estas imágenes. Para ello, fue necesario reducir el brillo de la pantalla, y utilizar una intensidad de 60 en el color rojo en una escala de 0 a 255.



Fig. 4.1 Imagen capturada por la cámara web

- El plano que contiene la cámara de manera solidaria con el plano de trabajo sería (x_c, y_c) . Estas son las coordenadas en la imagen tomada de la cámara web. La imagen se ve ampliada en relación al tamaño mostrado en la pantalla.

- El plano de la pieza de trabajo se mueve conjuntamente con la cámara, y en dicho plano se representan las características a mecanizar en coordenadas (x, y) . Este plano es simétrico al de la cámara y tiene unidades de mm.

Para dibujar el punto de destino en la pantalla (x_p, y_p) , que se corresponde con la herramienta colocada en el punto (x, y) de la pieza de trabajo, se realizan las siguientes transformaciones:

- Posicionar el centro de la cámara (x_{c0}, y_{c0}) en relación con el centro de referencia de la pieza de trabajo en coordenadas de la pieza de trabajo (x, y) .
- Posicionar el centro de la cámara con respecto a la pantalla (x_{pc0}, y_{pc0}) cuando la plataforma móvil está centrada en el sistema, lo que permite recorridos máximos en ambos ejes.

La relación de aspecto entre el píxel de la pantalla P_p y el píxel de la cámara P_c , junto con la relación P_p/mm , proporciona las dimensiones reales de la imagen tomada por la cámara.



Fig. 4.2 Montaje de la tarjeta de adquisición de datos

Los motores paso a paso que dotan de movimiento a los planos móviles se conectan al PC mediante una tarjeta de adquisición de datos de bajo coste, modelo National Instruments 6001 USB y un driver para motores de 4 canales modelo L293B. Las especificaciones, tanto para esta tarjeta, el driver, así como para los motores paso a paso, son recogidas en el anexo.

Como se ha descrito, para este sistema de posicionamiento utilizaremos píxeles en una pantalla, por lo tanto, es importante conocer los elementos con los que vamos a trabajar.

4.2. Concepto de píxel

La palabra píxel es una abreviatura de "picture element" o "elemento de imagen" en español. Indexado como una localización desde el origen de la imagen (x, y) o como columna-fila (c, f), representa el elemento constituyente más pequeño de una imagen digital y contiene un valor numérico, que es la unidad básica de información dentro de la imagen a un nivel de resolución y cuantificación espacial dada. Habitualmente, los píxeles contienen la respuesta de color o intensidad de la imagen como una pequeña muestra de luz coloreada de la escena. Sin embargo, no todas las imágenes contienen necesariamente información estrictamente visual. Una imagen es simplemente una señal de 2D digitalizada como una cuadrícula de píxeles, cuyos valores pueden estar relacionados con otras propiedades que no sean las de color o intensidad de la luz. El contenido de la información de los píxeles puede variar considerablemente dependiendo del tipo de imagen que estamos procesando:

- Imágenes a color/en escala de grises. Normalmente estas imágenes contienen información relacionada con la intensidad del color o de la escala de grises en un punto dado en la escena o imagen.

- Infrarrojos (IR). El espectro visual es sólo una pequeña parte del espectro electromagnético. Los infrarrojos nos ofrecen la capacidad de construir imágenes de escenas basadas tanto en la reflexión de la luz infrarroja, como en la radiación infrarroja que se está emitiendo. La radiación infrarroja es emitida en proporción al calor generado/reflejado por un objeto y, por lo tanto, las imágenes obtenidas a través de infrarrojos también se conocen como imágenes térmicas. Cuando la luz infrarroja es invisible para el ojo humano, la iluminación infrarroja y sistemas de obtención de imágenes nos ofrecen un método útil para la vigilancia encubierta (por ejemplo, las imágenes infrarrojas constituyen la base para los sistemas de visión nocturna).

- Imágenes médicas. Muchas imágenes médicas contienen valores que son proporcionales a las características de absorción del tejido con respecto a una señal proyectada a través del cuerpo. Los tipos más comunes son la tomografía computarizada (TC) y la imagen de resonancia magnética (MRI, por sus siglas en inglés *Magnetic Resonance Imaging*). Las imágenes de TC, como los rayos X convencionales, representan valores que son directamente proporcionales a la densidad del tejido a través del cual pasa la señal. Por el contrario, las imágenes de resonancia magnética se muestran con mayor detalle, pero no tienen una relación directa con una sola propiedad cuantificable del tejido. Para ambos casos, imágenes TC o imágenes de resonancia magnética, el formato de imagen 2D es frecuentemente convertido a imágenes en 3D; consiguiendo el volumen 3D con una acumulación en una pila de imágenes 2D.

- Imágenes de radar/sonar. Una imagen de radar o sonar representa una sección transversal de un objetivo concreto, en proporción a su distancia desde el sensor y su señal asociada 'reflectividad'. Los radares se utilizan comúnmente en la navegación de aeronaves, aunque también se han utilizado en proyectos de vehículos de carretera. El radar por satélite para monitorizar el clima es actualmente muy común, como también lo es el uso del sonar en la mayoría de los buques oceánicos modernos. El Geo-radar se está utilizando cada vez más para las investigaciones de ciencias arqueológicas y forenses.

- Imágenes 3D. Utilizando técnicas específicas de detección 3D, como la fotografía estéreo o el escaneo láser en 3D, podemos capturar información de los objetos del mundo que nos rodea y representarlos en los sistemas informáticos como imágenes en 3D. Las imágenes 3D corresponden a menudo a mapas de profundidad, en los que la ubicación de cada píxel contiene la distancia desde ese punto hasta el sensor. En este caso, tenemos información 3D explícita, en lugar de sólo una proyección 3D como conseguimos con las imágenes 2D convencionales. En función de la tecnología de captura, puede que tengamos una información de la profundidad 3D, o información de la profundidad 3D y también del color para cada posición de un píxel. La imagen de un mapa de profundidad puede ser re proyectada para dar una visión parcial del objeto 3D capturado (estos datos a veces se denominan 2¹/₂D).

- Imágenes científicas. Muchas ramas de la ciencia utilizan un formato discreto 2D, o 2D basado en uno 3D, para la captura de datos y análisis de resultados. Los valores de los píxeles pueden corresponder a las densidades químicas o biológicas de una muestra, a una impedancia acústica, a una intensidad sónica, etc. A pesar de la diferencia en el contenido de la información, los datos están representados de la misma forma, es decir, una imagen 2D. Por lo tanto, se pueden aplicar técnicas de procesamiento de imágenes digitales en muchas ramas diferentes del análisis científico.

La figura 4.1 muestra algunos ejemplos de imágenes con diferentes tipos de información de los píxeles. Estos son sólo algunos ejemplos, tanto de la variedad de imágenes digitales en uso como de la amplia escala de dominios de aplicación para el procesamiento digital de imágenes. En las imágenes a color que se van a tratar en este trabajo, los píxeles, por lo general, tienen valores enteros dentro de un rango de cuantificación dado (normalmente de 0 a 255 en imágenes de 8 bits), como se representa en la figura 4.2. Aunque para otras formas de imágenes con información (por ejemplo, las médicas, 3D, científicas), los píxeles con valores reales con coma flotante son comunes.

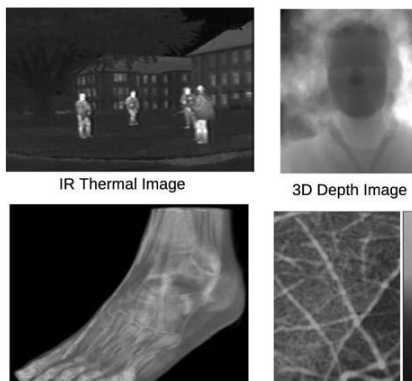


Fig. 4.3 Diferentes tipos de información en píxeles

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	3	3	2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	4	4	4	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	3	13	13	10	13	14	6	3	0	0	0	0	0	0	0	0
0	0	4	16	28	32	39	43	38	26	8	2	0	0	0	0	0	0	0	0
0	4	17	36	49	61	72	72	61	43	19	8	0	0	0	0	0	0	0	0
0	3	11	32	51	75	94	102	103	88	66	41	19	3	0	0	0	0	0	0
1	6	23	42	70	100	123	132	130	114	90	65	24	8	0	0	0	0	0	0
0	5	24	60	92	125	149	155	152	143	119	84	43	8	0	0	0	0	0	0
0	4	20	63	104	139	159	167	159	140	120	96	45	10	0	0	0	0	0	0
0	6	26	65	97	136	151	164	165	151	140	105	53	13	0	0	0	0	0	0
0	4	26	68	97	133	152	162	162	156	146	109	57	14	0	0	0	0	0	0
0	5	20	63	97	127	153	162	164	156	141	101	56	12	0	0	0	0	0	0
0	4	32	66	96	136	153	158	163	163	145	114	62	17	0	0	0	0	0	0
0	4	26	61	96	135	151	160	165	159	146	100	61	15	0	0	0	0	0	0
0	0	27	57	90	129	147	160	165	162	143	111	56	12	0	0	0	0	0	0
0	0	19	52	87	126	152	162	165	159	141	109	56	9	0	0	0	0	0	0
0	0	20	51	81	123	145	151	160	165	145	114	57	14	0	0	0	0	0	0
0	0	19	52	86	125	147	160	164	161	143	97	50	10	0	0	0	0	0	0
0	0	19	47	83	123	149	159	161	167	142	111	58	16	0	0	0	0	0	0
0	0	11	39	75	116	150	159	164	149	135	106	52	13	0	0	0	0	0	0
0	0	5	35	65	110	143	159	160	142	130	94	41	4	0	0	0	0	0	0
0	0	2	38	59	89	123	138	144	134	117	79	28	0	0	0	0	0	0	0
0	0	0	8	37	65	97	115	120	113	95	45	19	0	0	0	0	0	0	0
0	0	0	0	10	35	64	95	97	88	62	37	16	0	0	0	0	0	0	0
0	0	0	0	0	7	22	37	47	51	49	39	8	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 4.4 Valores ponderados de intensidad de los píxeles de un punto de control de posición

4.3. Fundamento matemático del algoritmo de control de posicionamiento

4.3.1. Cálculo de los centros de masa

Para obtener la posición de los píxeles en la imagen, transformamos esta en valores RGB, por lo que tendríamos una hoja de cálculo con una serie de valores, donde cada píxel se vería representado como vimos en la figura 4.3. El modo más favorable para el sistema de control es conocer la posición del centro de cada píxel.

De esta forma, para calcular el centro de cada píxel, podemos recurrir a la analogía de un sólido compuesto por un conjunto de partículas o masas puntuales. En nuestro caso, el sólido sería el píxel, y las masas puntuales todos los distintos valores que constituyen al mismo. Las fórmulas a emplear son, consecuentemente, las siguientes:

$$x_g = \frac{\sum_{i=1}^n \sum_{j=1}^m x_{i,j} \cdot B(i,j)}{\sum_{i=1}^n \sum_{j=1}^m B(i,j)} \quad (4.1)$$

$$y_g = \frac{\sum_{i=1}^n \sum_{j=1}^m y_{i,j} \cdot B(i,j)}{\sum_{i=1}^n \sum_{j=1}^m B(i,j)} \quad (4.2)$$

Por lo tanto, la posición del centro de masa del píxel se calcula como una función de su posición en la red de puntos (x, y) y el valor binario B (i, j), en relación con el grado de intensidad de cada píxel, dividido por el peso total de la cuadrícula.

Una vez que el centro de masa para cada punto de la red de píxeles se ha calculado, el siguiente paso consiste en calcular la distancia media entre cada par de píxeles consecutivos iluminados en la imagen capturada por la cámara web. Por ejemplo, la distancia media, en unidades de píxeles de la cámara, entre dos puntos iluminados consecutivos de una imagen a lo largo del eje Y se ha evaluado como sigue:

$$\bar{d}_{i+1,i} = \frac{\sum_{j=1}^m (y_{g\ i+1,j} - y_{g\ i,j})}{m} \quad (4.3)$$

donde i y j son el número de archivos y la columna de los datos registrados en el archivo .bmp de cada imagen procesada y m es el número total de columnas evaluados con el fin de obtener la distancia media de la cámara de píxeles.

4.3.2. Cálculo de la distancia entre dos píxeles iluminados

Para saber la distancia ente dos píxeles iluminados en la pantalla consultaremos en primer lugar las especificaciones del dispositivo:

Device	Resolution (px)	Pixels per inch	Screen Size (mm)	Dot pitch(mm)
iPhone 5S	1136×640	326	88.5×49.9	0.078

Tabla 4.1 Especificaciones de la pantalla

Sabemos, por tanto, que la distancia entre dos píxeles en pantalla es de 0.078 mm. También sabemos que iluminamos uno de cada dos píxeles de la pantalla, esto, unido al dato anterior de distancia entre píxeles, nos permite conocer la distancia que tenemos entre cada uno de los píxeles iluminados en nuestra cuadrícula, que es:

$$0.078 \text{ mm} \cdot 2 = 0.156 \text{ mm} \quad (4.4)$$

4.3.3. Conversión de distancia en píxeles a distancia en milímetros

Para realizar la conversión de la distancia resultante entre los píxeles de dos imágenes, en unidades de píxeles de la cámara web, a distancia en milímetros comenzamos con la siguiente relación:

$$\frac{\text{Pixel}}{\text{mm}} = \frac{\text{Pixel}}{\text{pulgada}} \cdot \frac{\text{pulgada}}{\text{mm}} = 326 \cdot 0,0393701 = 12.8346526 \frac{\text{Pixel}}{\text{mm}} \quad (4.5)$$

donde hemos hecho uso del dato de la pantalla, píxeles por pulgada, unido a que sabemos que 1 pulgada son 0.0393701 mm.

Con la relación anterior y el valor que obtengamos de dividir el resultado del promedio de las distancias medias entre píxeles de dos imágenes, por la distancia entre píxeles iluminados, efectuamos una relación que nos daría el valor por el que dividir las distancias medias entre píxeles de dos imágenes distintas para obtener su valor equivalente en milímetros. Dicha relación, que puede resultar compleja en su explicación, se ilustra en la expresión siguiente:

$$\begin{aligned} \text{Distancia en mm} &= \frac{\text{Distancia en píxeles}}{\text{Píxeles en imagen/mm}} = \\ &= \frac{\text{Distancia en píxeles}}{\frac{\text{Promedio de distancia en píxeles} / \text{Distancia entre píxeles iluminados}}{\frac{\text{Píxeles por pulgada en pantalla}}{\text{pulgada} / \text{mm}}}} \end{aligned} \quad (4.6)$$

4.3.4. Avance real

En la explicación de la transformación anterior se hacía referencia a un promedio de las distancias medias entre píxeles de dos imágenes distintas. Este valor promedio se obtiene, a su vez, tras otra transformación que agrupa los dos desplazamientos producidos, tanto en el eje x, como en el eje y.

En este caso el cálculo es sólo uno antes del promedio final, y es el siguiente:

$$Avance\ real = \sqrt{Desplazamiento\ en\ píxeles\ en\ X^2 + Desplazamiento\ en\ píxeles\ en\ Y^2} \quad (4.7)$$

Obteniendo un resultado por cada par de comparaciones entre imágenes para cada eje. Se aplica el promedio para todos los avances resultantes y se alcanza el promedio de las distancias medias entre píxeles de dos imágenes distintas utilizado en la conversión posterior.

4.4. Ensayo en lazo abierto

El análisis que se propone para la demostración de la validez del método en este trabajo, consiste en una serie de desplazamientos mediante valores teóricos, donde en cada uno se toma una imagen mediante una cámara web, hasta cubrir una distancia determinada. Esta serie de imágenes formarán lo que se conoce como carrera y cada una de esas imágenes se trabajará con el algoritmo desarrollado para obtener los valores de desplazamiento real. El procedimiento se detalla en mayor profundidad a continuación:

- El primer paso, antes de comenzar el ensayo en sí mismo, es colocar el dispositivo móvil con la pantalla encendida con la imagen de los píxeles iluminados en la base de la máquina. A continuación se conectan los dispositivos relacionados con el PC, la cámara web y la tarjeta de adquisición de datos y se abre la hoja de Excel en la que se ha preparado el código de control, incluyendo el paso a valores RGB de la imagen y los cálculos del apartado anterior.

- El ensayo comienza con la hoja de cálculo que contiene el código de reconocimiento y cálculo de los centros de masa de los píxeles iluminados. En este documento existen formularios para poder hacer uso del algoritmo. Al pulsar en el botón “Snapshot” la cámara web toma una primera imagen de los píxeles iluminados en la pantalla LCD.

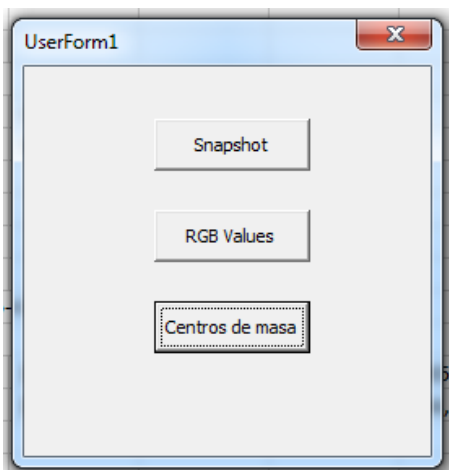


Fig. 4.5 Formulario

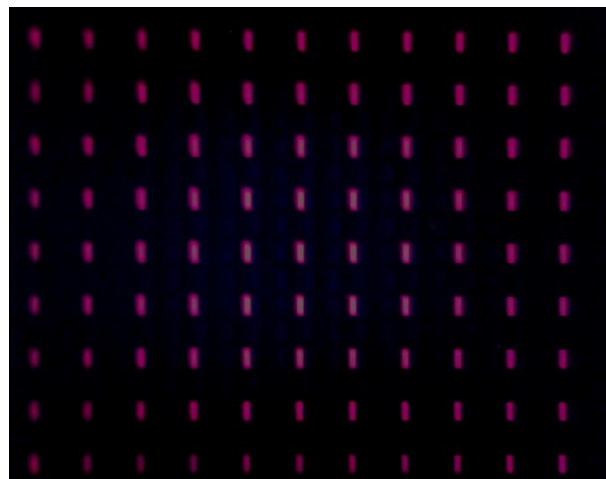


Fig. 4.6 Imagen captada por la cámara web

- Una vez guardada la imagen anterior, y volviendo al formulario de la Figura 4.5, se clicla sobre el botón “RGB Values”, que transformará la captura anterior en valores numéricos de intensidad lumínica, tal y como se explicó en el funcionamiento del sistema. Se obtiene un resultado en la hoja de cálculo como el de la Figura 4.7



Fig. 4.7 Fragmento de la imagen captada por la cámara web en valores RGB

- A partir de esos valores, y de nuevo haciendo uso del formulario de la Figura 4.5 mediante el botón “Centros de masa”, obtenemos a través del código que emplea las ecuaciones (4.1) y (4.2) una tabla como la mostrada en Tabla 4.2. En ella se representa la posición relativa, medida en píxeles tomados por la cámara web, los centros de masa correspondientes a cada píxel iluminado captado por la cámara web en la captura que se realizó en el segundo punto de este ensayo.

Imagen 1										
YG										
65,3658218	68,2365112	69,8821106	70,7569122	71,7524261	73,0657578	73,7142868	74,7190323	75,3880386	76,14888	76,8148041
182,279419	182,496567	183,402374	183,931412	184,766327	185,261703	186,752808	187,371521	188,548569	189,796188	190,068268
296,044464	296,958374	297,452118	297,234039	297,808167	298,181366	299,376953	300,311371	301,570374	302,199646	303,046661
410,336548	410,242096	410,652374	410,711609	411,151917	411,589844	412,464569	413,271759	413,955872	415,395233	416,20282
523,417236	524,620483	524,425415	524,464966	524,687866	525,100342	525,894043	526,45459	527,464844	528,233398	529,048096
638,089355	637,922913	638,404663	638,255493	638,780457	639,285828	639,567993	640,316711	640,859253	641,554504	641,937683
751,859924	752,076355	751,816711	752,159546	752,374268	752,667786	752,954224	753,440674	753,655334	754,248901	754,585938
865,775818	865,28186	865,048279	864,888306	865,322266	865,547241	865,549072	866,32489	866,245239	866,463074	866,828979
979,840149	978,968201	978,634094	977,959351	977,807678	978,2276	978,347046	977,947632	978,383972	979,162354	979,194458
XG										
56,2563286	171,407257	285,535645	399,925568	513,820801	627,683594	741,203674	854,459656	967,606079	1080,62634	1193,54919
56,5455704	170,98584	285,420532	399,206055	512,955566	626,680603	740,458191	853,938293	967,275635	1079,82568	1192,99744
55,7839127	170,128433	284,233765	398,099518	511,838837	625,634033	739,531799	853,306213	966,630615	1079,75049	1192,34851
55,4857254	169,875656	283,598206	397,391022	510,889343	625,002991	738,487732	852,525574	966,150269	1078,79102	1191,505
55,3770676	169,383041	283,182159	396,912354	510,325195	624,215088	738,070801	851,886047	965,475769	1078,33997	1191,06934
54,8929634	169,261124	282,998596	396,735382	510,201263	623,821899	737,661194	851,531189	964,650085	1077,26526	1189,81348
54,7013168	168,865189	283,030396	396,549164	510,380096	623,758484	737,363647	850,654358	963,647827	1076,19092	1188,7489
54,5136642	168,955902	282,774384	396,525299	510,170715	623,431274	736,912537	849,836243	962,97113	1075,29199	1187,81055
54,0978889	168,509979	282,466064	396,430145	510,17627	623,235107	736,408447	849,451538	962,353577	1074,5542	1187,12585

Tabla 4.2 Posición de los centros de masa de los píxeles de una imagen

- Como puede observarse son muchos datos los obtenidos de una sola imagen, por lo que se filtran los valores de trabajo, eliminándose aquellos que quedan más al borde de la imagen por ser los que peor enfoque presentan. De esta manera se consigue, por un lado, aligerar el coste computacional del proceso siendo este mucho más rápido, y por otro, eliminar posibles imprecisiones por píxeles captados incompletos o deformados por la cámara web. Este filtrado nos lleva a tener una tabla de resultados como la que sigue:

Imagen 1				
YG				
182,496567	183,402374	183,931412	184,766327	185,261703
296,958374	297,452118	297,234039	297,808167	298,181366
410,242096	410,652374	410,711609	411,151917	411,589844
524,620483	524,425415	524,464966	524,687866	525,100342
637,922913	638,404663	638,255493	638,780457	639,285828
XG				
170,98584	285,420532	399,206055	512,955566	626,680603
170,128433	284,233765	398,099518	511,838837	625,634033
169,875656	283,598206	397,391022	510,889343	625,002991
169,383041	283,182159	396,912354	510,325195	624,215088
169,261124	282,998596	396,735382	510,201263	623,821899

Tabla 4.3 Posición de los centros de masa de los píxeles con los que se va a trabajar

- En este punto del ensayo es cuando se ponen en marcha los motores paso a paso. A través de un nuevo formulario, también ubicado en la hoja de cálculo, seleccionamos el desplazamiento deseado. Para este ensayo se ha planteado realizar desplazamientos de 0.5 mm y únicamente en el eje Y. Por lo tanto, en el formulario de la Figura 4.8 se marcarán las opciones positivas de desplazamiento en los ejes y se añadirá el desplazamiento de 0 mm en el eje X y de 0.5 mm en el eje Y.

Fig. 4.8 Formulario de movimiento de motores

- Cuando termine el movimiento del motor paso a paso asignado al eje X y, por lo tanto, el desplazamiento de la plataforma móvil asociada a ese eje se toma una medida de dicho desplazamiento con una máquina de medición por coordenadas. Finalmente se repite el procedimiento explicado en este ensayo desde el segundo punto hasta el séptimo, deteniéndose cuando la suma de los

desplazamientos alcance los 5 mm, haciendo un total de 11 imágenes, 10 de desplazamientos más la captura inicial.

- En esta parte del ensayo se trabaja con las fórmulas de la hoja de cálculo, restando las posiciones de una imagen con la inmediatamente anterior, de forma que se vayan calculando los desplazamientos producidos, medidos en píxeles. Estas diferencias se llevan a cabo entre posiciones en X y de forma independiente, diferencias entre posiciones en Y, dando lugar a una disposición de datos como la mostrada en la Tabla 4.4. La pérdida de columnas con respecto a la Tabla 4.3 se debe a que al estar tan cerca la cámara web de la pantalla LCD, un pequeño movimiento como el realizado en este ensayo causa que varias filas de píxeles iluminados ya no se capten, por lo que no son muchos los píxeles a los que se les puede hacer un seguimiento visual entre dos imágenes captadas por la cámara web.

Movimiento 1-2				
YG				
362,092361	361,902069	361,899704	361,23819	361,347855
363,141602	362,443756	362,388763	362,305786	362,185333

XG				
0,17243958	-0,59786987	-0,75665283	-1,16061401	-1,19842529
0,48170471	0,27941895	0,20114136	-0,37957764	-0,40789795

Movimiento 2-3				
YG				
364,605057	364,026794	363,478912	363,127243	362,613052
363,976227	363,936157	364,198334	364,205994	364,070862

XG				
-0,30461121	-0,98843384	-1,12780762	-1,62567139	-1,70654297
-0,14813232	-0,27508545	-0,32302856	-0,50939941	-0,78918457

Tabla 4.4 Desplazamiento medido en píxeles entre imágenes

- Aplicando a la ecuación (4.7) se obtiene cual ha sido el desplazamiento real en píxeles tomados por la cámara web que se ha producido en el movimiento de la plataforma móvil a través del motor paso a paso. El resultado de estas operaciones se muestra en la Tabla 4.5.

Avance real 1-2				
362,092403	361,902563	361,900495	361,240054	361,349842
363,141921	362,443864	362,388819	362,305985	362,185563

Avance real 2-3				
364,605184	364,028136	363,480662	363,130882	362,617068
363,976257	363,936261	364,198477	364,20635	364,071717

Tabla 4.5 Avance real medido en milímetros

• Mediante la ecuación (4.3) se calcula la mayor cantidad de distancias entre píxeles que sea posible, es decir, se calcula para cada píxel capturado en la imagen la distancia que le separa, en píxeles tomados por la cámara web, del siguiente píxel iluminado. Con todos esos resultados se obtiene la Tabla 4.6.

Distancia media entre píxeles iluminados 1-2				
YG				
114,461807	114,049744	113,302628	113,04184	112,919662
113,283722	113,200256	113,47757	113,34375	113,408478
114,378387	113,773041	113,753357	113,53595	113,510498
113,302429	113,979248	113,790527	114,09259	114,185486

XG				
114,434692	113,785522	113,749512	113,725037	
114,105331	113,865753	113,739319	113,795197	
113,722549	113,792816	113,498322	114,113647	
113,799118	113,730194	113,412842	113,889893	
113,737473	113,736786	113,465881	113,620636	

Distancia media entre píxeles iluminados 2-3				
YG				
113,877182	113,564453	112,883789	112,863266	112,449905
114,177094	113,404663	113,041016	113,608215	113,204315
112,322937	113,234314	114,367706	113,542358	113,773651
115,511047	114,591431	113,791687	114,109436	113,757141

XG				
114,107956	113,991486	114,033173	113,869629	
113,737076	113,974335	113,504181	113,922729	
113,721161	113,299194	113,697723	113,873413	
113,664383	113,62674	113,345551	113,687225	
113,903046	113,787476	113,1586	113,766876	

Tabla 4.6 Distancias entre píxeles iluminados

• Partiendo de la Tabla 4.6, se hace un promedio de todos los valores contenidos en dicha tabla, y aplicando las ecuaciones (4.4), (4.5) y (4.6) se consigue la Tabla 4.7 con la que es posible convertir las medidas en píxeles tomados por la cámara web en medidas en milímetros.

Promedio Total
113,6293397
Mallado
2
Píxeles/Píxeles reales
56,81466986
pulgada/mm
0,0393701
Píxel/Pulgada
326
Píxel Real/ mm
12,8346526
mm/Píxel
0,077914068
mm Distancia entre Píxeles Iluminados
0,155828137
Píxeles imagen/mm
729,1965503
Píxeles/0,5mm
364,5982751

Tabla 4.7 Tabla de conversión píxeles-milímetros

• De la Tabla 4.7, el valor final que es necesario para la conversión de medidas es la relación Píxeles de imagen/mm, si se toma cualquiera de las posiciones de los centros de masa de las Tablas 4.2 o 4.3 y se divide por esta relación, se obtendrá la misma posición del centro de masa pero expresado en milímetros. Realizando este procedimiento para los centros de masa de los píxeles iluminados de la Tabla 4.3 se obtendrá la Tabla 4.8, en la cual también se han añadido los valores promedio de desplazamiento real entre dos imágenes así como varios indicadores de dispersión

	Distancia real recorrida					Promedio	Rango	Varianza	Varianza *10 ⁶	Desviación típica
M 1-2	0,49656352	0,49630317	0,49630034	0,49539463	0,49554519	0,496567285	0,002608168	5,67931E-07	0,567930599	0,000753612
	0,4980028	0,4970455	0,49697001	0,49685642	0,49669127					
M 2-3	0,50000947	0,49921813	0,49846734	0,49798766	0,49728303	0,498939688	0,002726447	6,4579E-07	0,645789993	0,000803611
	0,49914698	0,49909213	0,49945173	0,49946252	0,49927789					
M 3-4	0,49529507	0,49341067	0,49257764	0,49192207	0,49203968	0,493606765	0,003373001	1,32175E-06	1,321745303	0,001149672
	0,49454237	0,49459893	0,49432311	0,49404679	0,49331131					
M 4-5	0,48788938	0,48744502	0,4872434	0,48721844	0,48675521	0,487583858	0,001371077	1,82901E-07	0,182900721	0,000427669
	0,48787448	0,48750943	0,48812628	0,48800589	0,48777105					
M 5-6	0,50071988	0,50005099	0,49957589	0,50015722	0,49950403	0,50026081	0,001414858	2,1053E-07	0,210529955	0,000458835
	0,50026924	0,5003498	0,50091889	0,50054304	0,50051912					
M 6-7	0,50429595	0,503746	0,5038536	0,50328426	0,50268841	0,503866579	0,001739403	2,88136E-07	0,288135564	0,000536783
	0,50442781	0,5038859	0,5042635	0,50430434	0,50391602					
M 7-8	0,49684615	0,49616629	0,49676533	0,49690836	0,49635348	0,496497245	0,001236183	1,42148E-07	0,142148403	0,000377026
	0,4963157	0,49660584	0,49664082	0,49669829	0,49567218					
M 8-9	0,49950219	0,50013973	0,50009951	0,50020967	0,49968061	0,499785756	0,001051109	1,58822E-07	0,158822152	0,000398525
	0,49926252	0,49938162	0,49992789	0,50031363	0,49934018					
M 9-10	0,51191862	0,51177273	0,5122212	0,51232155	0,51186037	0,511798276	0,000933417	9,07534E-08	0,090753436	0,000301253
	0,51138813	0,5115209	0,51168096	0,51179357	0,51150473					
	Media	Media	Media	Media	Media	Media	Media	Media	Media	Media
	0,498939688	0,001414858	2,1053E-07	0,210529955	0,000458835					

Tabla 4.8 Resultados del ensayo en lazo abierto. Ida

El ensayo experimental termina con la obtención de los resultados de la Tabla 4.8, ahora comienza su análisis. Sin embargo, antes de seguir analizando estos resultados, ya podemos tener algunas ideas de la precisión que consigue el método, no sólo observando la distancia promedio de cada movimiento, muy cercana a los 0.5 mm que se enviaba teóricamente a los motores; si no también al promedio total, los indicadores de dispersión que son la varianza y la desviación típica, y al rango de las diferencias entre píxeles de dos imágenes, donde todos los resultados son muy pequeños, indicando un alto grado de precisión.

Durante la realización del ensayo, como se indica en el séptimo punto de la explicación del procedimiento de ensayo, se ha ido tomando una medida del desplazamiento para cada movimiento realizado con una máquina de medición por coordenadas, también conocida como máquina de medición tridimensional. La precisión del sistema utilizado para este ensayo es de 0.018 mm.



Fig. 4.5 Máquina de medición por coordenadas

El siguiente paso, por lo tanto, será analizar los resultados de la tabla 4.8, comparándolos con las medidas tomadas durante el proceso por la máquina de medición por coordenadas, así como con los desplazamientos teóricos. Se calculan los errores en micras y redondeados para representar un resultado inmediatamente reconocible de la precisión obtenida en el ensayo:

Distancias(mm)			ERROR VB/EXCEL						
Teórica	Med. Por Coordenadas	VB/Excel	Error ponderado	MMC	Diferencia	Acum. Image	Acum. MMC	Diferencia	
0,5	0,495	0,496567285	-3	-5	-2	-3	-5	-2	
0,5	0,501	0,498939688	-1	1	2	-4	-4	0	
0,5	0,499	0,493606765	-6	-1	5	-10	-5	5	
0,5	0,487	0,487583858	-12	-13	-1	-22	-18	4	
0,5	0,511	0,50026081	0	11	11	-22	-7	15	
0,5	0,491	0,503866579	4	-9	-13	-18	-16	2	
0,5	0,499	0,496497245	-4	-1	3	-22	-17	5	
0,5	0,502	0,499785756	0	2	2	-22	-15	7	
0,5	0,51	0,511798276	12	10	-2	-10	-5	5	
MEDIA								4,5555556	

Tabla 4.9 Error medio acumulativo del ensayo. Ida

El error medio acumulativo cometido durante toda la carrera, esto es, sumando los errores de cada movimiento de 0.5 mm, comparado con el mismo error de la máquina de medición por

coordenadas, que habíamos indicado previamente, tiene una precisión de 1 μm , es de 4.56 μm . Este valor tan pequeño nos demuestra la gran precisión que puede alcanzar el método propuesto.

En la siguiente Figura se representa el ensayo en forma gráfica, para mostrar mejor las diferencias entre las medidas teóricas, las de la máquina medición por coordenadas y las obtenidas por el método de posicionamiento cámara pantalla.

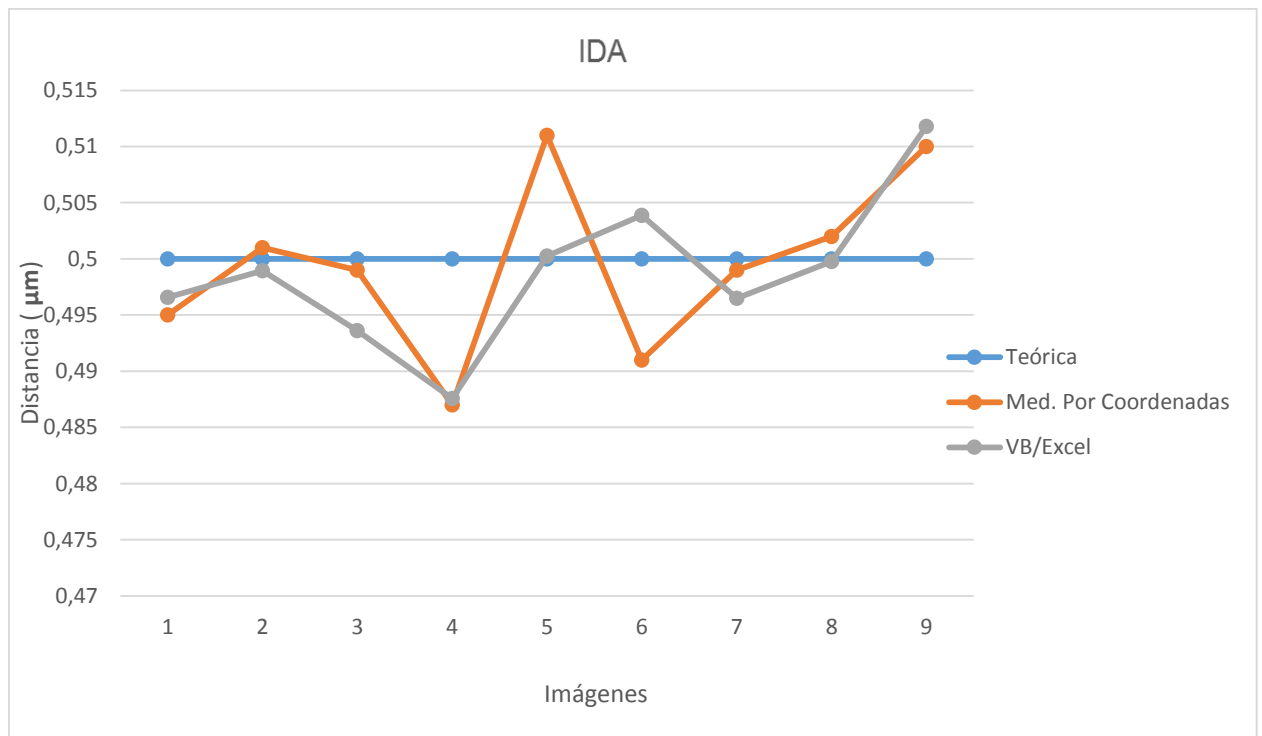


Fig. 4.6 Comparación gráfica de los resultados del ensayo. Ida

Para finalizar el ensayo en lazo abierto se realiza el mismo procedimiento explicado en este capítulo, pero esta vez desde el punto donde se detuvo la carrera anterior, hasta llegar al punto donde se inició la misma. Los datos obtenidos y cálculos realizados son análogos a los presentados durante el ensayo de ida, por lo que se expondrán únicamente las tablas de resultados finales (Tabla 4.10, Tabla 4.11 y Figura 4.7).

	Distancia real recorrida					Promedio	Rango	Varianza	Varianza *10 ⁶	Desviación típica
M 1-2	0,50157792	0,5006237	0,50036537	0,5007713	0,5003239	0,500333294	0,002545442	7,68032E-07	0,768031887	0,000876374
M 2-3	0,49594774	0,49589831	0,49625763	0,49594885	0,49601494	0,495435569	0,002671066	7,65731E-07	0,765730854	0,00087506
M 3-4	0,49596859	0,49564194	0,49611872	0,49560259	0,4952983	0,495148458	0,002295955	5,49129E-07	0,549128877	0,000741032
M 4-5	0,50056956	0,50090147	0,50089322	0,50116108	0,50084141	0,500435076	0,001788236	3,09873E-07	0,309872914	0,000556662
M 5-6	0,49871644	0,49916485	0,49923884	0,49922847	0,49927544	0,498687776	0,001560966	2,98616E-07	0,298615926	0,000546458
M 6-7	0,50207748	0,50165364	0,502236	0,50181681	0,50192822	0,501520801	0,002112739	3,99433E-07	0,399432764	0,000632007
M 7-8	0,49904399	0,49908957	0,49929154	0,49896901	0,49852134	0,498908179	0,000898195	9,84829E-08	0,098482892	0,00031382
M 8-9	0,49886431	0,49905477	0,49885281	0,49880087	0,49832261	0,498893727	0,001300399	1,17476E-07	0,11747643	0,000342748
M 9-10	0,5003756	0,49966417	0,49977	0,49923191	0,49849034	0,499339662	0,00188526	3,69269E-07	0,369268515	0,000607675
	0,49849347	0,49921374	0,49967895	0,49966325	0,4988152					
						Media	Media	Media	Media	Media
						0,498908179	0,00188526	3,69269E-07	0,369268515	0,000607675

Tabla 4.10 Resultados del ensayo en lazo abierto. Vuelta

Para esta prueba se aprecia que de nuevo, las medidas de desplazamiento real son muy cercanas a los valores teóricos de desplazamiento y como también, todos los indicadores de dispersión vuelven a arrojar promedios muy pequeños, sugiriendo un bajo error de precisión.

Al igual que para la carrera de ida, comparamos estos resultados con las medidas que se han ido tomando con la máquina de medición por coordenadas, obteniendo la tabla siguiente:

Distancias(mm)			ERROR VB/EXCEL						
Teórica	Med. Por Coordenadas	VB/Excel	Error ponderado	MMC	Diferencia	Acum. Image	Acum. MMC	Diferencia	
0,5	0,502	0,500333294	0	2	2	0	2	2	
0,5	0,496	0,495435569	-5	-4	1	-5	-2	3	
0,5	0,496	0,495148458	-5	-4	1	-10	-6	4	
0,5	0,49	0,500435076	0	-10	-10	-10	-16	-6	
0,5	0,509	0,498687776	-1	9	10	-11	-7	4	
0,5	0,503	0,501520801	2	3	1	-9	-4	5	
0,5	0,499	0,498908179	-1	-1	0	-10	-5	5	
0,5	0,501	0,498893727	-1	1	2	-11	-4	7	
0,5	0,493	0,499339662	-1	-7	-6	-12	-11	1	
MEDIA								2,7777778	

Tabla 4.11 Error medio acumulativo del ensayo. Vuelta

En esta ocasión los resultados del error acumulativo son incluso mejores que los obtenidos en la carrera de ida, quedando la media de la comparación entre el método cámara pantalla y la medición por coordenadas en 2.78 μm . Recordando que la precisión de la máquina de medición por coordenadas es de 18 μm , la precisión obtenida por el método propuesto está al mismo nivel o incluso supera la que proporciona la medición por coordenadas; lo que demuestra, en gran medida, la buena precisión que se puede conseguir con un posicionamiento cámara pantalla.

Finalmente se muestra la gráfica de comparación para las diferencias entre las medidas teóricas, la medición por coordenadas y las obtenidas por el método cámara pantalla:

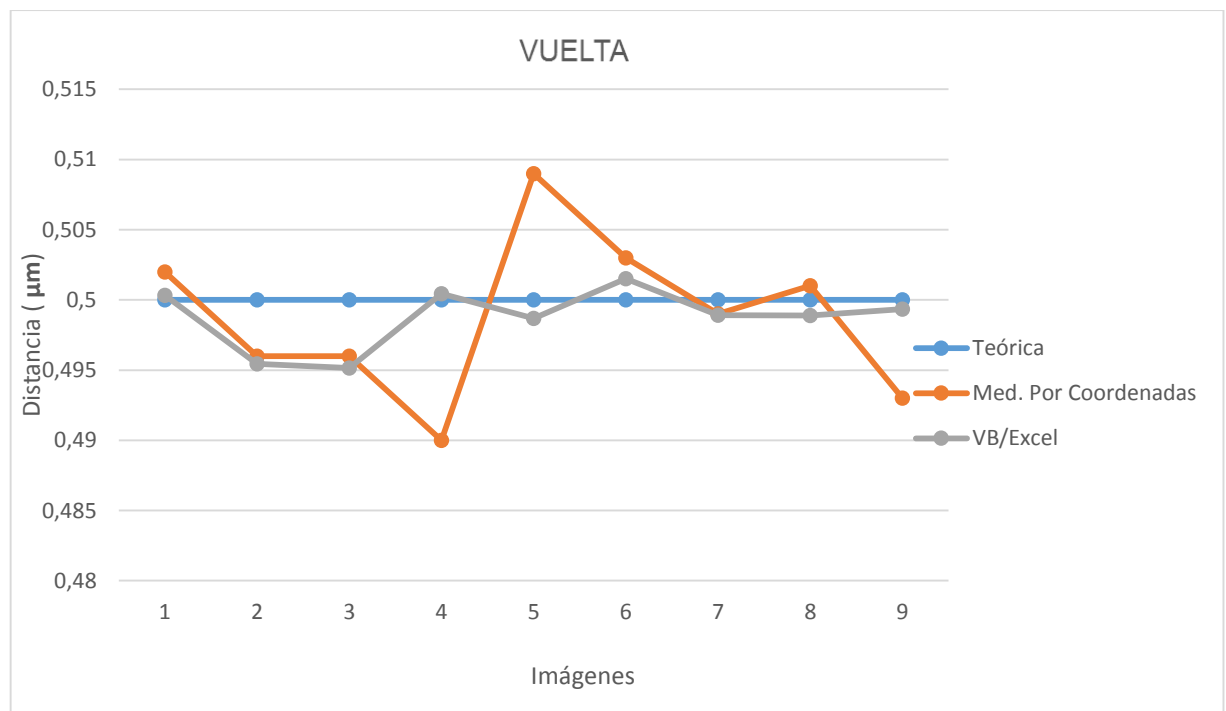


Fig. 4.7 Comparación gráfica de los resultados del ensayo. Vuelta

Puede observarse que, tal y como nos indicaba la tabla de error medio, muchas medidas son prácticamente idénticas mediante el método cámara pantalla y la medición por coordenadas, y en los casos con mayor diferencia, esta no supera las 10 μm . El método propuesto en este trabajo es, por lo tanto y potencialmente, de muy alta precisión.

4.5. Estudio de repetibilidad

Utilizando las medidas tomadas, es posible también llevar a cabo un estudio de repetibilidad, en el que se podrá observar la cercanía entre los resultados de mediciones sucesivas de la misma magnitud por medir, efectuadas en las mismas condiciones de medición. De esta forma es posible conocer las variaciones que aporta la máquina de medición por coordenadas que se está empleando durante este ensayo.

El propósito de este ensayo es, por lo tanto, saber cuál es el error que introduce la máquina de medición por coordenadas, para saber el grado de confianza que se puede tener en la comparación realizada en el apartado anterior con respecto a las medidas obtenidas por el sistema de posicionamiento cámara pantalla. Así como también, saber el error acumulado que se produce durante el proceso de ensayo en lazo abierto. La siguiente tabla (Tabla 4.12) expone los resultados de error en posicionamiento una vez introducidos los datos de las mediciones:

	1 Carrera Ida	1 Carrera Vuelta	2 Carrera Ida	2 Carrera Vuelta	3 Carrera Ida	3 Carrera Vuelta	MEDIA IDAS	MEDIA VUELTAS
Y	Error Posicionamiento Y (mm)						Error Posicionamiento Y (mm)	
0	0	0,000394035	0	0,002115055	0	0,005147954	0	0,002552348
0,5	-0,00343	0,000394035	-0,00686385	0,002115055	-0,01519694	-0,00944686	-0,00849784	-0,00231259
1	-0,00449	-0,00180744	-0,00976188	-8,64221E-05	-0,01579672	-0,01196342	-0,01001721	-0,00461909
1,5	-0,01089	-0,00204664	-0,01949925	-0,000325617	-0,01968091	-0,01080911	-0,01668881	-0,00439379
2	-0,0233	-0,00091703	-0,02048392	0,000803994	-0,02147838	-0,008702	-0,0217549	-0,00293834
2,5	-0,02304	-0,00257232	-0,01750931	-0,000851298	-0,02193561	-0,01114541	-0,02082884	-0,00485634
3	-0,01918	-0,00538102	-0,0125402	-0,003660002	-0,01868422	-0,01263135	-0,01679981	-0,00722412
3,5	-0,02268	-0,00266976	-0,01509507	-0,000948735	-0,01428692	-0,01207064	-0,01735326	-0,00522971
4	-0,02289	-0,00818959	-0,01838381	-0,006468567	-0,01911137	-0,01711516	-0,02012907	-0,01059111
4,5	-0,01109	-0,01074862	-0,01554965	-0,009027604	-0,0204841	-0,02114158	-0,01570916	-0,01363927
5	-0,01109	-0,01109374	-0,00937272	-0,009372718	-0,02123851	-0,02123851	-0,01390165	-0,01390165

Tabla 4.12 Error en posicionamiento Y

De acuerdo con la norma VDI/DGQ 3441 los parámetros de tolerancia de posición, la desviación posicional, de error de inversión, de la banda de dispersión de posicionamiento y la posición de incertidumbre se han determinado a partir de los datos de medición. La Figura 4.8 muestra un extracto del informe de calibración.

Entre los resultados cabe destacar el obtenido en Ps máx, que es el resultado de repetibilidad máxima, obtenido como el máximo valor entre los resultantes de restar el valor máximo y el mínimo para una misma posición de Y en cada carrera. Este resultado de 0.012 mm es muy próximo, e incluso

se encuentra por debajo, de la precisión teórica de la máquina de medición por coordenadas, por lo tanto sus medidas son perfectamente fiables como referencia.

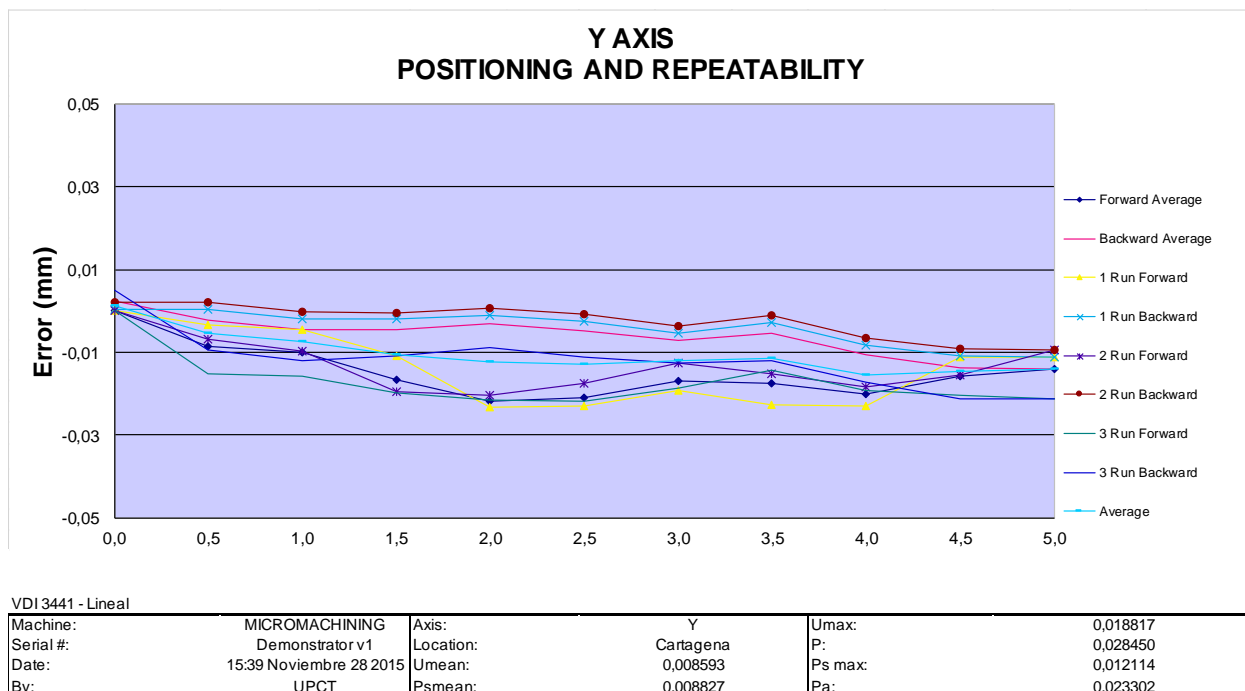


Fig. 4.8 Gráfica del estudio de repetibilidad

Los resultados acumulativos de la carrera, como puede apreciarse en la Figura 4.8, son muy cercanos entre sí. Esto indica un bajo error de posicionamiento por repetitividad de la medida, y a su vez, un bajo error acumulado, lo que reafirma que los resultados del ensayo en lazo abierto para el sistema de posicionamiento cámara pantalla son muy buenos e invitan a desarrollar un software para un sistema de posicionamiento con el método propuesto en lazo cerrado.

Capítulo 5

Desarrollo del algoritmo de control del hardware de un sistema de control en lazo cerrado

Una vez comprobada la precisión que puede alcanzar el método, está justificado el desarrollo de este sistema de posicionamiento en lazo cerrado. Pero antes de comenzar con el desarrollo del algoritmo de control, es necesario conocer cuál va a ser el funcionamiento del mismo, así como una serie de conceptos sobre los elementos que tomarán parte durante el proceso.

Como se puede observar en la Figura 5.2, el código deberá controlar por un lado la cámara web y las imágenes que toma, por otro lado la conversión de esas imágenes a valores RGB y finalmente el movimiento de los motores. A lo largo de los siguientes apartados se explica el funcionamiento del algoritmo y se incluyen esquemas que ayuden a comprender las distintas partes que conforman el código desarrollado.

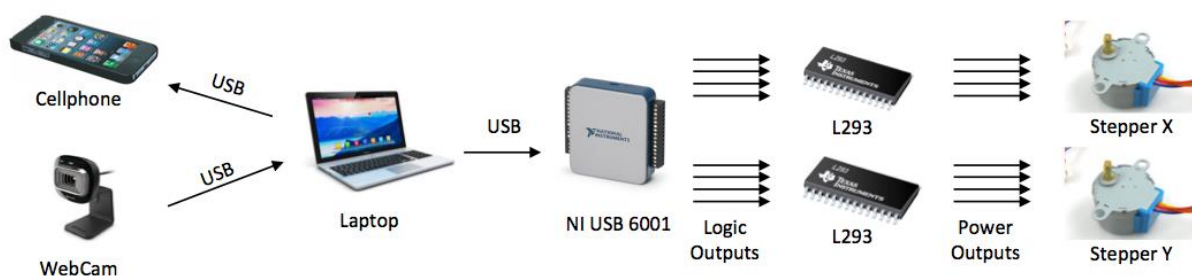


Fig. 5.1 Esquema de conexiones entre los diferentes elementos

5.1. Módulo de control de los motores paso a paso

En este módulo del código del algoritmo, se encuentra la parte encargada de recoger las órdenes de desplazamiento introducidas por el usuario del software, y convertirlas en señales lógicas con las que la tarjeta de adquisición de datos pueda trabajar.

Para llevar a cabo la comunicación del PC con los motores se utiliza la tarjeta NI USB 6001, junto con el circuito integrado L293B, como se explicó en el capítulo del desarrollo del hardware. Esto implica el uso de ciertas funciones específicas de las bibliotecas proporcionadas por el software comercial de National Instruments, por lo que es necesario la instalación de dicho software en el PC de trabajo para el correcto funcionamiento del proceso.

El funcionamiento de este módulo comienza con el formulario de la Figura 4.8, en el cual el usuario introduce el sentido, positivo o negativo, en el que quiere desplazar la plataforma móvil por el eje y los valores numéricos en milímetros que desea desplazar para cada eje. Los valores recogidos en este formulario se convierten en parámetros para la función más destacada de este módulo, denominada “Girarstepper”.

La función “Girarstepper” de este código es la rutina fundamental del módulo y está formada por otras subrutinas relacionadas con los puertos de la tarjeta de adquisición de datos. Estas subrutinas configuran una serie de variables que harán de puertas lógicas, activándose o no dependiendo de los parámetros recogidos en el formulario inicial. Para activar dichas variables se utilizan funciones IF, de manera que se crea una especie de selector de casos dependiendo de los sentidos que se hayan escogido para los ejes, y también de los ejes que se decida mover en el formulario.

Dentro de caso se aplican los desplazamientos especificados en el formulario convirtiendo el valor numérico introducido en número de pasos que debe dar el motor utilizando la relación de pasos por vuelta que da el fabricante, y que puede encontrarse en la hoja de especificaciones del motor paso a paso incluida en el anexo.

El software se encuentra funcionando en lazo cerrado, y por lo tanto, en etapas posteriores a la inicial en la que se introduce el desplazamiento a través del formulario, las entradas numéricas de los movimientos se obtienen de cálculos realizados en la hoja de cálculo, cuyo propósito es corregir los errores de posicionamiento al producirse los desplazamientos reales en los motores.

A continuación se presenta un flujograma del funcionamiento de este módulo de control de motores paso a paso:

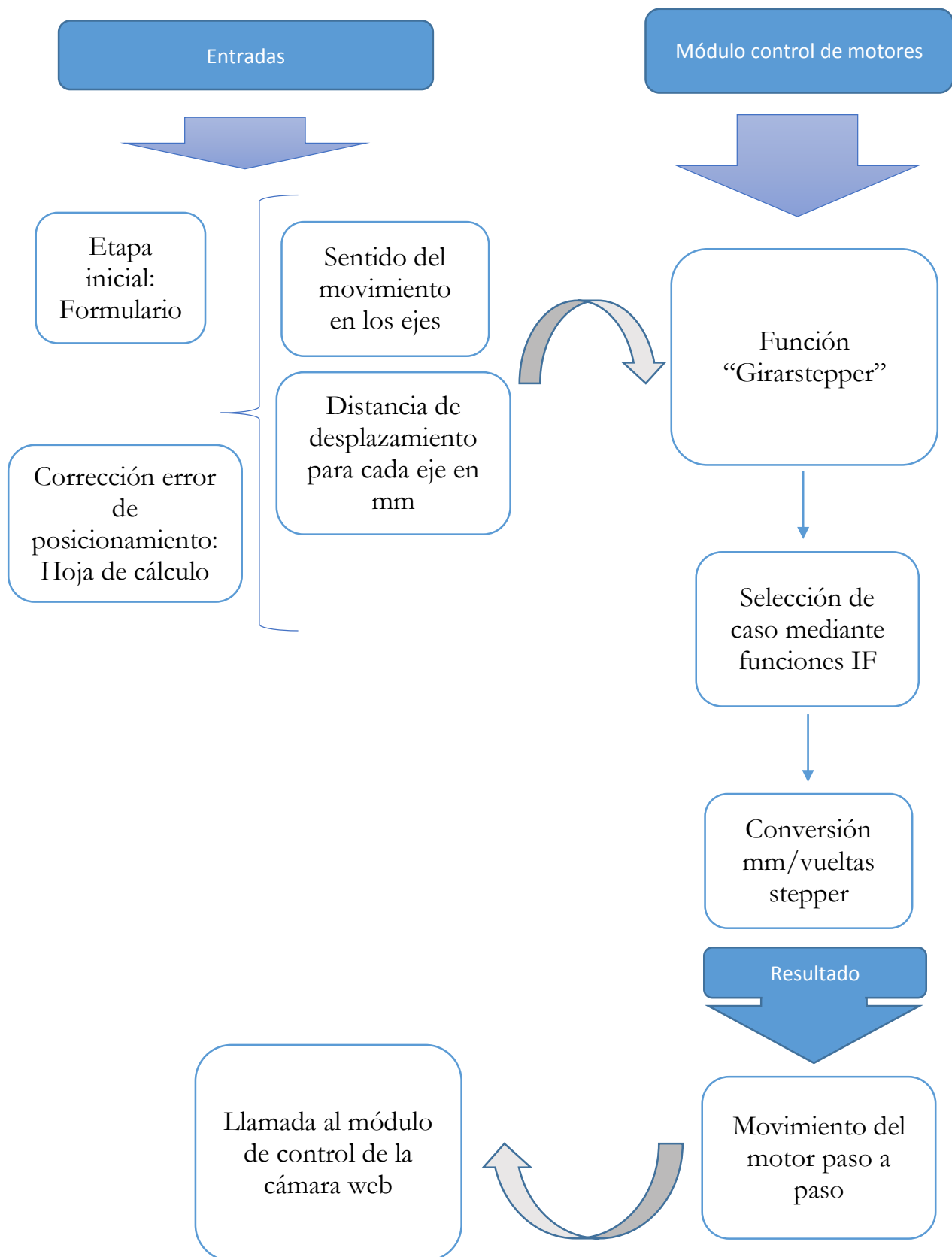


Fig. 5.2 Flujoograma del módulo de control de los motores paso a paso

5.2. Módulo de visión artificial

5.2.1. Módulo de control de la cámara web

Este apartado del código comprende las funciones que intervienen en la captura de la imagen de los píxeles en la pantalla, por parte de la cámara web, y su conversión a valores RGB. Este proceso de conversión es necesario para ser capaces, mediante otro de los módulos del software desarrollado, de calcular los centros de masa de los píxeles que son iluminados y ser capaces de comparar los de un movimiento con los de otro, de forma que se obtenga la distancia desplazada por los motores.

El módulo está compuesto inicialmente por declaraciones necesarias, principalmente para el proceso de conversión a valores RGB, que se detalla más adelante. La función “Snapshot” es la encargada de tomar la foto mediante la cámara web y enviarla a la función de conversión. Esta función denominada “Snapshot” llama a un archivo ejecutable de nombre “CommandCam”, de manera que se pueda tomar la captura de la cámara web sin intervención del usuario, pero que introduce un inconveniente en forma de limitación de la resolución hasta 640x480.

El significado de esta reducción de resolución es doble, por un lado la cámara web ve unos píxeles más pequeños, con lo que se puede perder precisión en el cálculo del centro de masa de los mismos, pero que como se verá más adelante, no supone ningún cambio drástico de los resultados que se han venido obteniendo. Por otro lado se gana en velocidad de cálculo, pues al trabajar con imágenes más pequeñas son muchas menos operaciones a realizar para obtener los centro de masa de los píxeles iluminados. Se reducen los cálculos puesto que significa tener menor densidad de píxeles, ya sean iluminados o no, a convertir en valores RGB, así como también una menor cantidad de sumatorios para calcular el centro de masa de un píxel iluminado, al captar cada uno de ellos con una menor cantidad de píxeles de la cámara web.

Al igual que sucedía en el módulo de control de los motores, la primera vez que se ejecute esta parte del algoritmo será una orden introducida por el usuario, mediante un formulario, la encargada de iniciar su funcionamiento. No obstante, cuando el proceso avance será el valor de corrección del error de posicionamiento el que vuelva a iniciar este módulo.

Finalmente, el módulo incorpora un último segmento correspondiente con el de la conversión de la imagen tomada por la cámara web en formato .bmp, en valores RGB. Este proceso consiste en aplicar a cada píxel de la imagen tomada por la cámara, un valor entre 0 y 255, siendo 0 la menos intensidad del color en ese píxel y 255 la máxima. Además, el proceso se lleva a cabo para cada uno de los tres colores fundamentales que conforman la imagen que son rojo, verde y azul, de ahí el nombre de valores RGB en inglés (red, green, blue).

Como los píxeles iluminados en la pantalla del dispositivo serán de color rojo, una vez obtenidas las tres hojas, correspondientes a los tres colores fundamentales mencionados, con los niveles de intensidad de color de 0 a 255 anteriormente explicados, se tomará la hoja que contenga los valores del color rojo para los posteriores cálculos.

Se incluye un flujograma de este módulo de visión artificial que controla la cámara web en la posterior Figura 5.3.

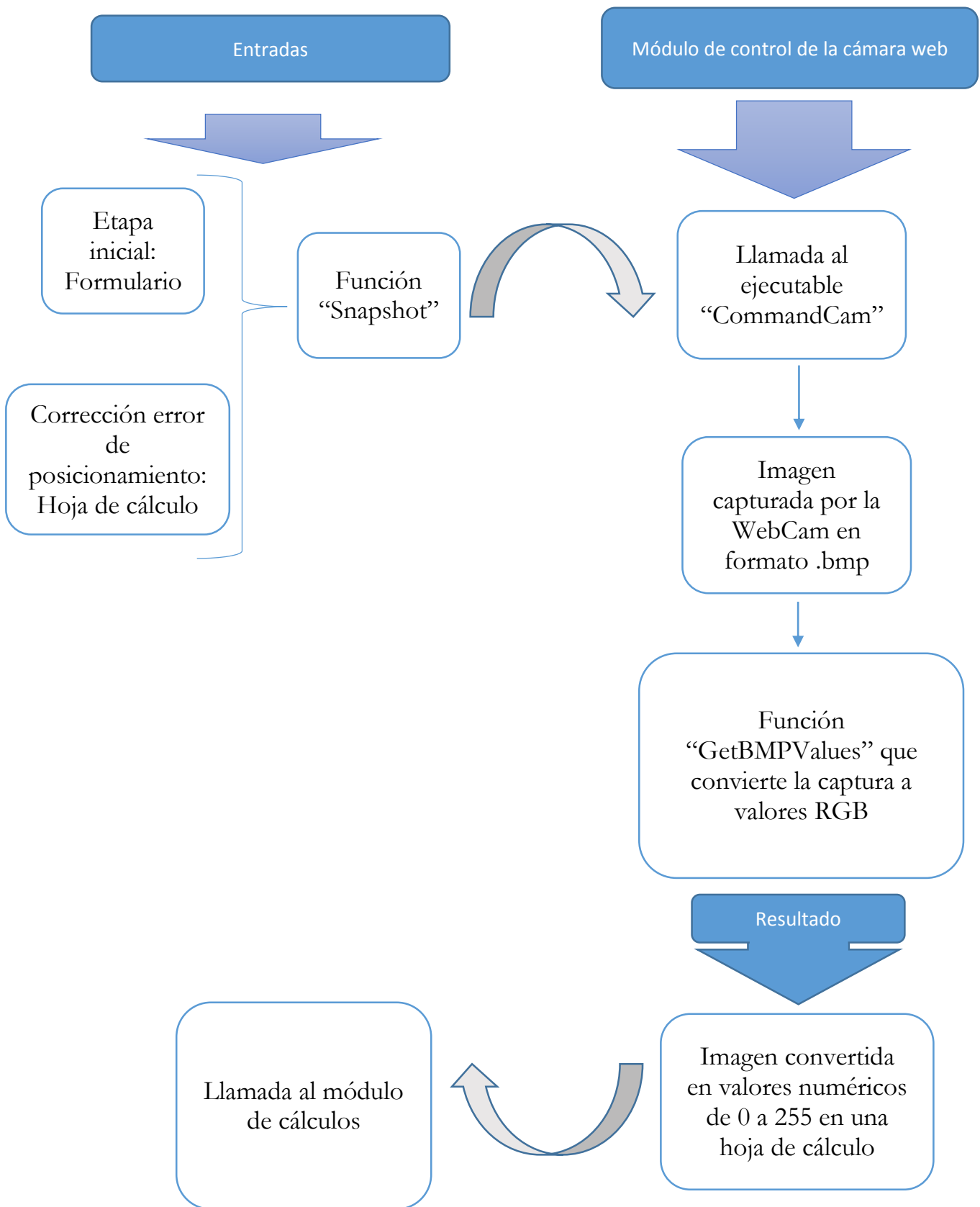


Fig. 5.3 Flujograma del módulo de control de la cámara web

5.2.2. Módulo de cálculos

Este módulo abarca los cálculos realizados en la imagen ya convertida en valores RGB, estos son, los cálculos de los centros de masa de los píxeles iluminados y captados en la imagen tomada por la cámara web.

El módulo de cálculos es puesto en marcha cuando una imagen tomada por la cámara web termina de ser convertida en valores RGB. Esa acción pone en funcionamiento otra función, en este caso llamada “EncontrarPixel” que comienza a recorrer la hoja de cálculo que contiene todos los valores RGB que dieron inicio a este módulo. Cada píxel de la captura tomada por la cámara web tiene, como ya fue explicado durante el apartado de funcionamiento del sistema de posicionamiento, un valor de intensidad lumínica en la hoja de cálculo, de modo que la función “EncontrarPixel”, formada por elementos If, comprueba cada uno de esos valores para ver si está o no por encima del valor que se ha determinado como referencia de intensidad. De esta forma cuando el valor de un píxel supera esa referencia se toma un área de examen aproximadamente del tamaño con el que suelen aparecer los píxeles iluminados en la imagen, pero que añade cierta holgura sobre ese. Para evitar calcular áreas donde no existen píxeles iluminados pero sí errores provocados por reflejos que dan lugar a valores RGB que superan la referencia, una vez encontrado un valor superior a la referencia se comprueban también los píxeles adyacentes para asegurar la existencia de un píxel iluminado en la pantalla LCD.

El área de examen seleccionada en la función anterior es enviada como parámetro de entrada para otra función denominada “CentroMasa” en la cual se aplican las ecuaciones (4.1) y (4.2) para obtener el centro de masa del área examinada, que resultará a su vez, en la posición del centro de masa del píxel iluminado en la pantalla LCD, capturado por la cámara web y hallado mediante la función “EncontrarPixel” de este módulo del algoritmo de control.

La función “CentroMasa” termina haciendo referencia a otra con el nombre de “Resultado”, que es la encargada de ir pegando en otra hoja de cálculo todas las posiciones de centros de masa de píxeles iluminados que se van encontrando en la imagen que tomó la cámara web.

De todos los resultados de posiciones de centros de masa que se obtienen, el algoritmo solamente copia y pega en la hoja de cálculo final, que incluye las operaciones vistas en el ensayo en lazo abierto, algunos de las posiciones de centros de masa calculadas y estas son seleccionadas mediante una serie de casos programados con If y ElseIf al final del módulo. Esto se debe a dos motivos, en primer lugar para evitar los bordes donde la calidad de la imagen puede no ser la idónea, y en segundo lugar para reducir los costes computacionales que se llevan a cabo al pegar dichos resultados en la nueva hoja de cálculo.

El módulo de cálculos continúa funcionando en bucle volviendo a la función “EncontrarPixel” cada vez que pega un resultado en la hoja de cálculo, hasta que se llega al final de la imagen capturada, marcado por la resolución de la cámara web durante la prueba. Por lo tanto, el bucle se detendrá cuando se llegue a la columna 640 y a la fila 480 de la hoja de cálculo que contiene los valores RGB.

En la siguiente Figura 5.4 se muestra un flujograma del módulo de cálculos sobre la imagen ya convertida en valores RGB.

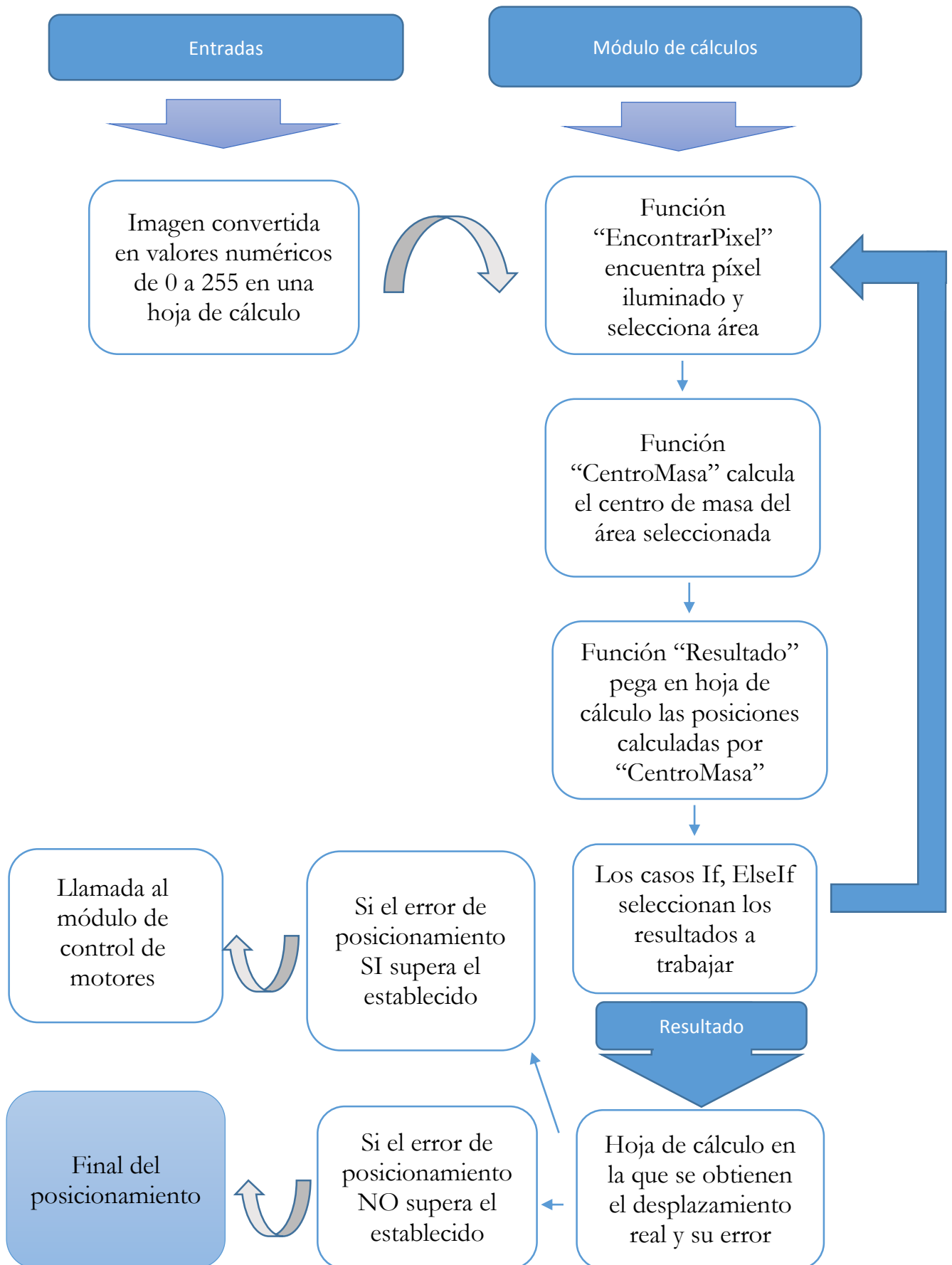


Fig. 5.4 Flujograma del módulo de cálculos

Para complementar el código anterior como se puede observar en el flujograma de la Figura 5.4, es necesario una serie de cálculos, transformaciones y comparaciones que se llevan a cabo en hojas de cálculo y no mediante operaciones en el código como hasta ahora se ha explicado.

Esas operaciones son las que comparan las distancias entre las posiciones de los centros de masa de los píxeles iluminados de la imagen que se captura inicialmente al comenzar el proceso, medidas en píxeles de la cámara web, con esas mismas posiciones en las imágenes que se toman una vez los motores ya han realizado desplazamientos. Además se obtiene una media de todas las distancias entre píxeles iluminados en la pantalla LCD, también medido en píxeles de la cámara web. Y finalmente también contiene la transformación a distancia métrica real de esas medidas, así como la del desplazamiento real recorrido y la comparación del error cometido entre la distancia recorrida deseada y la realmente desplazada, que es el error de posicionamiento.

En las siguientes Figuras 5.5 y 5.6 se muestra la disposición de la hoja de cálculo que lleva a cabo las operaciones citadas anteriormente.

Fig. 5.5 Hoja de cálculo. Centros de masa y distancias

Fig. 5.6 Hoja de cálculo. Conversión de unidades

A modo de corolario del algoritmo desarrollado se presenta la Figura 5.7 que muestra un pequeño esquema sobre la interacción existente entre los distintos módulos que conforman dicho algoritmo.

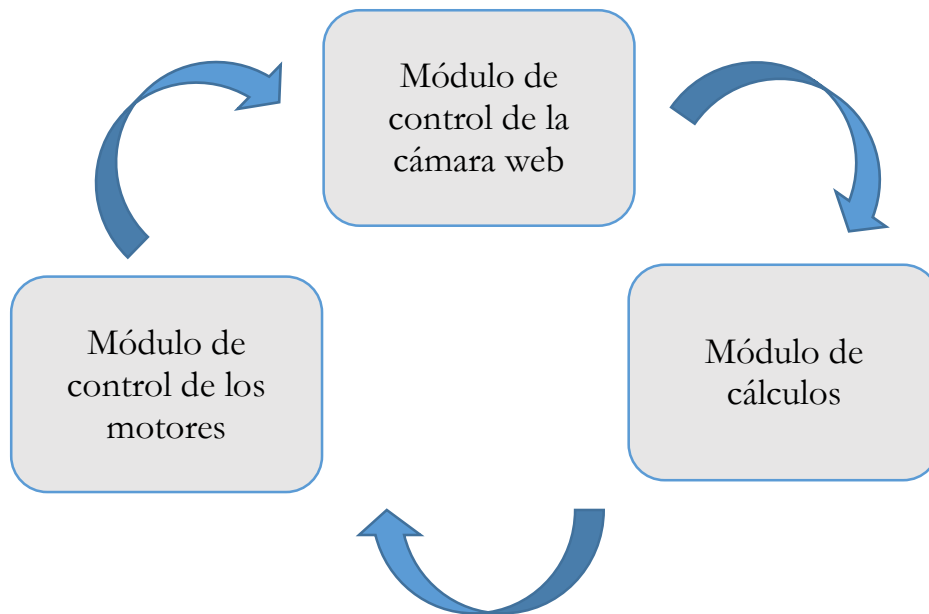


Fig. 5.7 Resumen funcionamiento del algoritmo

5.3. Uso mediante formularios

Los módulos del algoritmo anteriormente descritos son accesibles para el usuario mediante formularios, que a su vez tienen pequeños fragmentos de código simple.

Al arrancar el software, lo primero que el usuario de la aplicación recibe es el siguiente formulario denominado “UserForm1”:

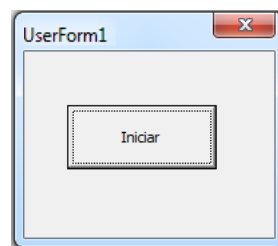


Fig. 5.8 Formulario inicio

Que incluye las siguientes líneas:

```
Private Sub SnapshotButton_Click ()  
Public Iteracion As Long  
Iteracion = 1  
Snapshot  
End Sub
```

Al pulsar el botón “Iniciar” ocurren dos cosas, la declaración e inicialización de la variable de iteración que rige el bucle final de comparación entre movimiento teórico y real, de manera que se consiga la precisión deseada; y la llamada al módulo que captura la imagen de los píxeles en la pantalla mediante la cámara web.

En ese momento comenzarán a funcionar los módulos detallados en los apartados anteriores y una vez tomada la foto mediante la cámara web, esta se convertirá a valores RGB. De estos valores se calcularán las posiciones de los centros de masa de los píxeles iluminados capturados en la imagen. A través de la conversión de esos resultados, que se han obtenido en distancias de píxeles de la cámara web, a distancia métrica se consigue como resultado la posición de referencia. Una vez que el programa ha terminado de realizar los procedimientos previos, automáticamente pone a disposición del usuario el siguiente formulario “UserForm2”:

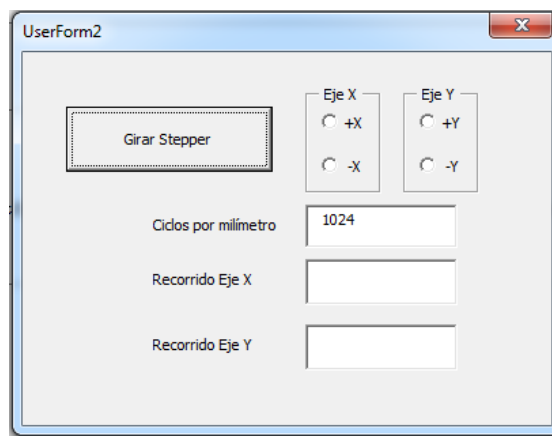


Fig. 5.9 Formulario de movimiento de motores

En este formulario lo que se muestra son las opciones disponibles para mover los motores paso a paso. En los botones de opciones superiores “Eje X” y “Eje Y” es posible elegir si la distancia que se quiere desplazar es positiva o negativa en cada eje, y a continuación, se elige el valor numérico de ese desplazamiento, es decir, cuánto se desea mover en cada eje, introduciendo los valores en las cajas “Recorrido Eje X” y “Recorrido Eje Y” respectivamente.

El cuadro de texto “Ciclos por milímetro” está por defecto fijado en 1024, debido a que es la cantidad de ciclos que ha de moverse el motor paso a paso para conseguir desplazarse un milímetro completo, y es un parámetro específico del motor que se está utilizando para este trabajo.

Finalmente, cuando se han escogido todos los parámetros, se pulsa el botón “Girar Stepper” lo que producirá el movimiento del hardware la distancia indicada, mediante los motores paso a paso. Al terminar el movimiento, automáticamente, se pondrá en marcha el mismo procedimiento que cuando se pulsó el botón “Iniciar” en el primer formulario (Figura 4.7).

En esta ocasión, al finalizar el cálculo de los centros de masa y obtener la posición métrica real, los datos pasarán a la hoja de cálculo mostrada en las figuras 5.5 y 5.6. Con el procedimiento en marcha esa hoja ya programada, contendrá los resultados de las comparaciones de los resultados de las posiciones antes y después del movimiento.

El valor promedio obtenido de avance real se contrasta con el valor introducido en el formulario “UserForm2”, si dicha diferencia es menor o igual de 0.002 mm, esto es, 2 μm , el desplazamiento se da por bueno, pues es suficientemente preciso. Sin embargo si la diferencia es mayor de esos 2 μm , entonces de forma automática, el programa envía a desplazarse al motor paso a paso dicha diferencia con el fin de compensarla.

En el supuesto de no conseguir la precisión deseada, comenzaría de nuevo el proceso que tenía lugar tras introducir los datos en el formulario “UserForm2”, se desplaza el hardware, la cámara web toma otra instantánea que vuelve a convertirse a valores RGB, de los que se obtiene la posición de los píxeles capturados a través del cálculo de sus centros de masa y conversión a distancia en milímetros y, por último, se pasan los datos a la hoja de cálculo que, nuevamente, vuelve a comparar la posición de partida, en este caso la resultante del primer desplazamiento, con la final, y a contrastarla con el desplazamiento requerido, ahora se tratará del desplazamiento que se solicitó automáticamente para corregir el error inicial de desplazamiento.

Este proceso tiene lugar hasta alcanzar un error de posicionamiento menor o igual a 2 μm , o hasta pasadas 5 iteraciones.

5.3. Visual Basic para aplicaciones

Para el desarrollo del código que controla el hardware de control se ha utilizado Visual Basic para aplicaciones. Esta elección se ha tomado, a pesar de sus limitaciones en cuanto potencia de procesamiento, por ser un entorno suficientemente sencillo y al alcance de cualquiera, puesto que la mayoría de los usuarios de productos ofimáticos están familiarizados con las soluciones de Microsoft Office.

El software Visual Basic para aplicaciones es una combinación de un entorno de programación integrado denominado Editor de Visual Basic y del lenguaje de programación Visual Basic, permitiendo diseñar y desarrollar con facilidad programas en Visual Basic. El término “para aplicaciones” hace referencia al hecho de que el lenguaje de programación y las herramientas de desarrollo están integrados con las aplicaciones del Microsoft Office (en este caso, Microsoft Excel), de forma que se puedan desarrollar nuevas funcionalidades y soluciones a medida, con el uso de estas aplicaciones.

El editor de Visual Basic contiene todas las herramientas de programación necesarias para escribir código en Visual Basic y crear soluciones personalizadas.

Este editor, es una ventana independiente de Microsoft Excel, pero tiene el mismo aspecto que cualquier otra ventana de una aplicación Microsoft Office, y funciona igual para todas las aplicaciones. Cuando se cierre la aplicación, consecuentemente también se cerrará la ventana del editor de Visual Basic asociada.

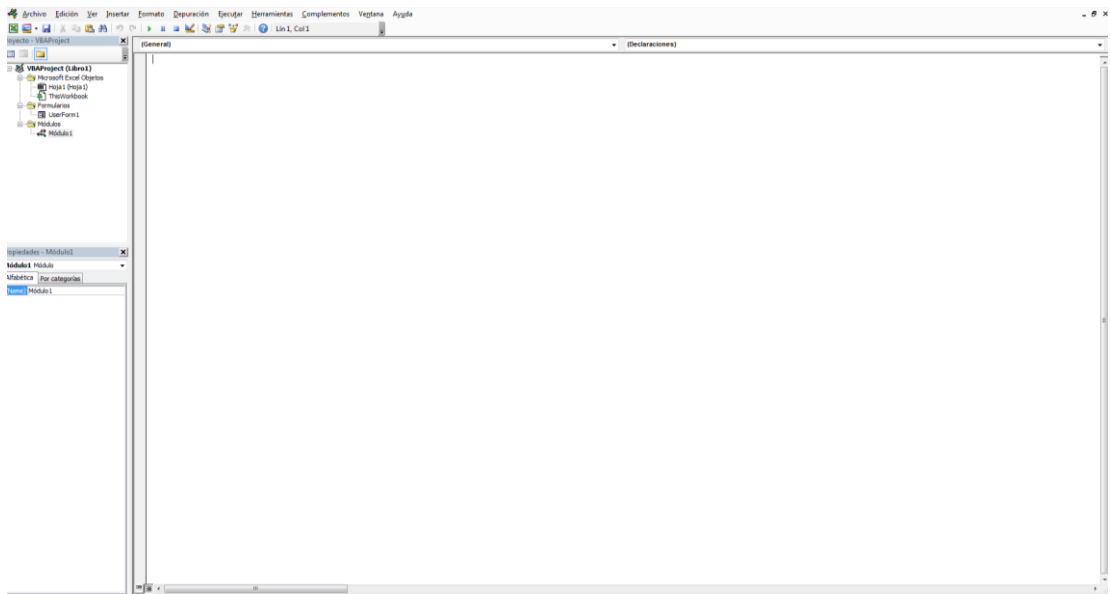


Fig. 5.10 Interfaz gráfica de usuario en Visual Basic para aplicaciones

Ensayo de posicionamiento en lazo cerrado

Ya se conoce el código desarrollado para el ensayo en lazo cerrado y como utilizarlo mediante los formularios para llevar a cabo la prueba, de manera que en este capítulo se exponen los resultados obtenidos una vez puesto en práctica el algoritmo explicado anteriormente.

El procedimiento es muy similar al explicado para el ensayo en lazo abierto, pero con algunas variantes que se explican a continuación:

- El ensayo comienza, al igual que en lazo abierto, mediante un formulario, pero ahora es el mostrado en la Figura 5.8, donde el usuario solamente dispone de una opción. Activar el botón “Iniciar” realiza el mismo procedimiento que en el ensayo en lazo abierto, es decir, captura de una primera imagen por la cámara web, transformación de esa imagen a valores RGB en una hoja de cálculo y cálculo de la posición de los centros de masa de los píxeles iluminados observados por la cámara web que posteriormente se pegan en otra hoja de cálculo que realiza las conversiones de distancia en píxeles a milímetros.

- Con la imagen de referencia completamente tratada, y como sucedía en el ensayo en lazo abierto, el siguiente paso es elegir el desplazamiento deseado aunque con la diferencia en este caso de que el formulario de la Figura 5.9 aparece de forma automática. La forma de proceder en los motores paso a paso es idéntica que la explicada para el ensayo en lazo abierto.

- El algoritmo, en este momento del ensayo, habrá realizado los cálculos para una segunda imagen tomada por la cámara web, de forma que se ha obtenido la distancia real que se ha desplazado la plataforma móvil. Para este ensayo en lazo cerrado se ha añadido un nuevo resultado en la hoja de cálculos, como se puede observar en la Figura 5.6 denominado “A corregir”. Este valor es el resultado de la diferencia entre el desplazamiento teórico indicado en el formulario para mover el motor paso a paso y el desplazamiento real obtenido mediante los cálculos del algoritmo de visión artificial.

- Es en esta parte del ensayo donde surgen las diferencias con el ensayo en lazo abierto, pues ahora no se sigue desplazando la plataforma móvil hasta terminar un carrera de determinada longitud, si no que la diferencia obtenida en el apartado de la hoja de cálculo “A corregir” es comparada con un valor de consigna que se ha situado en $2\ \mu\text{m}$. Si el error de posicionamiento cometido en el primer desplazamiento supera ese valor de referencia, el valor obtenido en el resultado “A corregir” se introduce automáticamente como parámetro de entrada para las funciones que gobiernan el formulario que controla los motores paso a paso, con el fin de subsanar el error de posicionamiento cometido.

- En caso de que el error de posicionamiento haya superado el valor establecido, el ensayo comenzaría de nuevo automáticamente desde el punto segundo de esta explicación, es decir, enviando un nuevo desplazamiento a los motores paso a paso. El procedimiento continuaría hasta que uno de los valores resultantes en “A corregir” en la hoja de cálculo fuese igual o menor a $2\ \mu\text{m}$ o hasta realizar 5 iteraciones del proceso.

Debido a que el programa funciona automáticamente, no es posible mostrar los resultados intermedios obtenidos antes de seleccionar las áreas de 5×5 de posiciones de centros de masa de píxeles iluminados, pues las imágenes totales convertidas a valores RGB y sus respectivos cálculos de centros de masa de todos los píxeles captados en la imagen, son sobrescritos con cada iteración del proceso y sólo dichas áreas seleccionadas se copian a otra hoja de cálculo distinta.

En el ensayo realizado el proceso encontró una precisión de $2\ \mu\text{m}$ o menos, tal y como se había indicado en la explicación del funcionamiento del software que era el límite establecido, tras dos iteraciones. Al no superar el límite propuesto de 5 iteraciones, el mensaje mostrado por el programa fue el mostrado en la Figura 6.1. En caso de no haber conseguido la precisión deseada, el mensaje resultante del ensayo habría sido distinto, avisando al usuario de que no se ha podido conseguir la precisión deseada en el desplazamiento.

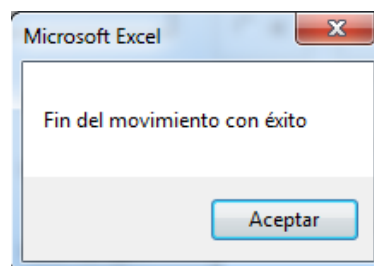


Fig. 6.1 Mensaje en pantalla final

A continuación, se muestran los resultados que se han ido obteniendo para los pasos del ensayo que se han explicado al comienzo del capítulo.

La siguiente tabla muestra cuales han sido los cálculos realizados durante el proceso:

Imagen 1				
YG				
111,506577	118,384476	118,763123	110,664238	116,387787
160,994614	163,439163	168,995453	169,305817	161,08786
210,422546	217,194016	218,134933	211,315338	214,605682
260,794556	266,049957	267,008118	261,509766	263,384064
309,826508	312,557678	314,498413	313,836578	309,993591
XG				
68,2824707	106,54509	155,374298	190,441727	224,764069
76,9953232	120,656273	160,261215	210,518158	245,582581
70,8023071	107,210335	155,239044	192,350754	226,841461
69,5074539	107,231956	154,820663	192,549194	227,540802
79,2030716	116,27623	157,929794	205,100204	238,873154

Imagen 2				
YG				
109,628052	110,068718	109,661514	109,254288	109,809677
160,425735	157,719376	156,901138	157,777374	159,006821
211,064514	206,623795	205,505173	207,623611	208,798889
259,767151	254,920197	253,663635	256,588623	258,370361
309,514618	304,369049	302,896545	305,956696	308,418213
XG				
69,8510132	105,091362	153,053711	192,971146	228,176529
69,6864929	105,659279	152,723969	193,273621	228,419174
69,3615189	105,752205	152,392731	193,411331	228,499908
70,4326859	105,699852	152,016022	193,921616	228,624069
70,1240158	105,573158	151,806595	193,899887	228,94989

Imagen 3				
YG				
110,126091	111,061401	109,988983	110,031685	109,549675
158,449173	158,307907	157,54277	158,40361	158,997269
210,492691	206,468338	205,598587	207,534073	208,369705
260,216461	255,310364	254,010437	256,68811	257,887573
309,642181	304,51947	303,233521	305,942871	308,067291
XG				
69,4551849	105,082085	153,670044	192,786179	227,622787
69,5326309	105,130539	153,243835	193,129196	227,858368
70,0220947	105,630615	152,917969	193,312973	227,758835
69,9753876	105,623817	152,562729	193,815186	228,211288
69,9775391	105,532196	152,377289	194,029495	228,305511

Tabla 6.1 Centros de masa de los píxeles utilizados

Partiendo de esos datos, el software utiliza el mismo procedimiento que se mostró en el ensayo de lazo abierto y cuyo desarrollo matemático está detallado en el capítulo de desarrollo del código, para obtener los siguientes resultados de avance en píxeles:

Diferencia 1-2				
YG				
148,260574	136,535721	134,900513	145,924385	141,982574
148,520004	140,929886	133,901093	136,650879	147,330353

Diferencia real 1-2				
148,276166	136,538337	134,942307	145,965872	142,035035
148,678871	141,734728	134,167743	137,657654	148,266244

XG				
2,15021515	-0,84523773	-3,35827637	3,47988892	3,86000061
-6,87130737	-15,0831146	-8,45462036	-16,6182709	-16,6326904

Diferencia 1-3				
YG				
148,709885	136,925888	135,247314	146,023872	141,499786
148,647568	141,080307	134,238068	136,637054	146,979431

Diferencia real 1-3				
148,71952	136,928987	135,276535	146,062834	141,541771
148,813133	141,888656	134,469383	137,628342	147,991386

XG				
1,69291687	-0,92127228	-2,81156921	3,37345886	3,44721985
-7,01778412	-15,1240768	-7,88392639	-16,4886627	-17,2770691

Tabla 6.2 Desplazamientos en píxeles

Aplicando la misma conversión que en el ensayo en lazo abierto, obtendremos las distancias que se desplaza el motor en cada movimiento, comparado siempre con la imagen de referencia que tomaba la cámara web al comienzo de la prueba, que como se explicó durante el apartado de funcionamiento, era lo primero que realizaba el software al pulsar el botón inicio. La tabla resultante se presenta a continuación:

	Distancia real recorrida					Promedio	Rango	Varianza	Varianza *10 ⁶	Desviación típica	A corregir
M1-2	0,52060951	0,4739705	0,47379326	0,51249788	0,4986964	0,497963496	0,050949733	0,000409008	409,0084538	0,020223957	0,002036504
	0,52202344	0,49764199	0,4710737	0,48332706	0,52057467						
M1-3	0,52216616	0,48076865	0,47496676	0,51283832	0,49696451	0,498334825	0,050362056	0,000398447	398,4471824	0,019961142	0,001665175
	0,52249484	0,49818245	0,47213279	0,48322415	0,51960962						
M3-4	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0						
M4-5	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0						

Tabla 6.3 Resultados ensayo en lazo cerrado

El error obtenido tras solamente dos iteraciones es de 1.66 μm . Este resultado de error de precisión tan bajo significa un grado de precisión altísimo del proceso de posicionamiento, lo que demuestra el gran potencial del método cámara pantalla propuesto durante el desarrollo de este trabajo.

Capítulo 7

Conclusiones y líneas futuras

En este trabajo se ha analizado la posibilidad de hallar un sistema de posicionamiento que simplifique los métodos actuales utilizados para tal fin, mediante el empleo de una pantalla y un algoritmo de detección de patrones de bordes. Además, también se ha estudiado el diseño del prototipo de máquina de micromecanizado para conseguir la mayor robustez del mismo.

Basándonos en los resultados obtenidos tanto del ensayo en lazo abierto, como especialmente, de los obtenidos en el ensayo en lazo cerrado, podemos ver cómo nos indican la altísima precisión potencial que poseen los sistemas de posicionamiento basados en el procesamiento de imágenes, y su bajo coste en relación con los métodos de posicionamiento frecuentemente utilizados hasta ahora.

Durante estas pruebas se utilizaron materiales de construcción del hardware de bajo coste, así como software también asequible, que a su vez, se encontraba instalado en un PC portátil de gama media. Los ensayos, si bien se llevaron a cabo con el mayor rigor posible, no fueron realizados en entornos controlados y el error obtenido en los mismos fue en todos los casos muy cercano a la micra.

Todas las consideraciones anteriores, además de arrojar resultados muy prometedores como se venía señalando, nos demuestran que es posible obtener precisiones con errores por debajo del micrómetro, utilizando el software desarrollado en Visual Basic para aplicaciones en un hardware de mayor fiabilidad que el utilizado durante estos ensayos.

Se podría conseguir una mayor fiabilidad empleando materiales más caros y de mayor calidad que los escogidos para estas pruebas, donde se podrían incluir tornillos guía de precisión, patines para el deslizamiento de los planos de mayor precisión o una cámara web con mejor enfoque y calibración. Por otro lado también se podría mejorar la fiabilidad realizando los ensayos en ambientes controlados, como ocurre en los movimientos de alta precisión de la industria.

Finalmente, y aunque el objetivo de demostrar la precisión que podía conseguir el uso de un método cámara pantalla sencillo se ha conseguido, la velocidad del proceso puede resultar insuficiente. Para esta situación es posible obtener mayores velocidades de procesamiento de los datos por parte del algoritmo y fluidez general del proceso instalando el software en un PC de mayores prestaciones que el portátil al que se ha recurrido durante estos ensayos, lo cual podría incluso conseguir reducir el error de precisión si dicho PC consiguiese trabajar con un número mayor de decimales. Esto último tiene importancia debido a que los errores de precisión en el posicionamiento que estamos alcanzando son del orden de unos pocos micrómetros.

La ampliación del estudio realizado en el presente Trabajo de Fin de Grado estaría enfocada en primer lugar a mejorar la potencia del software desarrollado. Para lograr este objetivo se podría recurrir a un lenguaje de programación con mayor capacidad que Visual Basic para aplicaciones. La alternativa, que significaría dar un paso más en este apartado, sería LabView. El programa LabView supone un gran aumento en el coste respecto a Visual Basic para aplicaciones, pero aporta una enorme mejora en el apartado de visión artificial, llevando ya implementado en su código un potente módulo que incluye reconocimiento de patrones, limpieza del ruido de la imagen, enfoque de la misma, etc.

Dichas ventajas en visión artificial, repercuten a su vez en otro de los puntos a mejorar comentado en las conclusiones extraídas de los ensayos realizados: el tiempo de ejecución del sistema de posicionamiento en lazo cerrado. El tiempo total que se necesita para realizar un movimiento, incluyendo las iteraciones necesarias para alcanzar la precisión deseada, se ve reducido en un alto grado al necesitar mucho menos tiempo para procesar las imágenes tomadas por la cámara web.

Se podrían ampliar también los estudios realizados en este trabajo utilizando herramientas características de los procesos de micromecanizado. De esta manera se probaría el método desarrollado en unas condiciones reales de funcionamiento, como podría ser un ensayo con herramienta láser.

En este caso, para llevar a cabo un ensayo en condiciones reales, se podría realizar una prueba en lazo cerrado en condiciones ambientales controladas donde, utilizando la herramienta láser, se realizaría un corte o muesca en una superficie una vez terminado el movimiento gobernado por el método de posicionamiento cámara pantalla. Las marcas en la superficie se medirían con herramientas lo suficientemente precisas, como pueden ser máquinas de medición por coordenadas, para contrastar y corroborar la precisión teórica que posee el sistema de posicionamiento. Teniendo de esta forma una comprobación experimental del método en condiciones de uso reales.

Capítulo 8

Bibliografía

- [1] D. Leviton. *Method and Apparatus for Two-dimensional Absolute Optical Encoding*. U.S. Patent 6,765,195 B1. (2002).
- [2] C. A. Montes, J. C. Zieger, C. Wong, L. Mears, T. Tucker. *2-D absolute positioning system for real time control applications*. ASPE Proceedings, Atlanta, GA, USA (2010).
- [3] D. Chu, H. Trujillo, E. Whitney, J. Young. *Rapid High Resolution Absolute 2-D Encoding by Low Resolution Digital Imaging of a Mathematically Generated 3-Tone Target*. ASPE Proceedings. Santa Clara, CA, USA (2006).
- [4] I. Oshita, H. Otsubo, M. Tsuboi, M. Sogabe, T. Iwashita, Y. Kakino, A. Matsubara. *Development of high precision machining center and its motion accuracy*. 2008 International Symposium on Flexible Automation, Atlanta, GA, USA (2008).
- [5] Y. Koren, *Computer control of manufacturing systems* (pp. 115-142). New York et al.: McGraw-Hill. (1983).
- [6] L. E. Sucar, G. Gómez. *Visión Computacional*. Publicado en internet. Página 39-51. <http://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf> (Visualizado el 14-06-2015).
- [7] E. Hildreth, D. Marr. *Theory of edge detection*. Proceedings of Royal Society of London, 207(187-217), (1980).
- [8] C. Solomon, T. Breckon. *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons. (2011).
- [9] J. P. García Sabater, G. Bravo I Reig. *Manual de Visual Basic para Excel*. (2003).

[10] ad hoc documentation and fast, instinctive documentation of ideas and solutions, *Running CommandCam from Excel using VBA*, <<https://batchloaf.wordpress.com/2013/01/07/running-commandcam-from-excel-using-vba/>>

[11] National Instruments, [Web], <<http://www.ni.com/>>

[12] MrExcel: Foro, [Web], <<http://www.mrexcel.com/forum/>>

[13] Access programmers: Foro, [Web], <<http://www.access-programmers.co.uk/forums/>>

[14] Ming Jong. [Web], <<http://www.mingjong.net/>>

[15] R. Z. Pedreño. *CNC con motor paso a paso: programación y control*. Universidad Politécnica de Cartagena. España (Murcia).

Capítulo 9

Anexos

1. SolidWorks
2. Hoja de especificaciones National Instruments USB 6001
3. Hoja de especificaciones L293B
4. Hoja de especificaciones motor ST28
5. Código del algoritmo de control
6. Planos

SolidWorks

El software SolidWorks

SolidWorks es una solución de diseño tridimensional completa que integra un gran número de funciones avanzadas para facilitar el modelado de piezas, crear grandes ensamblajes, generar planos y otras funcionalidades que le permiten validar, gestionar y comunicar proyectos de forma rápida, precisa y fiable.

SolidWorks se caracteriza por su entorno intuitivo y por disponer de herramientas de diseño fáciles de utilizar. Todo integrado en un único programa de diseño con más de 45 aplicaciones complementarias para facilitar el desarrollo de sus proyectos.

La característica que hace que SolidWorks sea una herramienta competitiva, ágil y versátil es su capacidad de ser paramétrico, variacional y asociativo, además de usar las funciones geométricas inteligentes y emplear un gestor de diseño (FeatureManager) que permite visualizar, editar, eliminar y actualizar cualquier operación realizada en una pieza de forma bidireccional entre todos los documentos asociados.

Características de SolidWorks

La definición de parámetros clave, la asociatividad, las funciones geométricas inteligentes y el gestor de diseño son las principales características de SolidWorks.

- Definición de parámetros clave

Los parámetros clave son las dimensiones (cotas) y las relaciones geométricas que definen un modelo tridimensional SolidWorks. Asocia a cada una de las cotas de un croquis, así como a las operaciones tridimensionales, un nombre que permite modificarla en cualquier momento y su actualización en el resto de documentos asociados.

- Asociatividad

SolidWorks contiene tres módulos: Pieza, Ensamblaje y Dibujo. La creación de un documento en cada uno de ellos genera un fichero con distinta extensión. Los documentos, aunque no puedan observarse, están asociados y vinculados entre ellos. En cada uno de ellos se disponen de múltiples herramientas de productividad, comunicación y análisis-simulación.

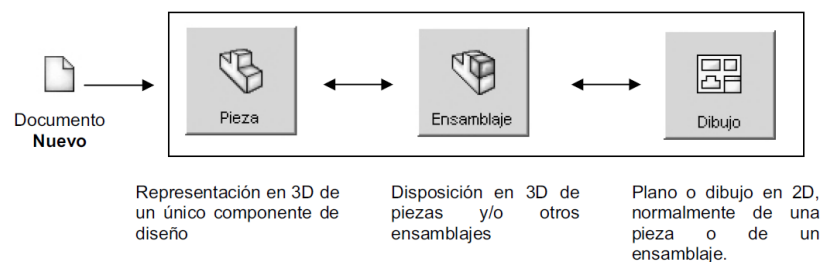


Fig. A.1 Módulos existentes en SolidWorks

Para realizar un conjunto o ensamblaje debe diseñar cada una de las piezas que lo conforman y guardar como ficheros de pieza distintos (cada uno con un nombre). El módulo de ensamblaje permite insertar cada una de las piezas y asignar relaciones geométricas de posición para definir tridimensionalmente el ensamblaje. Finalmente, puede obtener los planos las piezas o del propio ensamblaje de forma automática.

- Funciones geométricas inteligentes

La creación de Taladros, Chaflanes, Redondeos, Vaciados Nervios, entre otras operaciones, son creadas de forma rápida, ágil e intuitiva. En muchas operaciones el proceso de definición esta guiado y puede previsualizar la operación antes de su aceptación definitiva.

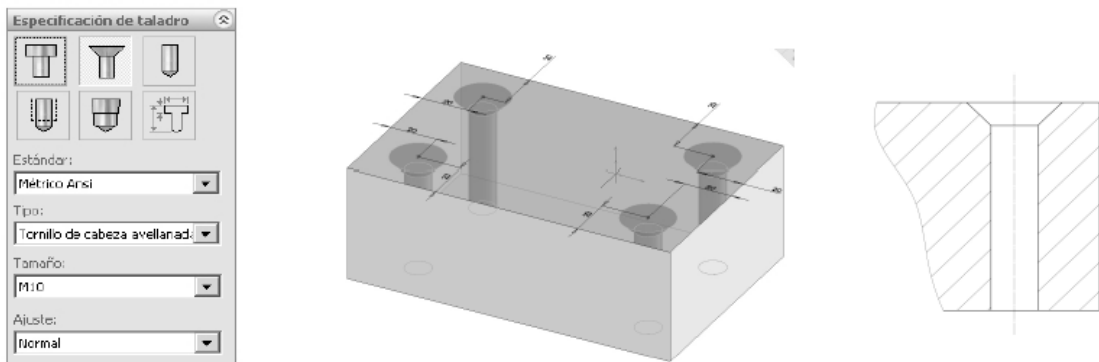


Fig. A.2 Función geométrica inteligente de taladro

- Gestor de diseño

También es conocido como árbol de operaciones o FeatureManager. En él se incluyen de forma histórica todas las operaciones que han sido necesarias efectuar para conformar la pieza durante su diseño. Las operaciones recientes se encuentran al final del árbol mientras que las más antiguas son las primeras en aparecer. El gestor de diseño permite visualizar/ocultar operaciones, suprimirlas o eliminarlas, cambiar el color y, lo que es más importante, modificar sus parámetros de definición. Puede establecer nuevas relaciones de posición geométrica o modificar la dimensión de una cota de croquis u operación.

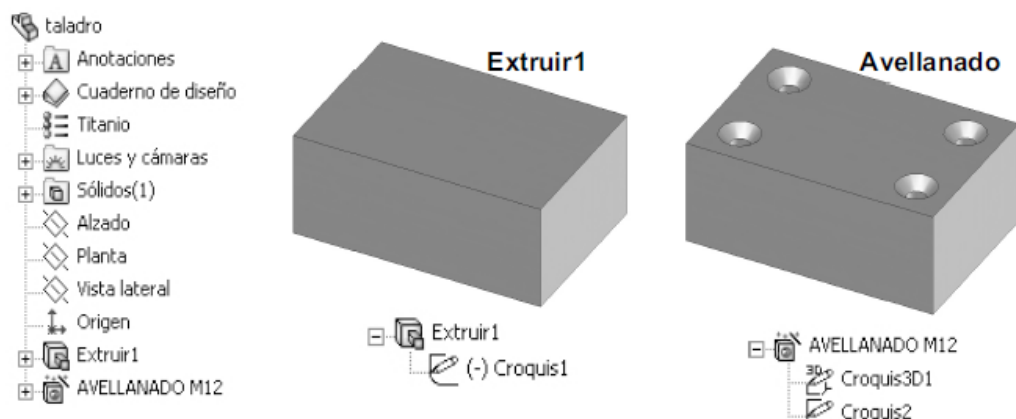


Fig. A.3 Gestor de diseño

Así, por ejemplo, si efectúa un taladro avellanado M12 a 10 mm de los lados de una placa puede modificar el tipo de taladro, su distancia a las caras y la profundidad del mismo en cualquier momento con sólo editar su operación desde el gestor de diseño.

Módulos de SolidWorks

SolidWorks contiene tres módulos: Pieza, Ensamblaje y Dibujo. En cada uno de ellos se disponen de múltiples herramientas de Productividad, Comunicación y Análisis-simulación.

Pieza

El módulo de pieza constituye un entorno de trabajo dónde puede diseñar modelos mediante el empleo de herramientas de diseño de operaciones ágiles e intuitivas. Su facilidad de uso se debe al empleo de un entorno basado en Microsoft Windows y en el uso de funciones clásicas como arrastrar y colocar, cortar y pegar o marcar y hacer clic con el ratón.

El conjunto de funciones e iconos permiten crear modelos tridimensionales (3D) partiendo de geometrías de croquis (2D) y obtener sólidos, superficies, estructuras metálicas, piezas de chapa, piezas multicuerpo, etc.

Los modelos creados se gestionan mediante el gestor de diseño dónde se incluyen todas las operaciones 3D y 2D utilizadas en la obtención de la pieza. Puede modificar operaciones sin necesidad de eliminar y volverlas a crear.

El módulo de pieza está totalmente integrado con el resto de módulos y funcionalidades de forma que cualquier cambio en su modelo 3D se actualiza en el resto de ficheros asociados (Ensamblajes, Dibujo, etc.) de forma bidireccional.

Ensamblaje

El módulo de ensamblaje está formado por un entorno de trabajo preparado para crear conjuntos o ensamblajes mediante la inserción de los modelos 3D creados en el módulo de pieza. Los ensamblajes se definen por el establecimiento de relaciones geométricas entre las piezas integrantes.

La creación de ensamblajes permite analizar las posibles interferencias o choques entre los componentes móviles insertados así como simular el conjunto mediante motores lineales, rotativos, resortes y gravedad y evaluar la correcta cinemática del conjunto.

Dibujo

Es el tercer módulo integrado en SolidWorks que permite crear planos con las vistas de los modelos o ensamblajes de forma automática y en muy poco tiempo. La obtención de las vistas, alzado, planta y perfil requiere únicamente pulsar sobre un icono o arrastrar la pieza 3D desde su ventana hasta la ventana del dibujo.

El módulo de dibujo permite obtener proyecciones ortogonales (vistas estándar), secciones y cortes, perspectivas, acotación, lista de materiales, vistas explosionadas, entre otras muchas funciones. Los documentos de dibujo están totalmente asociados a las piezas y ensamblajes de forma que cualquier cambio en ellas se actualizan en tiempo real en sus planos, sin tener que modificarlos de forma manual.

Conceptos básicos de SolidWorks

Las piezas son los bloques de construcción básicos en SolidWorks. Los ensamblajes contienen piezas u otros ensamblajes, denominados sub-ensamblajes.

Un modelo de SolidWorks consta de geometría en 3D que define sus aristas, caras y superficies. SolidWorks le permite diseñar modelos de forma rápida y precisa. Los modelos de SolidWorks:

- Están definidos por un diseño en 3D.
- Se basan en componentes.

Diseño en 3D

SolidWorks emplea un procedimiento de diseño en 3D. Al diseñar una pieza, desde el croquis inicial hasta el resultado final, está creando un modelo en 3D. A partir de este modelo, puede crear dibujos en 2D o componentes de relaciones de posición que consten de piezas o sub-ensamblajes para crear ensamblajes en 3D. También puede crear dibujos en 2D a partir de los ensamblajes en 3D. Cuando diseñe un modelo con SolidWorks, puede visualizarlo en tres dimensiones para ver su aspecto una vez fabricado.

Sistema basado en los componentes

Una de las prestaciones más versátiles de la aplicación SolidWorks es su capacidad para reflejar cualquier cambio realizado en una pieza en todos los dibujos o ensamblajes asociados a dicha pieza. Cuando se dice que SolidWorks es asociativo quiere decir que todos los documentos (Pieza, Ensamblaje o Plano) están vinculados y que la modificación de un fichero de pieza modifica el ensamblaje y los planos asociados de forma automática, sin la participación del usuario. Los ficheros se actualizan aunque se encuentren cerrados.

Interfaz de usuario y entorno gráfico

La aplicación SolidWorks contiene herramientas y posibilidades de interfaz de usuario que facilitan la creación y edición de los modelos de forma eficiente, entre las que se incluyen:

Funciones de Windows

SolidWorks contiene funciones conocidas de Windows, como la de arrastrar y cambiar el tamaño de las ventanas. Muchos de los iconos, como el de impresión, el de abrir, guardar, cortar y el de pegar, entre otros, también son parte de la aplicación SolidWorks.

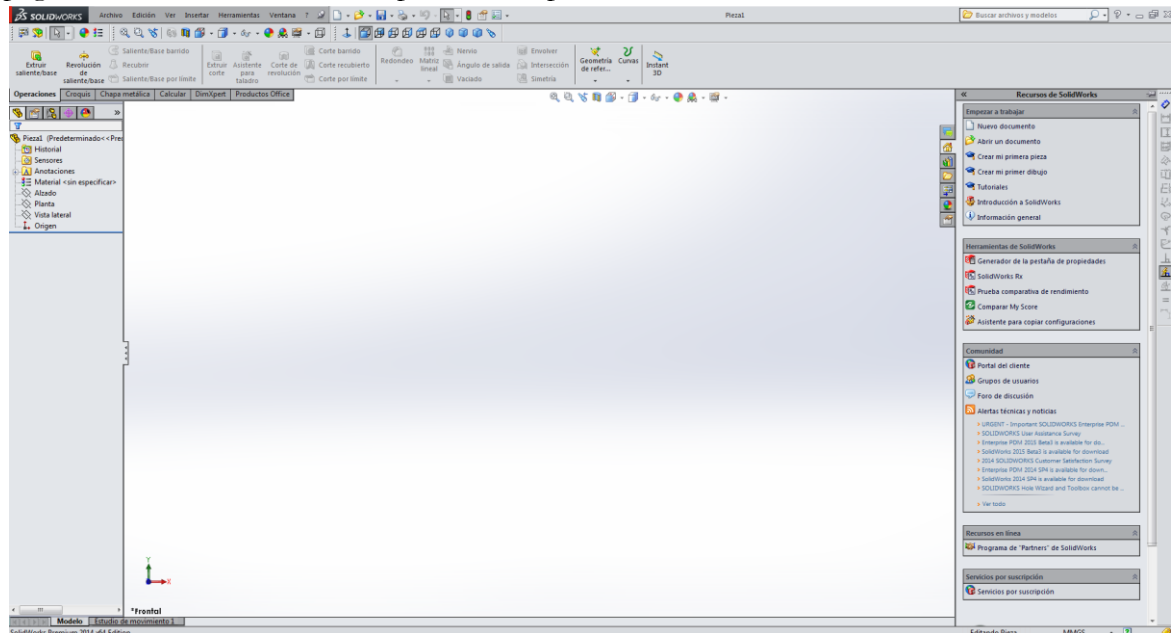


Fig. A.4 Interfaz gráfica de usuario en SolidWorks

Ventanas de documentos de SolidWorks

Las ventanas de documentos de SolidWorks tienen dos paneles. El panel izquierdo, o panel de gestión, contiene:

- Gestor de diseño del Feature Manager: se incluyen de forma histórica todas las operaciones que han sido necesarias efectuar para conformar la pieza durante su diseño. Las operaciones recientes se encuentran al final del árbol mientras que las más antiguas son las primeras en aparecer.

El gestor de diseño permite visualizar/ocultar operaciones, suprimirlas o eliminarlas, cambiar el color y, lo que es más importante, modificar sus parámetros de definición. Se pueden establecer nuevas relaciones de posición geométrica o modificar la dimensión de una cota de croquis u operación.

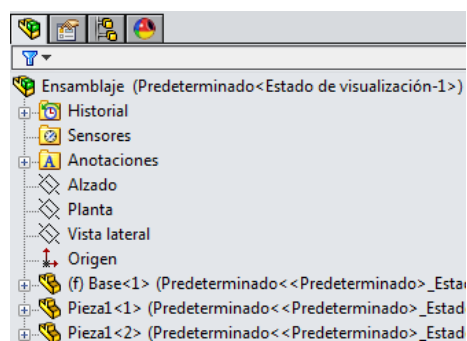


Fig. A.5 Feature Manager

- Property Manager: Brinda configuraciones para muchas funciones tales como croquis, operaciones de redondeo y relaciones de posición de ensamblaje.

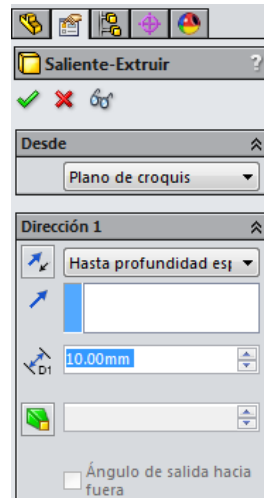


Fig. A.6 Property Manager

- Configuration Manager: permite crear, seleccionar y ver múltiples configuraciones de piezas y ensamblajes en un documento. Las configuraciones son variaciones de una pieza o ensamblaje dentro de un documento individual. Por ejemplo, se pueden usar configuraciones de un perno para especificar diferentes longitudes y diámetros.

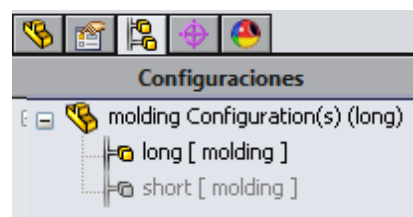


Fig. A.7 Configuration Manager

Menús

Es posible acceder a todos los comandos de SolidWorks a través de los menús. Los menús de SolidWorks utilizan las convenciones de Windows y presentan submenús y marcas de selección para indicar que un elemento está activo. También se pueden utilizar menús contextuales sensibles al contexto haciendo clic con el botón derecho del ratón.

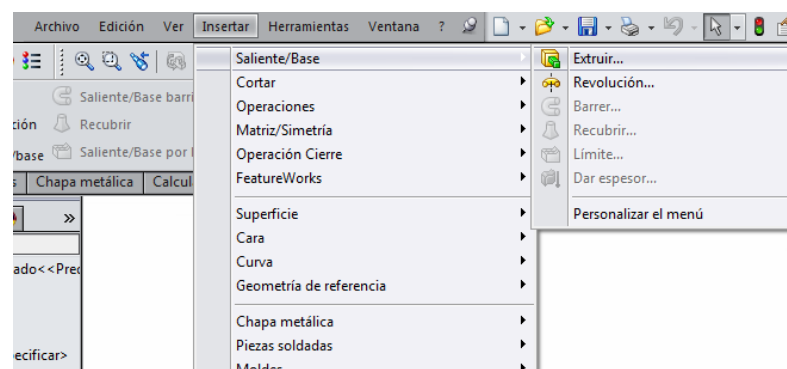


Fig. A.8 Menú

Barras de herramientas

Las barras de herramientas nos permiten acceder a todas las funciones de SolidWorks. Las barras de herramientas están organizadas según las funciones, por ejemplo, la barra de herramientas Croquis o Ensamblaje. Cada barra de herramientas contiene una serie de iconos exclusivos de herramientas específicas, como Girar vista, Matriz circular y Círculo.

Administrador de comandos

El Administrador de comandos es una barra de herramientas sensible al contexto que se actualiza dinámicamente en función del tipo de documento activo. Al hacer clic en una pestaña del administrador de comandos, este se actualiza para mostrar las herramientas relacionadas. Cada tipo de documento, como los documentos de pieza, ensamblaje o dibujo, tiene diferentes pestañas definidas para sus tareas. El contenido de las pestañas puede personalizarse, como sucede con las barras de herramientas. Por ejemplo, si hace clic en la pestaña Operaciones, aparecen las herramientas relacionadas con las operaciones. Si lo desea, también puede agregar o eliminar herramientas para personalizar el Administrador de comandos. La información sobre herramientas aparece al pasar el ratón sobre cada icono.

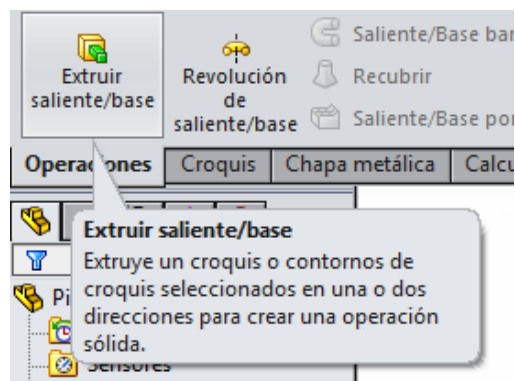


Fig. A.9 Administrador de comandos

Hoja de especificaciones National Instruments USB 6001

SPECIFICATIONS

NI USB-6001

Low-Cost DAQ USB Device

The following specifications are typical at 25 °C, unless otherwise noted. For more information about the NI USB-6001, refer to the *NI USB-6001/6002/6003 User Guide* available at ni.com/manuals.

Analog Input

Number of channels

Differential.....4

Single-ended.....8

ADC resolution.....14-bit

Maximum sample rate (aggregate).....20 kS/s

Converter type.....Successive approximation

AI FIFO.....2,047 samples

Trigger sources.....Software, PFI 0, PFI 1

Input range.....	±10 V
Working voltage.....	±10 V
Overvoltage protection	
Powered-on.....	±30 V
Powered-off.....	±20 V
Input impedance.....	>1 GΩ
Input bias current.....	±200 pA
Absolute accuracy	
Typical at full scale.....	6 mV
Maximum over temperature,.....	26 mV
full scale	
System noise.....	0.7 mVrms
DNL.....	14-bit, no missing codes
INL.....	±0.5 LSB
CMRR.....	56 dB (DC to 5 kHz)
Bandwidth.....	300 kHz

Analog Output

Analog outputs.....	2
DAC resolution.....	14-bit
Output range.....	±10 V
Maximum update rate.....	5 kS/s simultaneous per channel, hardware-timed
AO FIFO.....	2,047 samples
Trigger sources.....	Software, PFI 0, PFI 1
Output current drive.....	±5 mA
Short circuit current.....	±11 mA
Slew rate.....	3 V/μs
Output impedance.....	0.2 Ω

Absolute accuracy (no load)

Typical at full scale.....	9.1 mV
Maximum over temperature, full scale.....	34 mV

DNL..... 14-bit, no missing codes

INL..... ± 1 LSB

Power-on state..... 0 V

Startup glitch..... -7 V for 10 μ s

Timebase



Note The following specifications apply to the sampling accuracy for hardware-timed analog input and analog output.

Timebase frequency..... 80 MHz

Timebase accuracy..... ± 100 ppm

Timing resolution..... 12.5 ns

Digital I/O

13 digital lines

Port 0.....	8 lines
Port 1.....	4 lines
Port 2.....	1 lines

Function

P0.<0..7>.....	Static digital input/output
P1.0.....	Static digital input/output
P1.1/PFI 1.....	Static digital input/output, counter source or digital trigger

P1.<2..3>.....	Static digital input/output
P2.0/PFI 0.....	Static digital input/output, counter source or digital trigger
Direction control.....	Each channel individually programmable as input or output
Output driver type.....	Each channel individually programmable as open collector or active drive
Absolute maximum voltage range.....	-0.3 V to 5.5 V with respect to D GND
Pull-down resistor.....	47.5 k Ω to D GND
Power-on state.....	Input

Digital Input

Input voltage range (powered on).....	0 to 5 V
Input voltage range (powered off).....	0 to 3.3 V
Input voltage protection.....	± 20 V on two lines per port (maximum of five lines for all ports) for up to 24 hours



Caution Do not leave a voltage above 3.3 V connected on any DIO line for extended periods of time when the device is powered off. This may lead to long term reliability issues.

Minimum V_{IH}	2.3 V
Maximum V_{IL}	0.8 V
Maximum input leakage current	
At 3.3 V.....	0.8 mA
At 5 V.....	4.5 mA

Digital Output (Active Drive)

Maximum V_{OL} (4 mA).....	0.7 V
Maximum V_{OL} (1 mA).....	0.2 V
Minimum V_{OH} (4 mA).....	2.1 V
Minimum V_{OH} (1 mA).....	2.8 V
Maximum V_{OH}	3.6 V
Maximum output current per line.....	± 4 mA

Digital Output (Open Collector)

Maximum V_{OL} (4 mA).....0.8 V

Maximum V_{OL} (1 mA).....0.2 V



Note Minimum V_{OH} dependent on user-provided pull-up resistor and voltage source. Recommended pull-up resistor is 1 k Ω .

Using a 1 k Ω pull-up resistor and 5 V voltage source:

Minimum V_{OH}3.5 V

Typical V_{OH}4.5 V

Maximum output (sinking) current per line.....-4 mA

Maximum pull-up voltage.....5 V

Maximum leakage current

At 3.3 V.....0.8 mA

At 5 V.....4.5 mA

Counter

Number of counters.....1

Resolution.....32-bit

Counter measurements.....Edge counting, rising or falling

Counter direction.....Count up

Counter source.....PFI 0 or PFI 1

Maximum input frequency.....5 MHz

Minimum high pulse width.....100 ns

Minimum low pulse width.....100 ns

+5 V Power Source

Output voltage.....	+5 V, $\pm 3\%$
Maximum current.....	150 mA
Overcurrent protection.....	200 mA
Short circuit current.....	50 mA
Overvoltage protection.....	± 20 V

Bus Interface

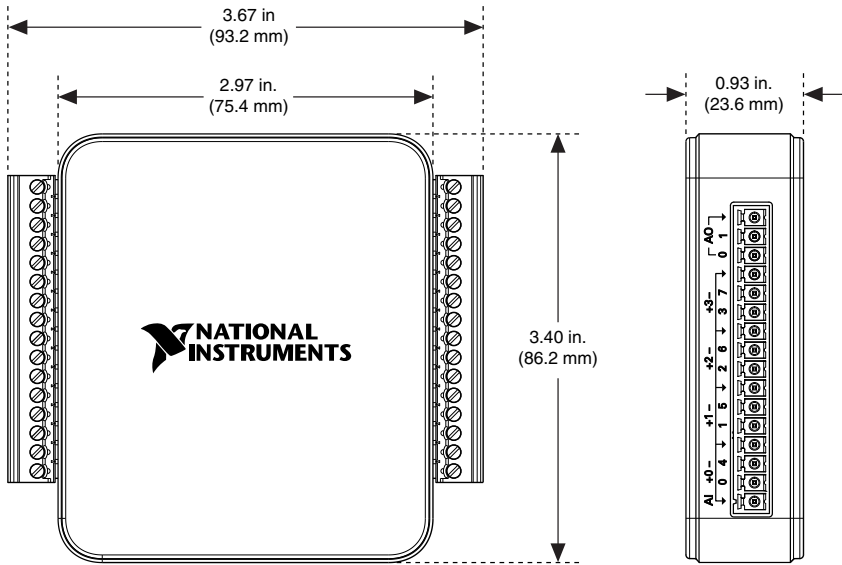
USB specification.....	USB Full Speed
USB bus speed.....	12 Mb/s

Physical Characteristics

Dimensions

Without screw terminal connector plugs....	75.4 mm \times 86.2 mm \times 23.6 mm, (2.97 in. \times 3.40 in. \times 0.93 in.)
With screw terminal connector plugs.....	93.2 mm \times 86.2 mm \times 23.6 mm, (3.67 in. \times 3.40 in. \times 0.93 in.)

Figure 1. NI USB-6001 Dimensions



Weight

Without screw terminal connector plugs....83 g (2.93 oz)

With screw terminal connector plugs.....105 g (3.70 oz)

I/O connectors: USB Micro-B receptacle, (1)

16-position screw terminal plugs

Screw-terminal wiring.....1.31 to 0.08 mm² (16 to 28 AWG)

Torque for screw terminals.....0.22 - 0.25 N · m (2.0 - 2.2 lb. · in.)

If you need to clean the module, wipe it with a dry towel.

Environmental

Temperature (IEC 60068-2-1 and

IEC 60068-2-2)

Operating.....0 to 45 °C

Storage.....-40 to 85 °C

Humidity (IEC 60068-2-56)

Operating.....5 to 95% RH, noncondensing

Storage.....5 to 90% RH, noncondensing

Pollution Degree (IEC 60664).....2
Maximum altitude.....2,000 m
Indoor use only.

Safety

This product meets the requirements of the following standards of safety for electrical equipment for measurement, control, and laboratory use:

- IEC 61010-1, EN 61010-1
- UL 61010-1, CSA 61010-1



Note For UL and other safety certifications, refer to the product label or the [Online Product Certification](#) section.

Electromagnetic Compatibility

This product meets the requirements of the following EMC standards for sensitive electrical equipment for measurement, control, and laboratory use:

- EN 61326-1 (IEC 61326-1): Class A emissions; Basic immunity
- EN 55011 (CISPR 11): Group 1, Class A emissions
- EN 55022 (CISPR 22): Class A emissions
- EN 55024 (CISPR 24): Immunity
- AS/NZS CISPR 11: Group 1, Class A emissions
- AS/NZS CISPR 22: Class A emissions
- FCC 47 CFR Part 15B: Class A emissions
- ICES-001: Class A emissions



Note In the United States (per FCC 47 CFR), Class A equipment is intended for use in commercial, light-industrial, and heavy-industrial locations. In Europe, Canada, Australia, and New Zealand (per CISPR 11) Class A equipment is intended for use only in heavy-industrial locations.



Note Group 1 equipment (per CISPR 11) is any industrial, scientific, or medical equipment that does not intentionally generate radio frequency energy for the treatment of material or inspection/analysis purposes.



Note For EMC declarations and certifications, and additional information, refer to the [Online Product Certification](#) section.

CE Compliance

This product meets the essential requirements of applicable European Directives, as follows:

- 2006/95/EC; Low-Voltage Directive (safety)
- 2004/108/EC; Electromagnetic Compatibility Directive (EMC)

Online Product Certification

To obtain product certifications and the DoC for this product, visit ni.com/certification, search by model number or product line, and click the appropriate link in the Certification column.

Environmental Management

NI is committed to designing and manufacturing products in an environmentally responsible manner. NI recognizes that eliminating certain hazardous substances from our products is beneficial to the environment and to NI customers.

For additional environmental information, refer to the *Minimize Our Environmental Impact* web page at ni.com/environment. This page contains the environmental regulations and directives with which NI complies, as well as other environmental information not included in this document.

Waste Electrical and Electronic Equipment (WEEE)



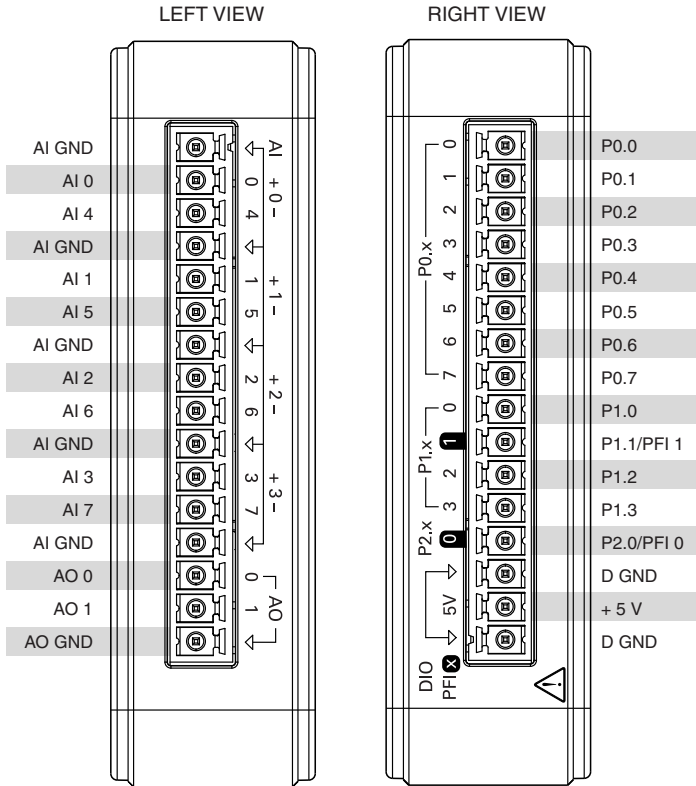
EU Customers This symbol indicates that waste products should be disposed of separately from municipal household waste according to WEEE Directive 2002/96/EC of the European Parliament and the Council on waste electrical and electronic equipment (WEEE). All products at the end of their life cycle must be sent to a WEEE collection and recycling center. Proper WEEE disposal reduces environmental impact and the risk to human health due to potentially hazardous substances used in such equipment. Your cooperation in proper WEEE disposal will contribute to the effective usage of natural resources. For information about the available collection and recycling scheme in a particular country, go to ni.com/environment/weee.

电子信息产品污染控制管理办法（中国 RoHS）



中国客户 National Instruments 符合中国电子信息产品中限制使用某些有害物质指令 (RoHS)。关于 National Instruments 中国 RoHS 合规性信息，请登录 ni.com/environment/rohs_china。(For information about China RoHS compliance, go to ni.com/environment/rohs_china.)

Device Pinout



Refer to the *NI Trademarks and Logo Guidelines* at ni.com/trademarks for information on National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at ni.com/patents. You can find information about end-user license agreements (EULAs) and third-party legal notices in the `readme` file for your NI product. Refer to the *Export Compliance Information* at ni.com/legal/export-compliance for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data. NI MAKES NO EXPRESS OR IMPLIED WARRANTIES AS TO THE ACCURACY OF THE INFORMATION CONTAINED HEREIN AND SHALL NOT BE LIABLE FOR ANY ERRORS. U.S. Government Customers: The data contained in this manual was developed at private expense and is subject to the applicable limited rights and restricted data rights as set forth in FAR 52.227-14, DFAR 252.227-7014, and DFAR 252.227-7015.

Hoja de especificaciones L293B

PUSH-PULL FOUR CHANNEL DRIVERS

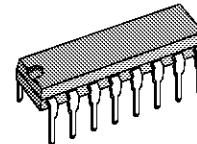
- OUTPUT CURRENT 1A PER CHANNEL
- PEAK OUTPUT CURRENT 2A PER CHANNEL (non repetitive)
- INHIBIT FACILITY
- HIGH NOISE IMMUNITY
- SEPARATE LOGIC SUPPLY
- OVERTEMPERATURE PROTECTION

DESCRIPTION

The L293B and L293E are quad push-pull drivers capable of delivering output currents to 1A per channel. Each channel is controlled by a TTL-compatible logic input and each pair of drivers (a full bridge) is equipped with an inhibit input which turns off all four transistors. A separate supply input is provided for the logic so that it may be run off a lower voltage to reduce dissipation.

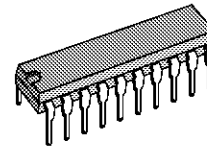
Additionally, the L293E has external connection of sensing resistors, for switchmode control.

The L293B and L293E are package in 16 and 20-pin plastic DIPs respectively ; both use the four center pins to conduct heat to the printed circuit board.



DIP16

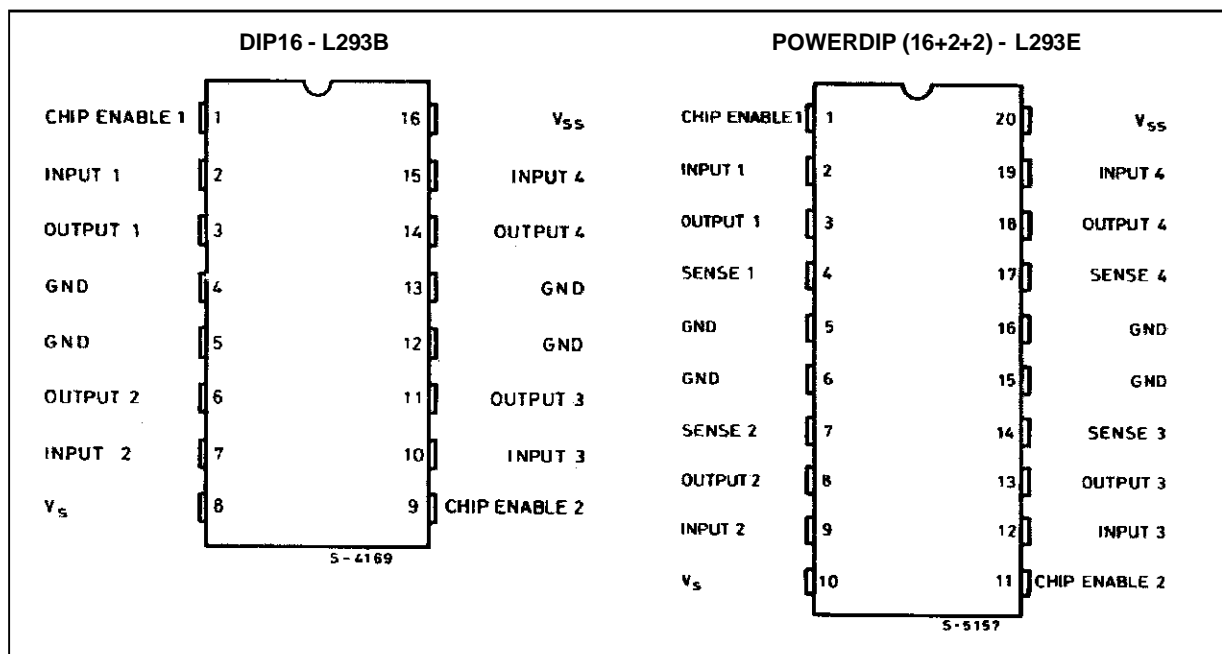
ORDERING NUMBER : L293B



POWERDIP (16 + 2 + 2)

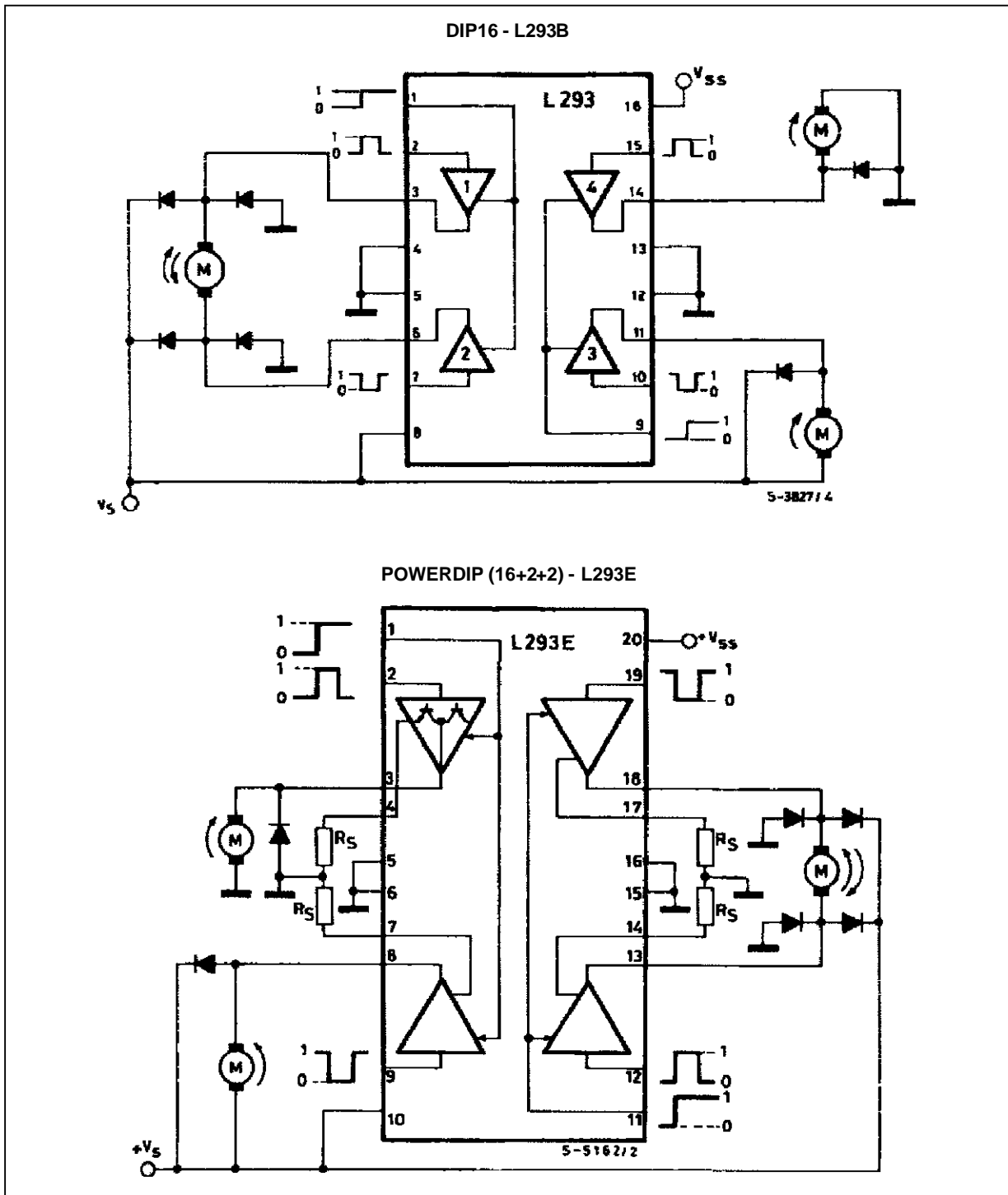
ORDERING NUMBER : L293E

PIN CONNECTIONS

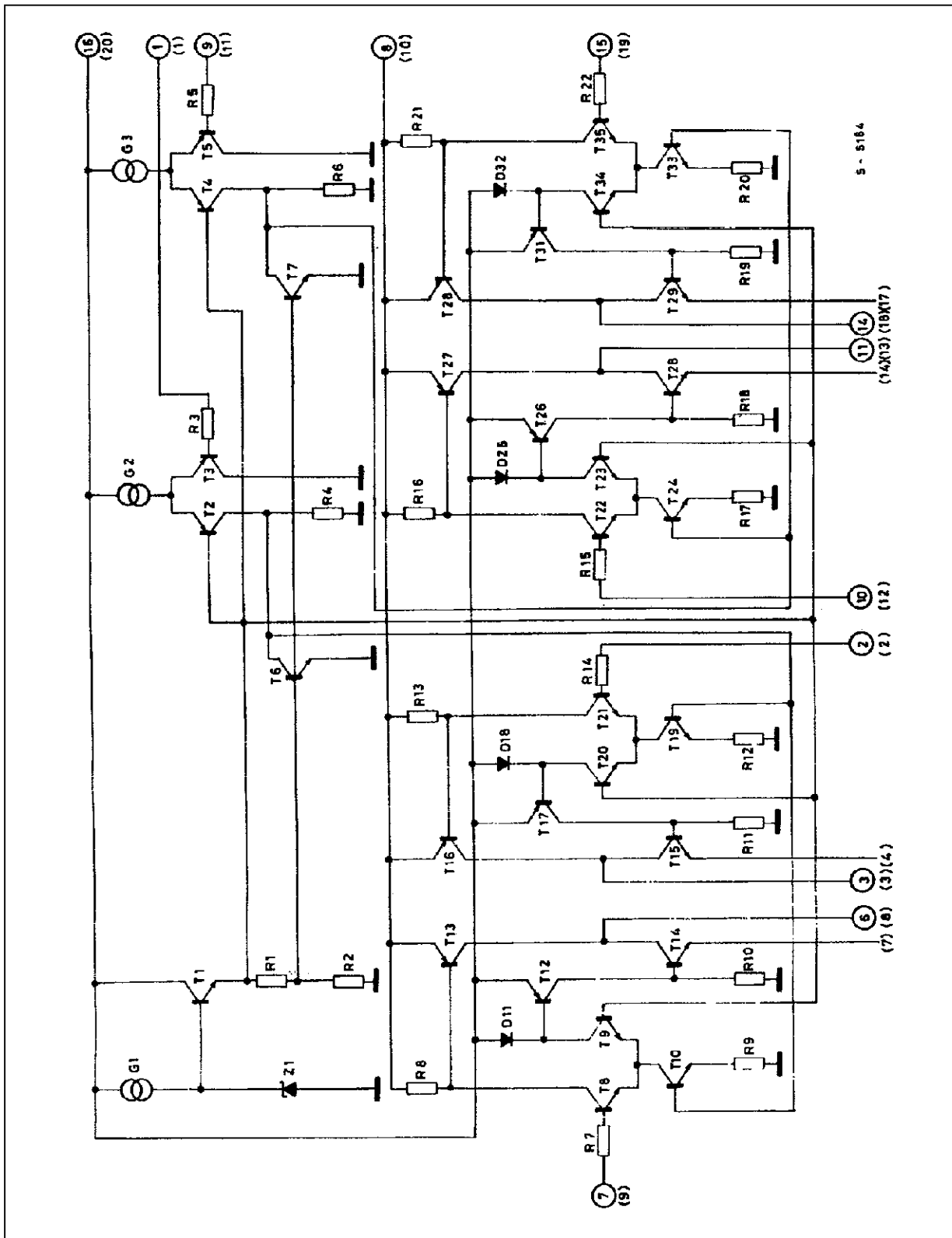


L293B - L293E

BLOCK DIAGRAMS



SCHEMATIC DIAGRAM



(*) In the L293 these points are not externally available. They are internally connected to the ground (substrate).
 O Pins of L293 () Pins of L293E.

L293B - L293E

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_s	Supply Voltage	36	V
V_{ss}	Logic Supply Voltage	36	V
V_i	Input Voltage	7	V
V_{inh}	Inhibit Voltage	7	V
I_{out}	Peak Output Current (non repetitive t = 5ms)	2	A
P_{tot}	Total Power Dissipation at $T_{ground-pins} = 80^\circ\text{C}$	5	W
T_{stg}, T_j	Storage and Junction Temperature	-40 to +150	$^\circ\text{C}$

THERMAL DATA

Symbol	Parameter	Value	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max. 14	$^\circ\text{C/W}$
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max. 80	$^\circ\text{C/W}$

ELECTRICAL CHARACTERISTICS

For each channel, $V_s = 24\text{V}$, $V_{ss} = 5\text{V}$, $T_{amb} = 25^\circ\text{C}$, unless otherwise specified

Symbol	Parameter	Test Conditions	Min.	TYp.	Max.	Unit
V_s	Supply Voltage		V_{ss}		36	V
V_{ss}	Logic Supply Voltage		4.5		36	V
I_s	Total Quiescent Supply Current	$V_i = L \quad I_o = 0 \quad V_{inh} = H$ $V_i = H \quad I_o = 0 \quad V_{inh} = H$ $V_i = H \quad I_o = 0 \quad V_{inh} = L$		2 16	6 24 4	mA
I_{ss}	Total Quiescent Logic Supply Current	$V_i = L \quad I_o = 0 \quad V_{inh} = H$ $V_i = H \quad I_o = 0 \quad V_{inh} = H$ $V_i = H \quad I_o = 0 \quad V_{inh} = L$		44 16 16	60 22 24	mA
V_{iL}	Input Low Voltage		-0.3		1.5	V
V_{iH}	Input High Voltage	$V_{ss} \leq 7\text{V}$ $V_{ss} > 7\text{V}$	2.3 2.3		V_{ss} 7	V
I_{iL}	Low Voltage Input Current	$V_{iL} = 1.5\text{V}$			-10	μA
I_{iH}	High Voltage Input Current	$2.3\text{V} \leq V_{iH} \leq V_{ss} - 0.6\text{V}$		30	100	μA
V_{inhL}	Inhibit Low Voltage		-0.3		1.5	V
V_{inhH}	Inhibit High Voltage	$V_{ss} \leq 7\text{V}$ $V_{ss} > 7\text{V}$	2.3 2.3		V_{ss} 7	V
I_{inhL}	Low Voltage Inhibit Current	$V_{inhL} = 1.5\text{V}$		-30	-100	μA
I_{inhH}	High Voltage Inhibit Current	$2.3\text{V} \leq V_{inhH} \leq V_{ss} - 0.6\text{V}$			± 10	μA
V_{CEsatH}	Source Output Saturation Voltage	$I_o = -1\text{A}$		1.4	1.8	V
V_{CEsatL}	Sink Output Saturation Voltage	$I_o = 1\text{A}$		1.2	1.8	V
V_{SENS}	Sensing Voltage (pins 4, 7, 14, 17) (**)				2	V
t_r	Rise Time	0.1 to $0.9 V_o$ (*)		250		ns
t_f	Fall Time	0.9 to $0.1 V_o$ (*)		250		ns
t_{on}	Turn-on Delay	$0.5 V_i$ to $0.5 V_o$ (*)		750		ns
t_{off}	Turn-off Delay	$0.5 V_i$ to $0.5 V_o$ (*)		200		ns

* See figure 1

** Referred to L293E

TRUTH TABLE

V_i (each channel)	V_o	$V_{inh}^{(\infty)}$
H	H	H
L	L	H
H	X ^(*)	L
L	X ^(*)	L

(*) High output impedance

(**) Relative to the considerate channel

Figure 1 : Switching Timers

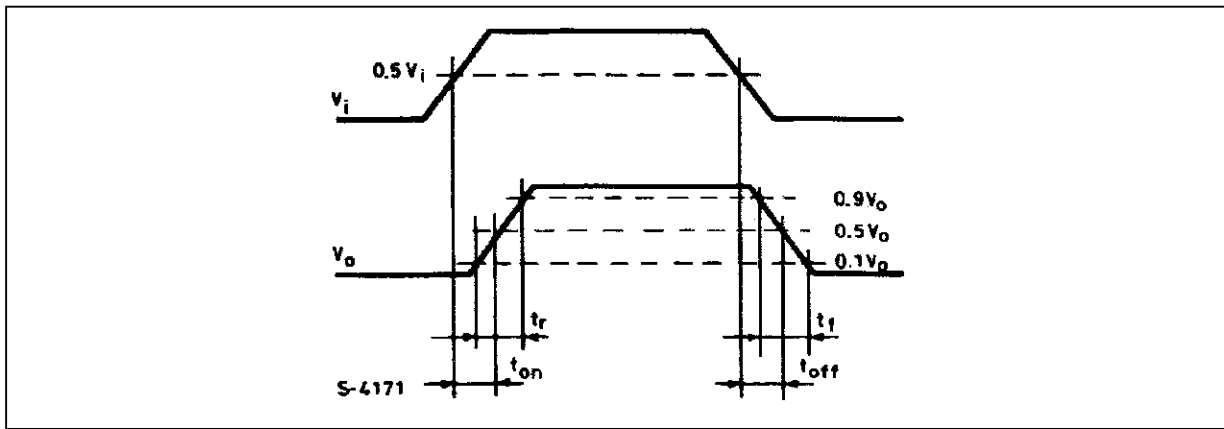


Figure 2 : Saturation voltage versus Output Current

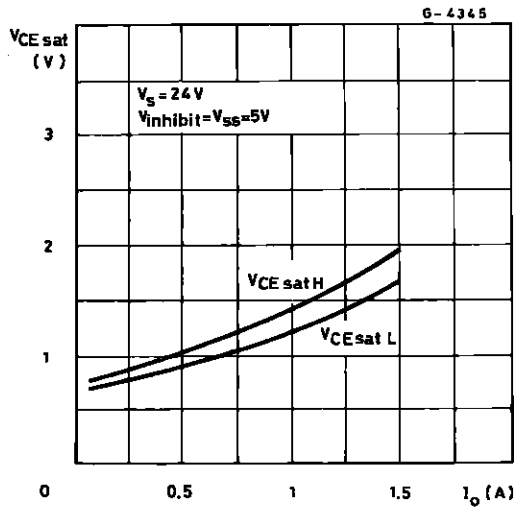


Figure 4 : Sink Saturation Voltage versus Ambient Temperature

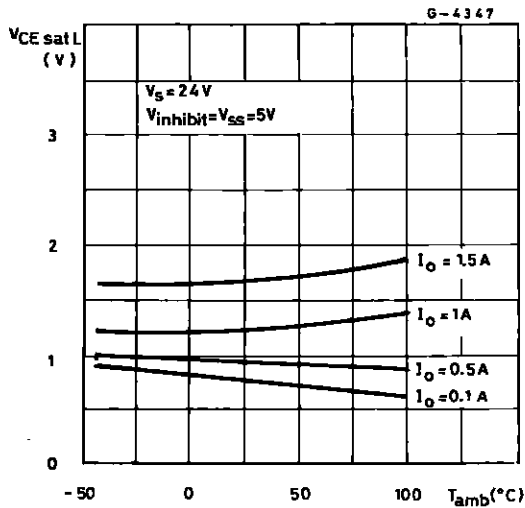


Figure 3 : Source Saturation Voltage versus Ambient Temperature

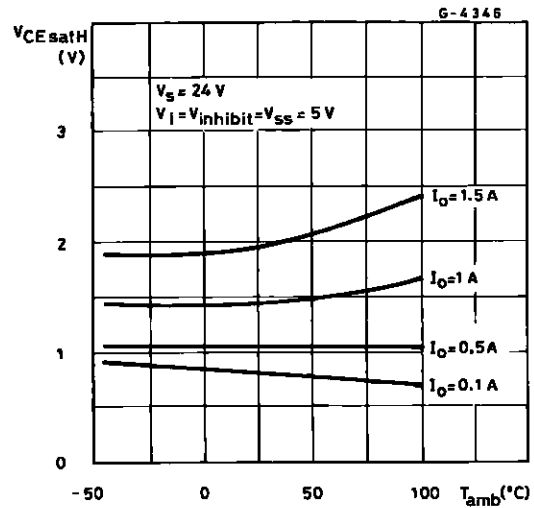


Figure 5 : Quiescent Logic Supply Current versus Logic Supply Voltage

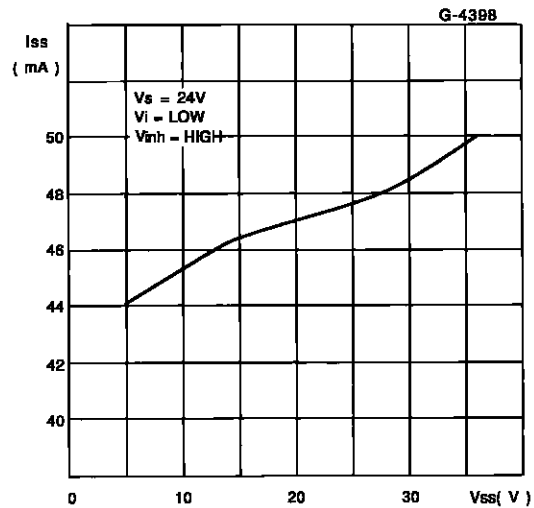


Figure 6 : Output Voltage versus Input Voltage

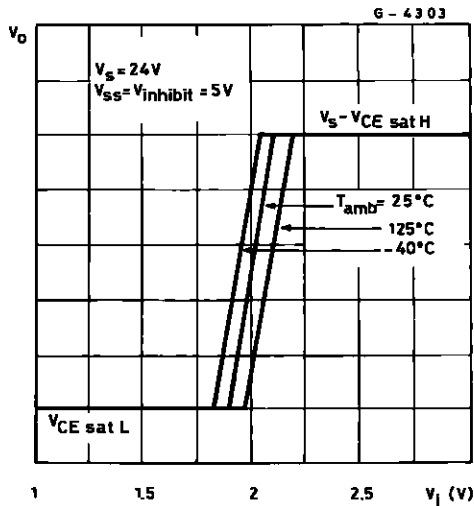
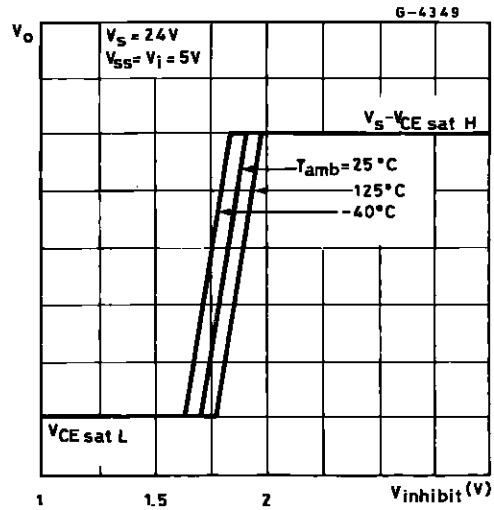
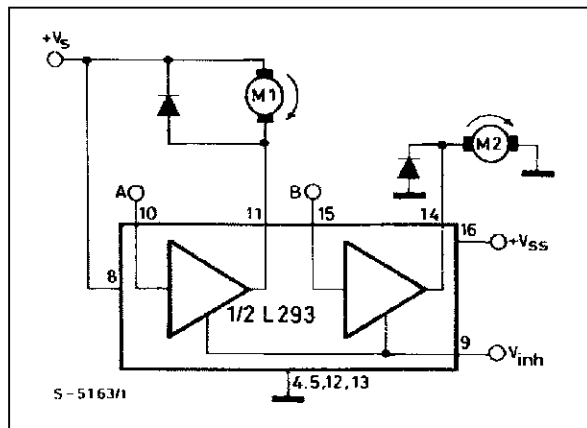


Figure 7 : Output Voltage versus Inhibit Voltage



APPLICATION INFORMATION

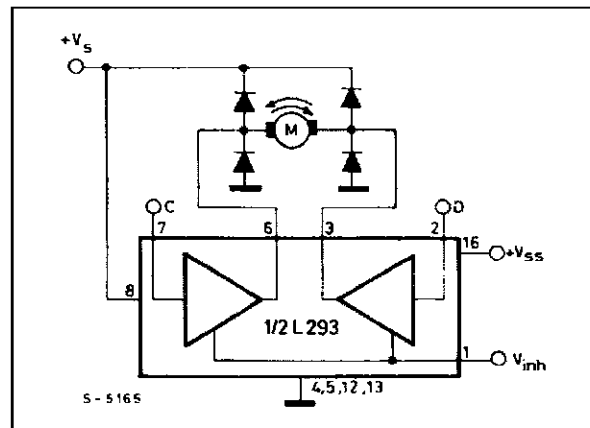
Figure 8 : DC Motor Controls (with connection to ground and to the supply voltage)



V _{inh}	A	M1	B	M2
H	H	Fast Motor Stop	H	Run
H	L	Run	L	Fast Motor Stop
L	X	Free Running Motor Stop	X	Free Running Motor Stop

L = Low H = High X = Don't Care

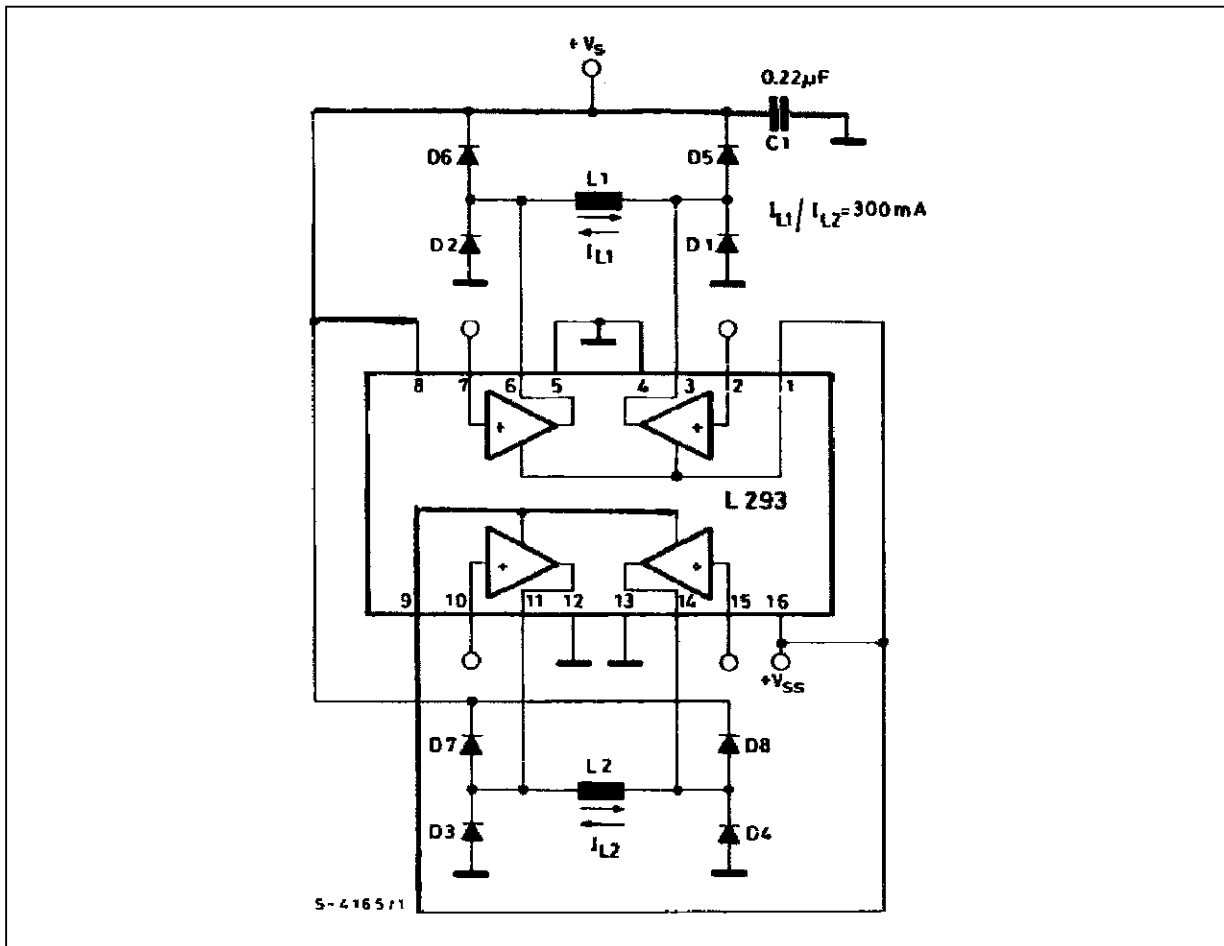
Figure 9 : Bidirectional DC Motor Control



Inputs	Function	
V _{inh} = H	C = H ; D = L	Turn Right
	C = L ; D = H	Turn Left
	C = D	Fast Motor Stop
V _{inh} = L	C = X ; D = X	Free Running Motor Stop

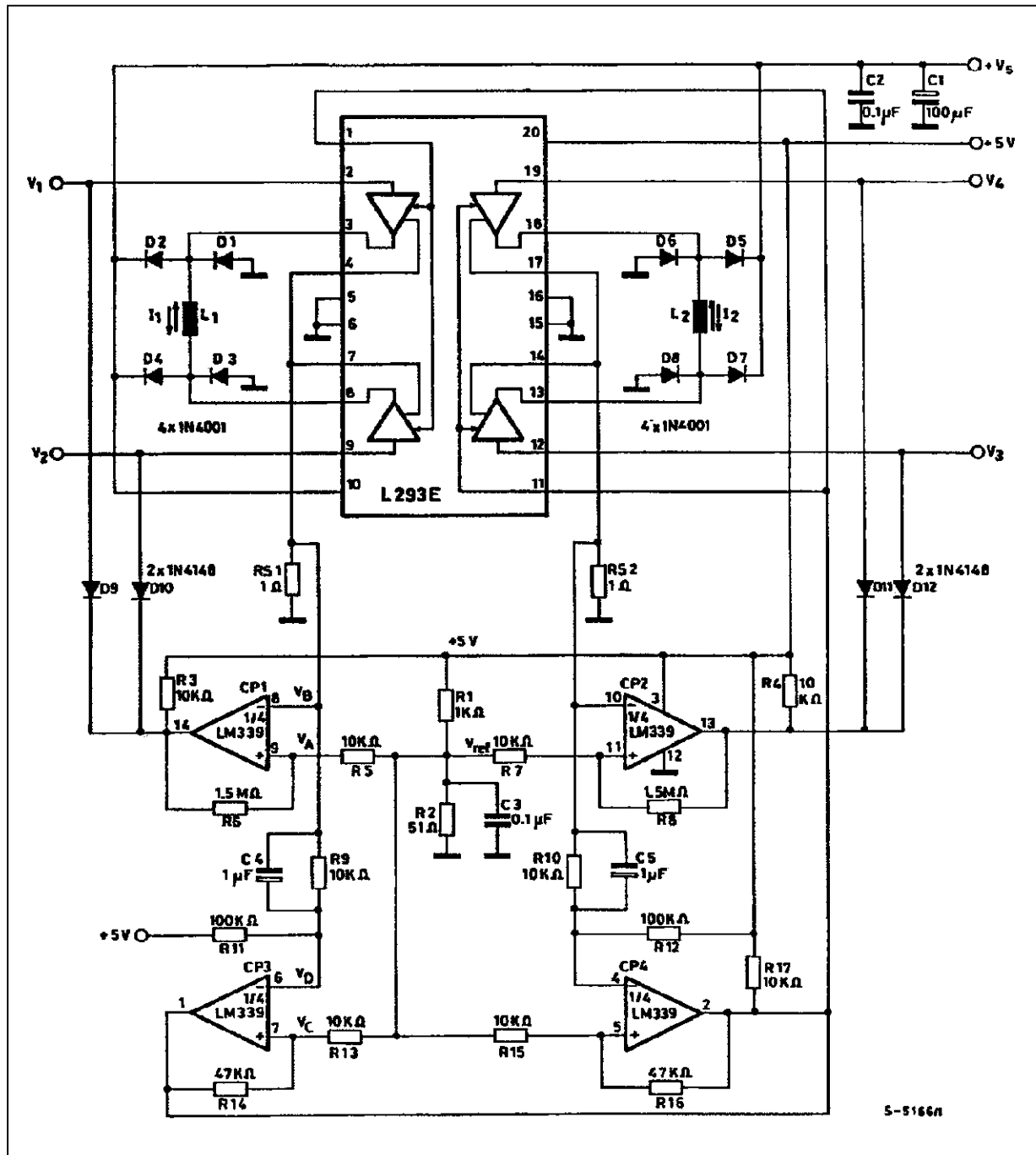
L = Low H = High X = Don't Care

Figure 10 : Bipolar Stepping Motor Control



L293B - L293E

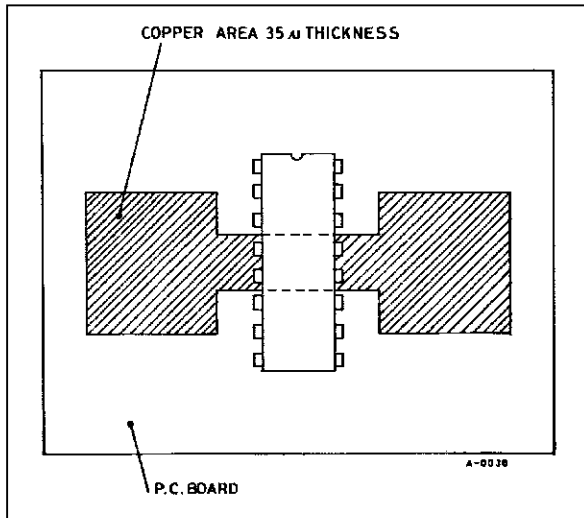
Figure 11 :Stepping Motor Driver with Phase Current Control and Short Circuit Protection



MOUNTING INSTRUCTIONS

The $R_{th\ j-amb}$ of the L293B and the L293E can be reduced by soldering the GND pins to a suitable copper area of the printed circuit board as shown in figure 12 or to an external heatsink (figure 13).

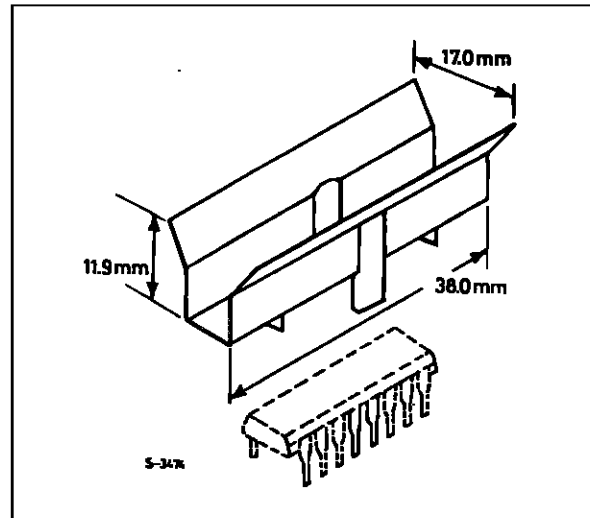
Figure 12 :Example of P.C. Board Copper Area which is Used as Heatsink



During soldering the pins temperature must not exceed 260°C and the soldering time must not be longer than 12 seconds.

The external heatsink or printed circuit copper area must be connected to electrical ground.

Figure 13 :External Heatsink Mounting Example ($R_{th} = 30^{\circ}\text{C/W}$)

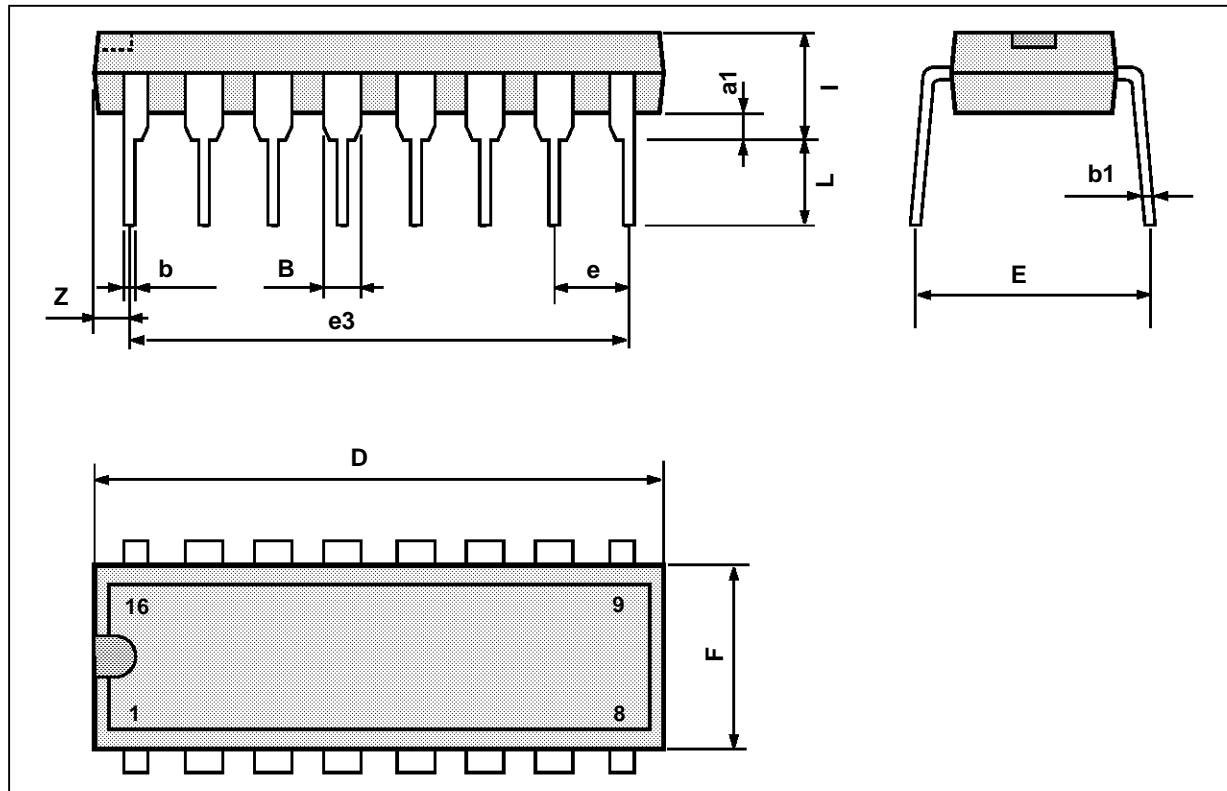


L293B - L293E

DIP16 PACKAGE MECHANICAL DATA

Dimensions	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
a1	0.51			0.020		
B	0.77		1.65	0.030		0.065
b		0.5			0.020	
b1		0.25			0.010	
D			20			0.787
E		8.5			0.335	
e		2.54			0.100	
e3		17.78			0.700	
F			7.1			0.280
i			5.1			0.201
L		3.3			0.130	
Z			1.27			0.050

DIP16PW:TBL

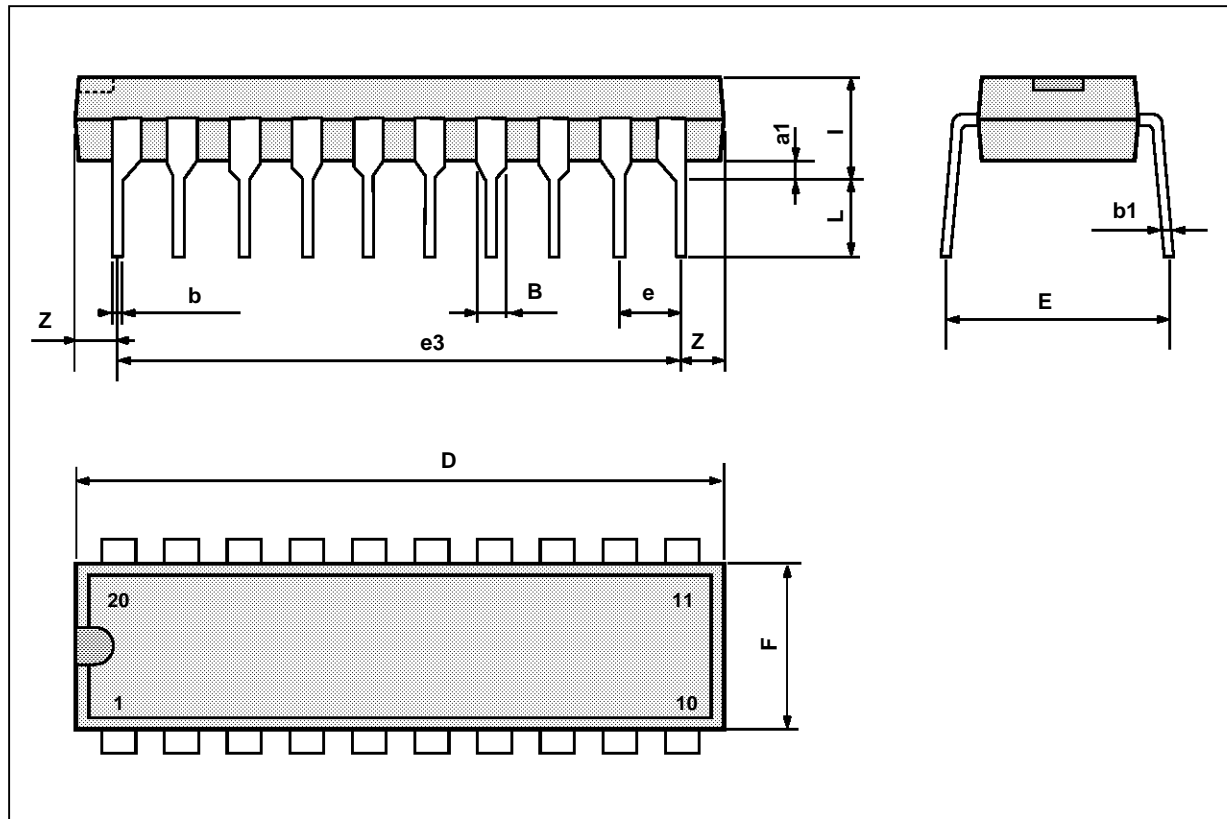


PMDIP16W/EPS

POWERDIP (16+2+2) PACKAGE MECHANICAL DATA

Dimensions	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
a1	0.51			0.020		
B	0.85		1.4	0.033		0.055
b		0.5			0.020	
b1	0.38		0.5	0.015		0.020
D			24.8			0.976
E		8.8			0.346	
e		2.54			0.100	
e3		22.86			0.900	
F			7.1			0.280
i			5.1			0.201
L		3.3			0.130	
Z			1.27			0.050

DIP20PW.TBL



PMDIP20WEPS

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of SGS-THOMSON Microelectronics.

© 1994 SGS-THOMSON Microelectronics - All Rights Reserved

SGS-THOMSON Microelectronics GROUP OF COMPANIES

Australia - Brazil - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.

This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.

Hoja de especificaciones motor ST28

ST28**USAGE**

Office Machines

Automatic Control Machines

Split Air Conditioner

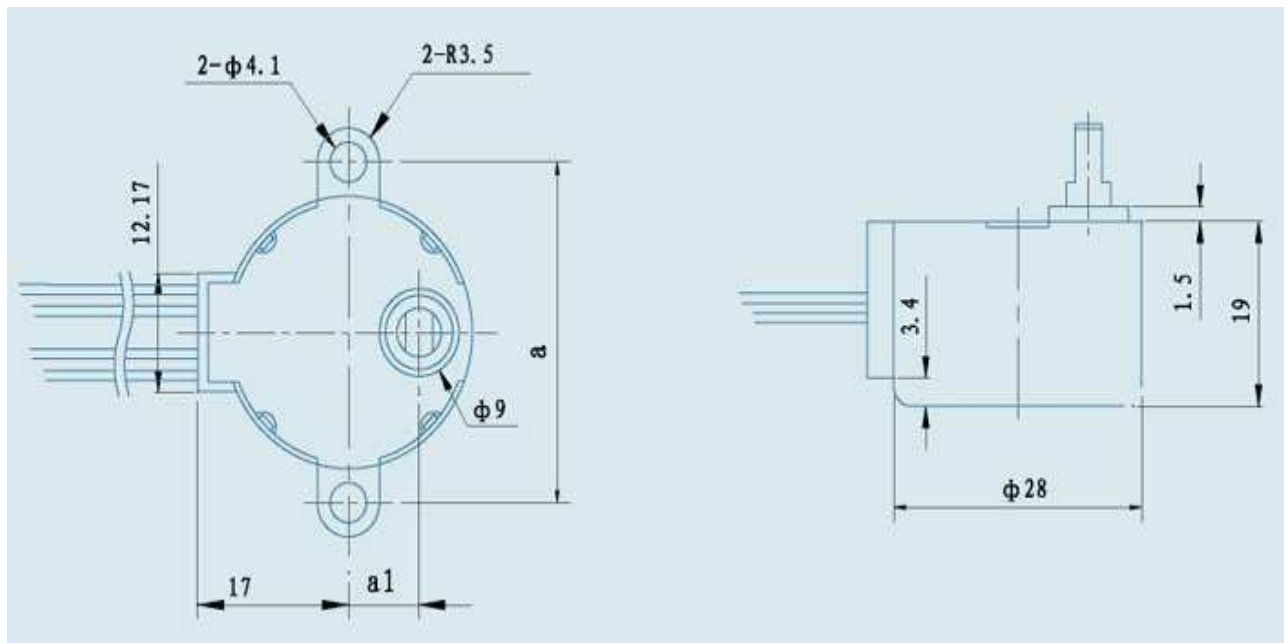
STEPPING MOTOR ST28

USAGE
Office Machines
Automatic Control Machines
Split Air Conditioner

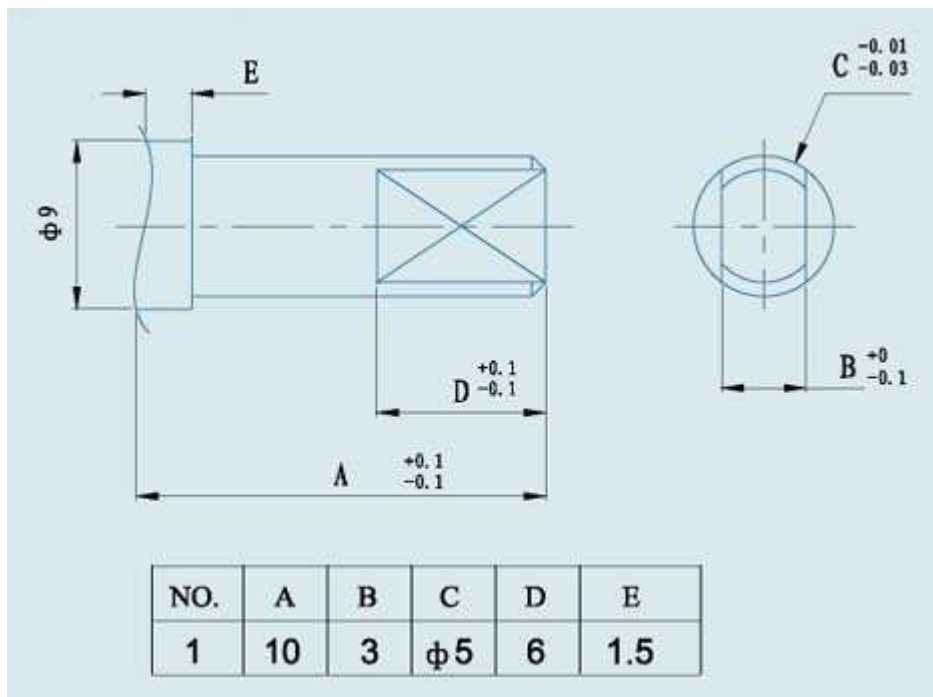
ITEM	Specifications						Unit
Rated Voltage	12						VDC
Current	75	60	48	40	31.5	21.8	mA
Impedance	160	200	250	300	380	550	Ohm
Pull-in Torque (at 100pps)	550	550	450	350	300	240	gf - cm
1 step 1-2 Step Angle	5.625 ϕ X /64						ϕ X
Reduction Ratio	1/64						
Phase	4						
Detent Torque min	350						gf - cm
Max Starting Pulse Rate	550						pps
Max Starting Pulse Rate	900						pps
Insulation Strength	AC 600V-1sec Cut-Off Current :10mA						
Temperature Rise	50						deg
Noise	40						dB
Diameter of Model	Ø28.0						mm

Distance of Model (a)	35.0	39.0	mm
Distance of Model (a1)	8.0	8.0	mm
Height of Model	19.0		mm

1:Dimensions



2:Shaft Type



3:Specification of terminal :

NO.	規格 Specification	出線長度 Length of output wire	端子規格 Terminal specification
1	AWG 1095#28	240 mm	JST SXH-001T-P0.6
2	AWG 1095#28	420 mm	JST SPH-002T-P0.5S
3	AWG 1095#28	420 mm	JST SXH-001T-P0.6
4	AWG 1095#28	1100 mm	JST SXH-001T-P0.6

Código del algoritmo de control

Módulo de control de los motores paso a paso

```
Option Base 1

Public RangMotor(3, 9) As Long
Public RangPuerto(8, 3) As Long
Public DistY As Long, DistX As Long

Dim PasoStepper As Long
Dim AnguloServo As Long
Dim PuertoSalida As String
Dim Zero As Long
Dim sampsPerChanWritten As Long
Dim Puerto0 As String, Puerto1 As String
Dim Tarea0 As Long
Dim Delay As Double
Dim Status As Double

'Subrutina que abre el puerto 0 y lo inicializa desde el pin 0 al pin 7
'Corresponde con la salida de datos para controlar los 2 motores: pin 0-1-2-3 Motor 1 y pin 4-5-6-7 Motor 2
Private Sub Configurar_Port0()
    Status = DAQmxCreateTask("", Tarea0)
    Status = DAQmxCreateDOChan(Tarea0, "dev1/port0/line0:7", "",
DAQmx_Val_ChanForAllLines)
    Status = DAQmxStartTask(Tarea0)
End Sub

'Subrutina para parar Tarea en TI
Sub Stop_Task()
Dim i As Long, j As Long
For i = 1 To 8: For j = 1 To 2
RangPuerto(i, j) = 0
Next: Next
PasarDatosAPuerto0
Status = DAQmxStopTask(Tarea0)
Status = DAQmxClearTask(Tarea0)
End Sub

'Subrutina que inicializa el puerto del DAQ y pone todos los datos a 0
Sub ReiniciarDAQ()
    Status = DAQmxCreateTask("", Tarea0)
    Status = DAQmxCreateDOChan(Tarea0, "dev1/port0/line0:7", "",
DAQmx_Val_ChanForAllLines)
    Status = DAQmxStartTask(Tarea0)
End Sub

'Ahora vuelve a inicializar y a crear las tareas 0 y 1
Sub Inicio()
Dim i As Long, j As Long
Configurar_Port0
StartCap = False
PasoStepper = 1
'With Worksheets("Motores")
'Set RangMotor = Range(.Cells(5, 8), .Cells(8, 13))
'Set RangPuerto = Range(.Cells(6, 2), .Cells(13, 4))
For i = 1 To 8: For j = 1 To 3
RangPuerto(i, j) = 0
Next: Next
UserForm1.Show
'End With
```

```

End Sub

Private Sub PasarDatosPuerto0()
Dim i As Long, Vector(0 To 7) As Byte, Zero As Long
'With Worksheets("Motores")
For i = 0 To 7
Vector(i) = RangPuerto(i + 1, 1)
Next i
Zero = 0
Status = DAQmxWriteDigitalLines(Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)
'End With
End Sub

Sub GirarStepper(Motor As Long, Direccion As Byte, Vueltas As Long)
Dim i As Long, j As Long, Vector(0 To 7) As Byte
Dim Salida As Long, Paso As Long
Dim PinA As Byte, PinB As Byte, PinC As Byte, PinD As Byte 'corresponde
Pin(A,B,C,D) a (0,1,2,3) y(4,5,6,7)
Dim TiempoInicial As Long, TiempoTotal As Long, Espera As Long

'With Worksheets("Motores")
For i = 0 To 7
Vector(i) = 0
Next i
If Motor = 1 Then
PinA = 0
PinB = 1
PinC = 2
PinD = 3
Espera = 200
End If

If Motor = 2 Then
PinA = 4
PinB = 5
PinC = 6
PinD = 7
Espera = 200
End If

Vector(PinA) = 0
Vector(PinB) = 0
Vector(PinC) = 0
Vector(PinD) = 0

Status = DAQmxWriteDigitalLines(Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)
Retraso (Espera)

'TiempoInicial = GetTickCount64()

If Direccion = 0 Then
For j = 1 To Vueltas

Vector(PinA) = 1
Vector(PinB) = 1
Vector(PinC) = 0
Vector(PinD) = 0
Status = DAQmxWriteDigitalLines(Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)

```

```

Retraso (Espera)

Vector (PinA) = 0
Vector (PinB) = 1
Vector (PinC) = 1
Vector (PinD) = 0
Status = DAQmxWriteDigitalLines (Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)
Retraso (Espera)
Vector (PinA) = 0
Vector (PinB) = 0
Vector (PinC) = 1
Vector (PinD) = 1
Status = DAQmxWriteDigitalLines (Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)
Retraso (Espera)

Vector (PinA) = 1
Vector (PinB) = 0
Vector (PinC) = 0
Vector (PinD) = 1
Status = DAQmxWriteDigitalLines (Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)
Retraso (Espera)
Next j
End If

If Direccion = 1 Then
For j = 1 To Vueltas

Vector (PinA) = 1
Vector (PinB) = 0
Vector (PinC) = 0
Vector (PinD) = 1
Status = DAQmxWriteDigitalLines (Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)
Retraso (Espera)

Vector (PinA) = 0
Vector (PinB) = 0
Vector (PinC) = 1
Vector (PinD) = 1
Status = DAQmxWriteDigitalLines (Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)
Retraso (Espera)

Vector (PinA) = 0
Vector (PinB) = 1
Vector (PinC) = 1
Vector (PinD) = 0
Status = DAQmxWriteDigitalLines (Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)
Retraso (Espera)

Vector (PinA) = 1
Vector (PinB) = 1
Vector (PinC) = 0
Vector (PinD) = 0
Status = DAQmxWriteDigitalLines (Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)
Retraso (Espera)

```

```

Next j
End If

Vector(PinA) = 0
Vector(PinB) = 0
Vector(PinC) = 0
Vector(PinD) = 0
Status = DAQmxWriteDigitalLines(Tarea0, 1, True, 10#,
DAQmx_Val_GroupByChannel, Vector(0), sampsPerChanWritten, ByVal 0&)

Snapshot

'TiempoTotal = GetTickCount64() - TiempoInicial
' Label4.Text = TiempoTotal
' Label5.Text = Vueltas
'End With

End Sub

Sub Retraso(i As Long)
Dim j As Long, k As Long
For j = 0 To i
For k = 0 To 1000
Next k
Next j
End Sub

```

Módulo de visión artificial

Módulo de la cámara web

```

Option Explicit

''' User-Defined Types for API Calls

'Declare a UDT to store a GUID for the IPicture OLE Interface
Private Type GUID
Data1 As Long
Data2 As Integer
Data3 As Integer
Data4(0 To 7) As Byte
End Type

'Declare a UDT to store the bitmap information
Private Type uPicDesc
Size As Long
Type As Long
hPic As Long
hPal As Long
End Type

Private Type BitMapFile 'bitmap file header structure
bfType As Integer
bfSize As Long
bfReserved1 As Integer
bfReserved2 As Integer
bfOffBits As Long
End Type

```

```

Private Type BitMapInfo 'bitmap info header structure
    biSize As Long
    biWidth As Long
    biHeight As Long
    biPlanes As Integer
    biBitCount As Integer
    biCompression As Long
    biSizeImage As Long
    biXPelsPerMeter As Long
    biYPelsPerMeter As Long
    biClrUsed As Long
    biClrImportant As Long
End Type
Private Type RGBQuads 'RGBQuad structure (temporary)
    cBlue As Byte
    cGreen As Byte
    cRed As Byte
    cReserved As Byte
End Type
Private Type RGBQuad 'RGBQuad (used in index)
    Blue As Byte
    Green As Byte
    Red As Byte
End Type

Dim BMPData() As Long 'Holds the pixel data
Dim MatrizBMP() As Long 'Matriz mejor estructurada
Public StartCap As Boolean
Public ExitTime As Long

'Does the clipboard contain a bitmap/metafile?
Private Declare PtrSafe Function IsClipboardFormatAvailable Lib "User32"
    (ByVal wFormat As Integer) As Long

'Open the clipboard to read
Private Declare PtrSafe Function OpenClipboard Lib "User32" (ByVal hwnd As
Long) As Long

'Get a pointer to the bitmap/metafile
Private Declare PtrSafe Function GetClipboardData Lib "User32" (ByVal
wFormat As Integer) As Long

'Close the clipboard
Private Declare PtrSafe Function CloseClipboard Lib "User32" () As Long

'Convert the handle into an OLE IPicture interface.
Private Declare PtrSafe Function OleCreatePictureIndirect Lib "olepro32.dll"
    (PicDesc As uPicDesc, RefIID As GUID, ByVal fPictureOwnsHandle As Long, IPic
As IPicture) As Long

'Create our own copy of the bitmap, so it doesn't get wiped out by
subsequent clipboard updates.
Declare PtrSafe Function CopyImage Lib "User32" (ByVal handle As Long, ByVal
un1 As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal un2 As Long) As Long

Const WM_CAP As Long = &H400
Const WM_CAP_DRIVER_CONNECT As Long = WM_CAP + 10
Const WM_CAP_DRIVER_DISCONNECT As Long = WM_CAP + 11
Const WM_CAP_EDIT_COPY As Long = WM_CAP + 30
Const WS_CHILD As Long = &H40000000
Const WS_VISIBLE As Long = &H10000000

```

```

Declare PtrSafe Function SendMessage Lib "User32" Alias "SendMessageA"
(ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As
Any) As Long

Declare PtrSafe Function DestroyWindow Lib "User32" (ByVal hwnd As Long) As
Boolean

Declare PtrSafe Function capCreateCaptureWindowA Lib "avicap32.dll" _
(ByVal lpszWindowName As String, ByVal dwStyle As Long, ByVal x As Long,
ByVal y As Long, ByVal nWidth As Long, _
ByVal nHeight As Integer, ByVal hWndParent As Long, ByVal nID As Long) As
Long

Declare PtrSafe Function capGetDriverDescriptionA Lib "avicap32.dll" _
(ByVal wDriver As Integer, ByVal lpszName As String, _
ByVal cbName As Long, ByVal lpszVer As String, ByVal cbVer As Long) As
Boolean

Declare PtrSafe Function GetDesktopWindow Lib "User32" () As Long

Public hwnd As Long
Public iDevice As Long
Dim Parar As Boolean
Dim k As Single, A As Single, B As Single, C As Single, D As Single, XG As
Single, YG As Single, contador1 As Single, contador2 As Single

Sub Snapshot()

    DimRetVal
    Dim myWb As Excel.Workbook
    Dim Ready As Boolean

    Ready = False

    ' Capturar la nueva imagen
   RetVal = Shell("J:\TFG\Códigos\Definitivos\Bucle_cerrado\CommandCam.exe
/filename J:\TFG\Códigos\Definitivos\Bucle_cerrado\image.bmp",
vbNormalFocus)

    ' Pequeño retraso para permitir que se guarde la nueva imagen
Application.Wait (Now + TimeValue("00:00:07"))

    ' Comprobamos que la imagen se guarda correctamente antes de continuar
el proceso

    If Not Dir("J:\TFG\Códigos\Definitivos\Bucle_cerrado\image.bmp",
vbDirectory) = vbNullString Then
        MsgBox ("Imagen guardada")
        Ready = True
    Else
        MsgBox ("Imagen NO guardada")
        Ready = False
    End If

    ' Pasamos la imagen para transformarla en valores RGB

    If Ready = True Then
        Call
GetBmpValues("J:\TFG\Códigos\Definitivos\Bucle_cerrado\image.bmp")

```

```

End If

End Sub

Sub GetBmpValues (FileName As String)
' Muestra información sobre el mapa de bits
' Manualmente se dibuja en otro PictureBox
    Dim bmf As BitMapFile
    Dim bmi As BitMapInfo
    Dim bmpPixel As RGBQuad
    Dim lngCol As Long
    Dim lngRow As Long
    Dim nRowBytes As Long
    Dim i As Long, j As Long, k As Long
    Dim u As Long

' Muestra las propiedades del mapa de bits

    Open FileName For Binary Access Read As #1 Len = 1
    Get #1, , bmf          'load the file header
    Get #1, , bmi          'A continuación la información de cabecera
    lngRow = bmi.biHeight
    lngCol = bmi.biWidth

' Tamaño de fila en bytes
    nRowBytes = lngCol * 3

' Hacer el n° de bytes de fila múltiplo de 4
    If nRowBytes Mod 4 <> 0 Then
        nRowBytes = nRowBytes + (4 - nRowBytes Mod 4)
    End If

' Vaciar celdas de datos

    For k = 3 To 5
        Worksheets(k).Cells.ClearContents
    Next k

' Dar tamaño a celdas en hoja 5
'   With Worksheets(5)
'       .Rows("1:" & bmi.biHeight).RowHeight = 2
'       .Columns("1:" & bmi.biWidth).ColumnWidth = 1.7
'   End With
    Application.ScreenUpdating = False

' Leer cada píxel y pasarlo a hoja de cálculo

    For i = 0 To lngRow - 1: For j = 0 To lngCol - 1
        Get #1, bmf.bfOffBits + 1 + i * nRowBytes + j * 3, bmpPixel
        Worksheets(3).Cells(lngRow - i, j + 1).Value = bmpPixel.Blue
        Worksheets(4).Cells(lngRow - i, j + 1).Value = bmpPixel.Green
        Worksheets(5).Cells(lngRow - i, j + 1).Value = bmpPixel.Red
'         Worksheets(5).Cells(lngRow - i, j + 1).Interior.Color =
        RGB(bmpPixel.Red, bmpPixel.Green, bmpPixel.Blue)
    Next: Next
    Application.ScreenUpdating = True

    With Worksheets("hoja1")
        .Cells(1, 2).Value = bmi.biSize
        .Cells(2, 2).Value = bmi.biBitCount
        .Cells(3, 2).Value = bmi.biCompression
    End With
End Sub

```

```

.Cells(4, 2).Value = bmi.biWidth
.Cells(5, 2).Value = bmi.biHeight
.Cells(6, 2).Value = bmi.biClrImportant
.Cells(7, 2).Value = bmi.biClrUsed
.Cells(8, 2).Value = bmi.biPlanes
.Cells(9, 2).Value = bmi.biSizeImage
.Cells(10, 2).Value = bmi.biXPelsPerMeter
.Cells(11, 2).Value = bmi.biYPelsPerMeter
.Cells(1, 3).Value = bmf.bfType
.Cells(2, 3).Value = bmf.bfSize
.Cells(3, 3).Value = bmf.bfReserved1
.Cells(4, 3).Value = bmf.bfReserved2
.Cells(5, 3).Value = bmf.bfOffBits
End With

Close #1

Iniciar

End Sub

```

Módulo de cálculos

```

Sub Iniciar()

k = 5

'Para Sub Resultado() y para inicializar el bucle
contador1 = 2
contador2 = 1
'Inicializamos A y B
A = 45
B = 70

Call EncontrarPixel(k, A, B)
End Sub

Sub EncontrarPixel(k, x As Single, y As Single)
'Función que busca el siguiente píxel

Dim i As Single, j As Single
Dim Salir As Boolean

Salir = False

With Worksheets(k)

    For i = A To A + 35
'Iniciamos un bucle para recorrer la imagen buscando el inicio del píxel
        For j = B To B + 40
            If .Cells(i, j).Value > 6 Then
                If .Cells(i + 1, j).Value > 0 And .Cells(i, j + 1).Value > 0
Then
                    A = i
'Guardamos las coordenadas del inicio del píxel
                    B = j
                    Salir = True
'La "patada" para salir del bucle una vez encontrado el inicio del píxel
                    Exit For
                End If
            End If
        End For
    End For
End With

```



```

        End If
    Next j
    If Salir = True Then
'La otra "patada" para salir del segundo for que forma el bucle
        Exit For
    End If
Next i

    Call CentroMasa(k, A, B)
'Llamamos a la función que calculará los centros

End With
End Sub

Sub CentroMasa(k, x As Single, y As Single)

'Función que calcula los centros de masa de los píxeles

Dim i As Single, j As Single, NumeradorX As Single, NumeradorY As Single,
SumNumX As Single, SumNumY As Single, Denominador As Single
Dim Salir As Boolean

Salir = False

With Worksheets(k)

    XG = 0
    YG = 0
    NumeradorX = 0
    NumeradorY = 0
    SumNumX = 0
    SumNumY = 0
    Denominador = 0

'Buclé que calcula los centros en Y a partir de la referencia A y B
    For i = x - 5 To x + 35
        For j = y - 15 To y + 15
            Denominador = Denominador + .Cells(i, j).Value
            NumeradorY = i * .Cells(i, j).Value
            SumNumY = SumNumY + NumeradorY
        Next j
    Next i
'Buclé que calcula los centros en X a partir de la referencia A y B
    For i = x - 5 To x + 35
        For j = y - 15 To y + 15
            NumeradorX = j * .Cells(i, j).Value
            SumNumX = SumNumX + NumeradorX
        Next j
    Next i
'Cálculo de los centros de masa
    XG = SumNumX / Denominador
    YG = SumNumY / Denominador
'Llamamos a la función que imprime los resultados
    Call Resultado2(YG, XG)

'Condiciones para llamar a la función EncontrarPixel y volver a encontrar un
nuevo píxel, la de A es para saber cuando parar de cambiar filas

'Para que estando en la última fila la termine de calcular a través del
siguiente bucle Do While

```

```

    If A + 80 < 480 Then
        Parar = False
        If B + 70 < 640 Then 'Esta condición con B es para cambiar las
columnas
            B = B + 40
            contador2 = contador2 + 1
            Salir = False
            Call EncontrarPixel(k, A - 5, B)
        Else 'Cambiamos fila en este Else
            A = A + 50
            B = 30
            contador2 = 1
            contador1 = contador1 + 1
            Salir = False
            Call EncontrarPixel(k, A, B)
        End If
    Else
        Parar = True
    End If

'Cuando estamos en la última fila, este bucle termina de calcular las
columnas de esa última fila
    If Parar = True Then
While B + 70 < 640
            B = B + 40
            contador2 = contador2 + 1
            Salir = False
            Call EncontrarPixel(k, A - 5, B)
        Loop
    End If

End With
End Sub

Sub Resultado(y As Single, x As Single)
'Función que imprime el resultado y continua el bucle en caso de no
conseguir la precisión deseada

With Worksheets("Hoja2")

    .Cells(contador1, contador2 + 14) = y
    .Cells(contador1 + 14, contador2 + 14) = x
End With

If Worksheets("Hoja2").Cells(24, 25) <> "" And Worksheets("Hoja3").Cells(13,
6) = "" And Iteracion = 1 Then
    Range(Worksheets("Hoja2").Cells(3, 16), Worksheets("Hoja2").Cells(7,
20)).Copy
    Worksheets("Hoja3").Cells(3, 2).PasteSpecial xlValues
    Range(Worksheets("Hoja2").Cells(2, 15), Worksheets("Hoja2").Cells(11,
25)).ClearContents
    Range(Worksheets("Hoja2").Cells(17, 16), Worksheets("Hoja2").Cells(21,
20)).Copy
    Worksheets("Hoja3").Cells(9, 2).PasteSpecial xlValues
    Range(Worksheets("Hoja2").Cells(16, 15), Worksheets("Hoja2").Cells(25,
25)).ClearContents
    Iteracion = Iteracion + 1
    Load UserForm2
    UserForm2.Show

```

```

ElseIf Worksheets("Hoja2").Cells(24, 25) <> "" And
Worksheets("Hoja3").Cells(27, 6) = "" And Iteracion = 2 Then
    Range(Worksheets("Hoja2").Cells(3, 16), Worksheets("Hoja2").Cells(7,
20)).Copy
    Worksheets("Hoja3").Cells(17, 2).PasteSpecial xlValues
    Range(Worksheets("Hoja2").Cells(2, 15), Worksheets("Hoja2").Cells(11,
25)).ClearContents
    Range(Worksheets("Hoja2").Cells(17, 16), Worksheets("Hoja2").Cells(21,
20)).Copy
    Worksheets("Hoja3").Cells(23, 2).PasteSpecial xlValues
    Range(Worksheets("Hoja2").Cells(16, 15), Worksheets("Hoja2").Cells(25,
25)).ClearContents
        If Abs(DistY - Worksheets("Hoja3").Range("Promedio1").Value) > 0.002
Then
            Iteracion = Iteracion + 1
            DistY = (DistY - Worksheets("Hoja3").Range("Promedio1").Value)
            If DistY > 0 Then
                Call GirarStepper(2, 1, DistY * CiclosPormm)
            End If
            If DistY < 0 Then
                Call GirarStepper(2, 0, Abs(DistY) * CiclosPormm)
            End If
        Else
            MsgBox ("Fin del movimiento con éxito")
            Load UserForm1
            UserForm1.Show
        End If

ElseIf Worksheets("Hoja2").Cells(24, 25) <> "" And
Worksheets("Hoja3").Cells(41, 6) = "" And Iteracion = 3 Then
    Range(Worksheets("Hoja2").Cells(3, 16), Worksheets("Hoja2").Cells(7,
20)).Copy
    Worksheets("Hoja3").Cells(31, 2).PasteSpecial xlValues
    Range(Worksheets("Hoja2").Cells(2, 15), Worksheets("Hoja2").Cells(11,
25)).ClearContents
    Range(Worksheets("Hoja2").Cells(17, 16), Worksheets("Hoja2").Cells(21,
20)).Copy
    Worksheets("Hoja3").Cells(37, 2).PasteSpecial xlValues
    Range(Worksheets("Hoja2").Cells(16, 15), Worksheets("Hoja2").Cells(25,
25)).ClearContents
        If Abs(DistY - Worksheets("Hoja3").Range("Promedio2").Value) > 0.002
Then
            Iteracion = Iteracion + 1
            DistY = (DistY - Worksheets("Hoja3").Range("Promedio2").Value)
            If DistY > 0 Then
                Call GirarStepper(2, 1, DistY * CiclosPormm)
            End If
            If DistY < 0 Then
                Call GirarStepper(2, 0, Abs(DistY) * CiclosPormm)
            End If
        Else
            MsgBox ("Fin del movimiento con éxito")
            Load UserForm1
            UserForm1.Show
        End If

ElseIf Worksheets("Hoja2").Cells(24, 25) <> "" And
Worksheets("Hoja3").Cells(55, 6) = "" And Iteracion = 4 Then
    Range(Worksheets("Hoja2").Cells(3, 16), Worksheets("Hoja2").Cells(7,
20)).Copy

```

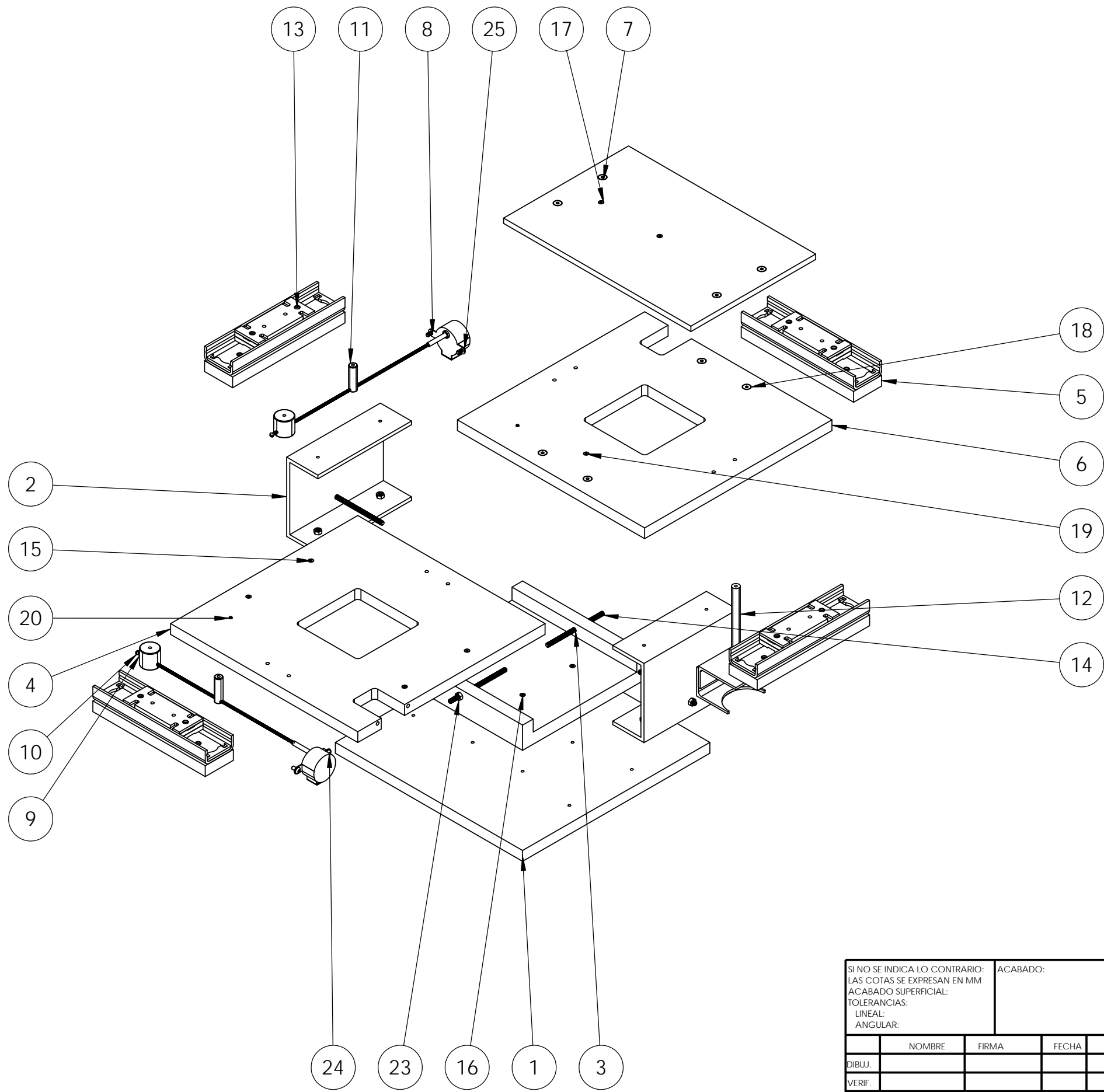
```

Worksheets("Hoja3").Cells(45, 2).PasteSpecial xlValues
Range(Worksheets("Hoja2").Cells(2, 15), Worksheets("Hoja2").Cells(11,
25)).ClearContents
Range(Worksheets("Hoja2").Cells(17, 16), Worksheets("Hoja2").Cells(21,
20)).Copy
Worksheets("Hoja3").Cells(51, 2).PasteSpecial xlValues
Range(Worksheets("Hoja2").Cells(16, 15), Worksheets("Hoja2").Cells(25,
25)).ClearContents
    If Abs(DistY - Worksheets("Hoja3").Range("Promedio3").Value) > 0.005
Then
        Iteracion = Iteracion + 1
        DistY = (DistY - Worksheets("Hoja3").Range("Promedio3").Value)
        If DistY > 0 Then
            Call GirarStepper(2, 1, DistY * CiclosPormm)
        End If
        If DistY < 0 Then
            Call GirarStepper(2, 0, Abs(DistY) * CiclosPormm)
        End If
    Else
        MsgBox ("Fin del movimiento con éxito")
        Load UserForm1
        UserForm1.Show
    End If

ElseIf Worksheets("Hoja2").Cells(24, 25) <> "" And
Worksheets("Hoja3").Cells(69, 6) = "" And Iteracion = 5 Then
    Range(Worksheets("Hoja2").Cells(3, 16), Worksheets("Hoja2").Cells(7,
20)).Copy
    Worksheets("Hoja3").Cells(59, 2).PasteSpecial xlValues
    Range(Worksheets("Hoja2").Cells(2, 15), Worksheets("Hoja2").Cells(11,
25)).ClearContents
    Range(Worksheets("Hoja2").Cells(17, 16), Worksheets("Hoja2").Cells(21,
20)).Copy
    Worksheets("Hoja3").Cells(65, 2).PasteSpecial xlValues
    Range(Worksheets("Hoja2").Cells(16, 15), Worksheets("Hoja2").Cells(25,
25)).ClearContents
        If Abs(DistY - Worksheets("Hoja3").Range("Promedio4").Value) > 0.002
Then
            MsgBox ("No se pudo conseguir la precisión deseada")
        Else
            MsgBox ("Fin del movimiento con éxito")
            Load UserForm1
            UserForm1.Show
        End If
    End If
End Sub

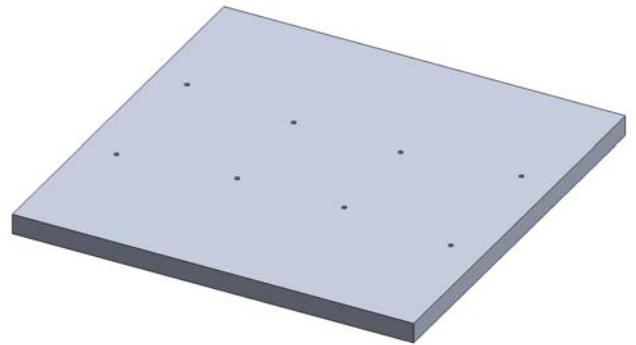
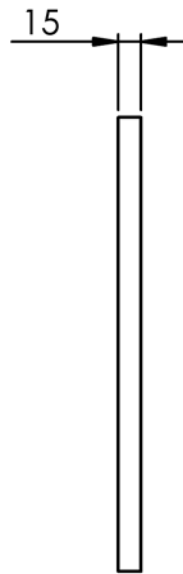
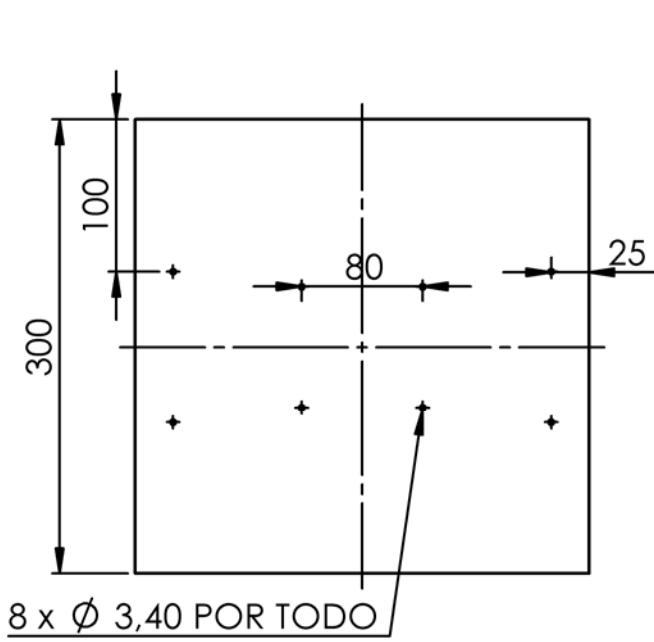
```

Planos



N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Base	1
2	Soporte nivel 1	2
3	Soporte pantalla	1
4	Nivel 1	1
5	Soporte patin	4
6	Nivel 2	1
7	Soporte superior	1
8	Motor paso a paso	2
9	Soporte tornillo guía 1	2
10	Tornillo guía	2
11	Soporte tornillo guía 2	2
12	Soporte cámara	1
13	Patin	4
14	ISO 4026 - M5 x 25-S	4
15	ISO 10642 - M3 x 25 --- 25S	8
16	ISO 10642 - M3 x 30 --- 18S	8
17	ISO 10642 - M3 x 30 --- 30S	5
18	ISO 10642 - M5 x 20 --- 20S	8
19	ISO 10642 - M3 x 10 --- 10S	2
20	ISO 10642 - M3 x 16 --- 16S	2
21	ISO 10642 - M3 x 12 --- 12S	1
22	ISO 10642 - M5 x 8 --- 8S	1
23	Hexagon Nut ISO - 4034 - M5 - N	12
24	ISO 7380 - M4 x 8 --- 8N	2
25	ISO 7380 - M4 x 8 --- 8S	2
26	Hexagon Nut ISO - 4034 - M5 - S	1

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:				ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
						TÍTULO: Lista de materiales	
						N.º DE DIBUJO	
						A3	
						ESCALA:1:5	
						HOJA 1 DE 1	



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA		
DIBUJ.					
VERIF.					
APROB.					
FABR.					
CAUID.				MATERIAL:	
				PESO:	

TÍTULO:

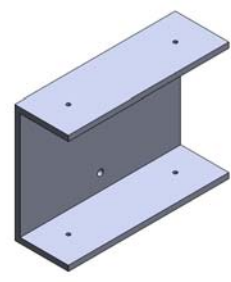
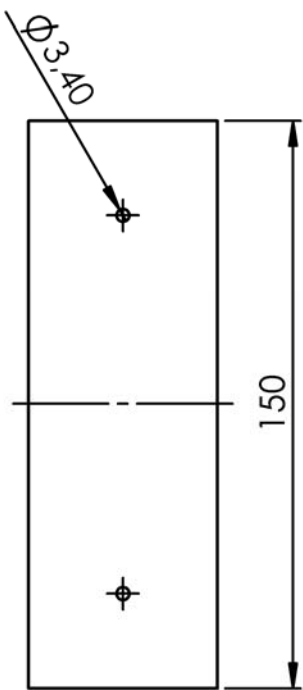
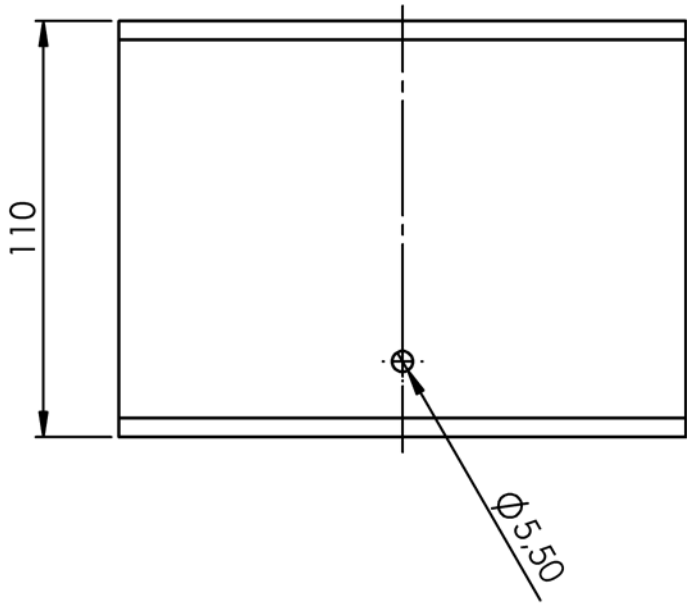
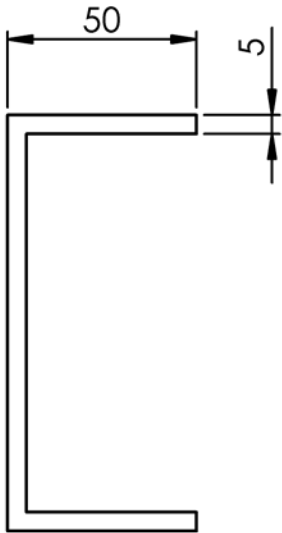
Base

N.º DE DIBUJO

A4

ESCALA:1:5

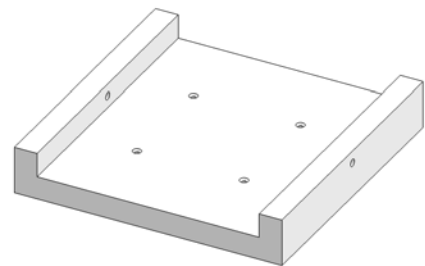
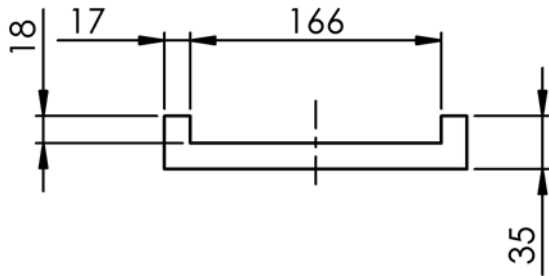
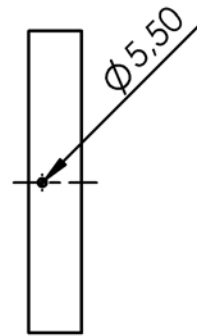
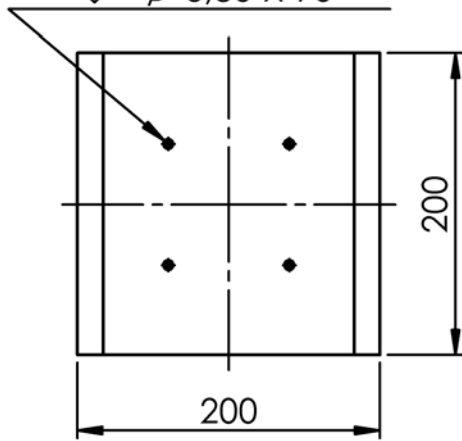
HOJA 1 DE 1



ESCALA 1 : 5

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:			ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN		
NOMBRE			FIRMA		FECHA		TÍTULO: <h1>Soporte nivel 1</h1>				
DIBUJ.			VERIF.		APROB.		MATERIAL:				
FABR.			CAUID.		PESO:		N.º DE DIBUJO				
ESCALA:1:2			HOJA 1 DE 1		A4						

4 x \varnothing 3,40 POR TODO
 \sphericalangle \varnothing 6,30 X 90°



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA	
DIBUJ.				
VERIF.				
APROB.				
FABR.				
CAUID.				
				MATERIAL:
				PESO:

TÍTULO:

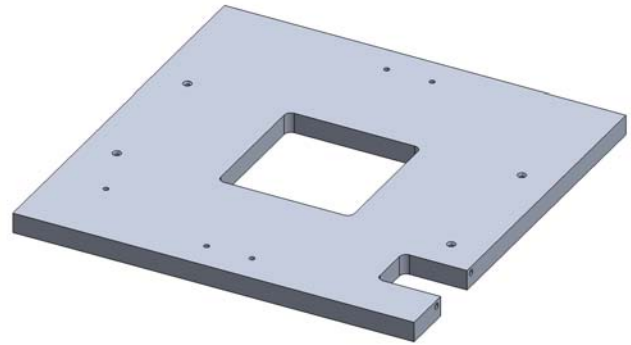
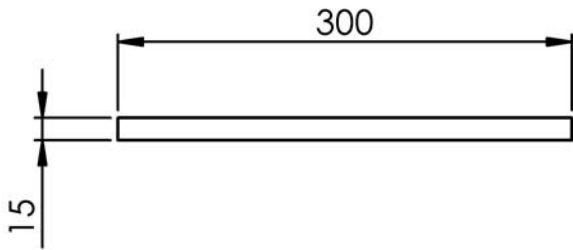
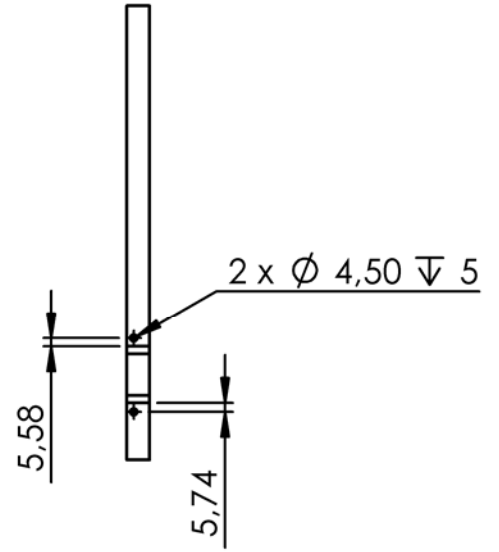
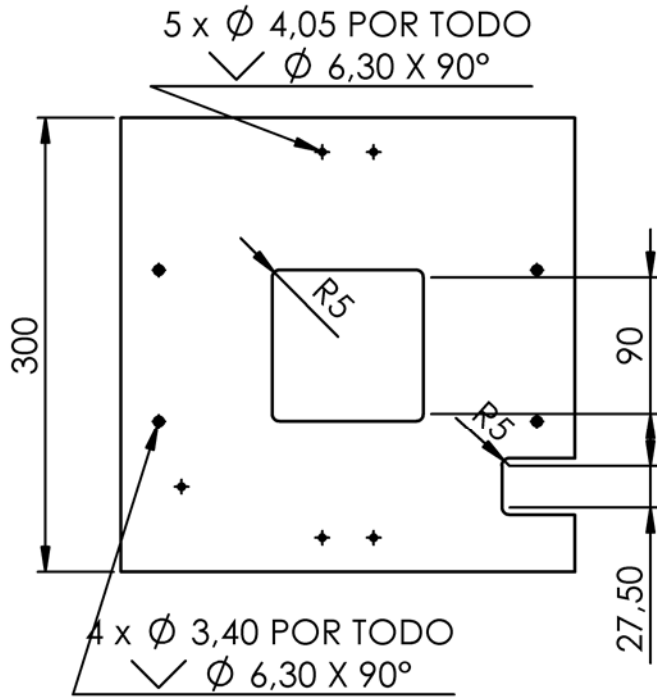
Soporte pantalla

N.º DE DIBUJO

A4

ESCALA:1:5

HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA	
DIBUJ.				
VERIF.				
APROB.				
FABR.				
CAUID.				
				MATERIAL:
				PESO:

TÍTULO:

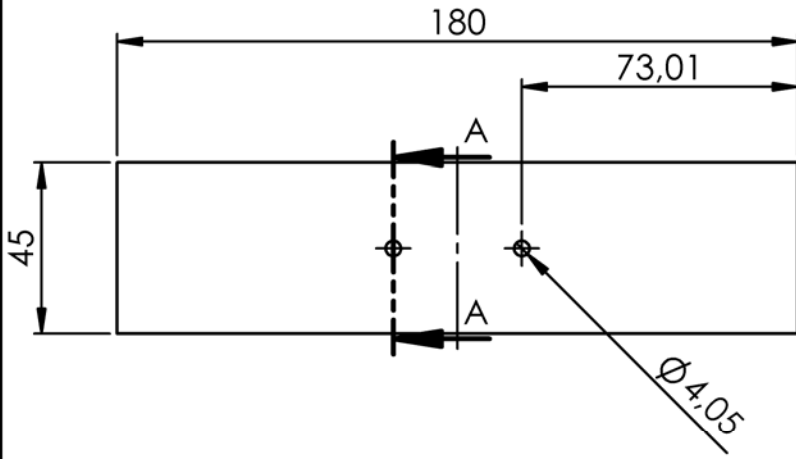
Nivel 1

N.º DE DIBUJO

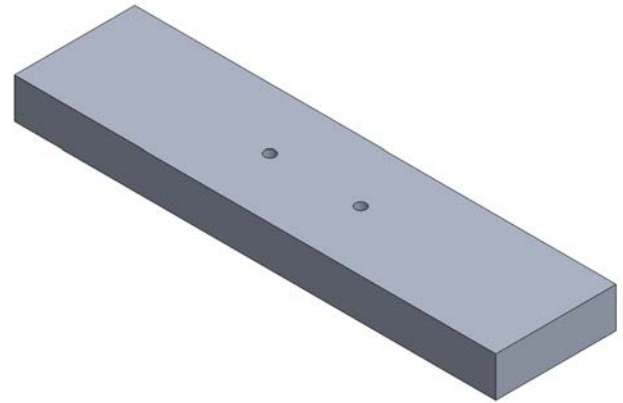
A4

ESCALA:1:5

HOJA 1 DE 1



SECCIÓN A-A



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA		
DIBUJ.					
VERIF.					
APROB.					
FABR.					
CAUID.					
				MATERIAL:	
				PESO:	

TÍTULO:

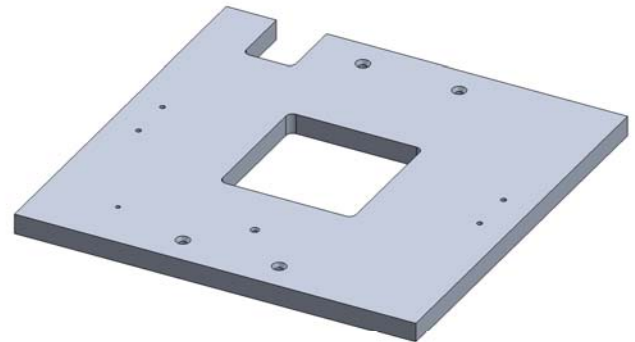
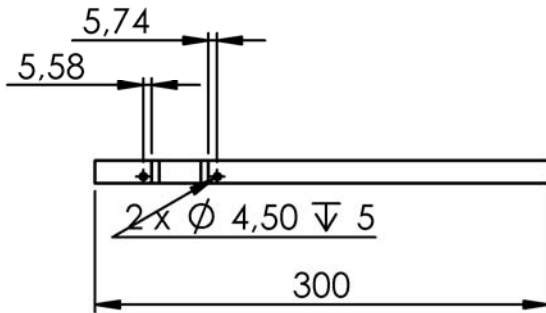
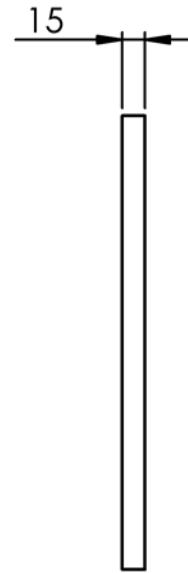
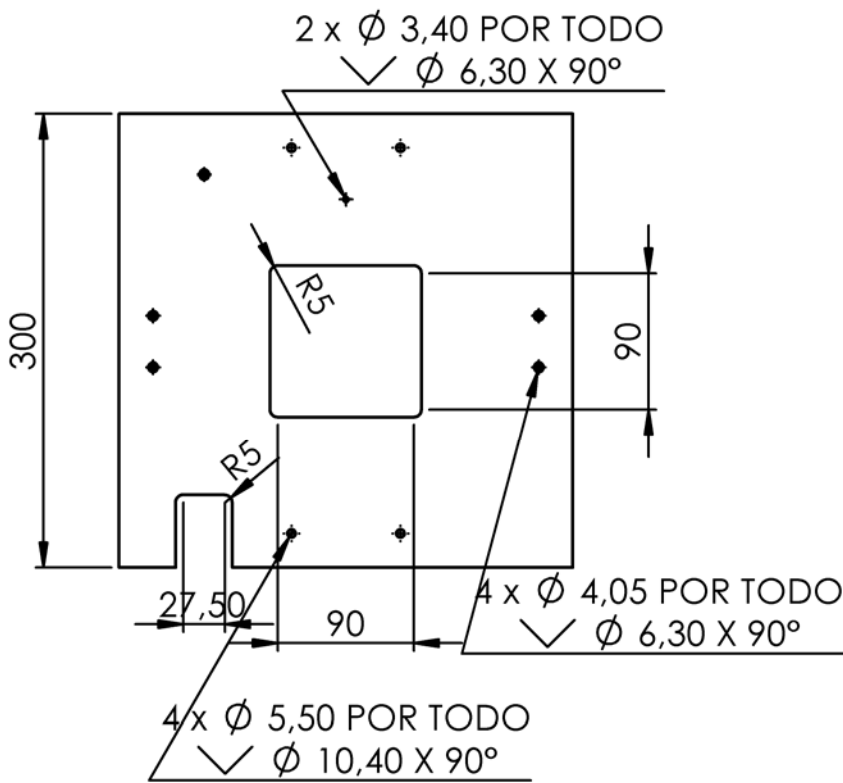
Soporte patín

N.º DE DIBUJO

A4

ESCALA:1:2

HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA	
DIBUJ.				
VERIF.				
APROB.				
FABR.				
CALID.				
				MATERIAL:
				PESO:

TÍTULO:

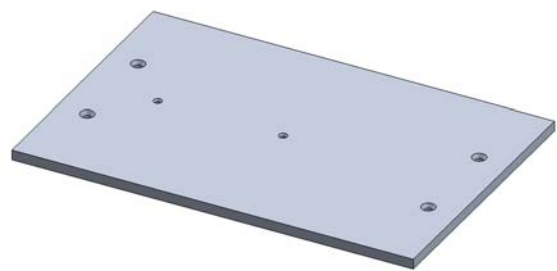
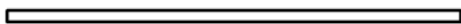
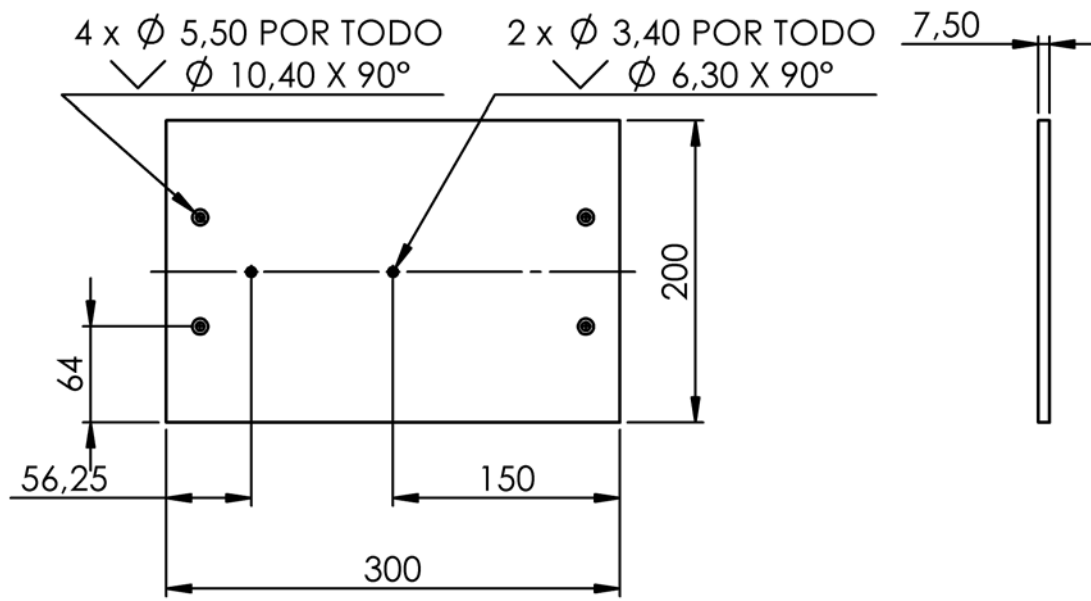
Nivel 2

N.º DE DIBUJO

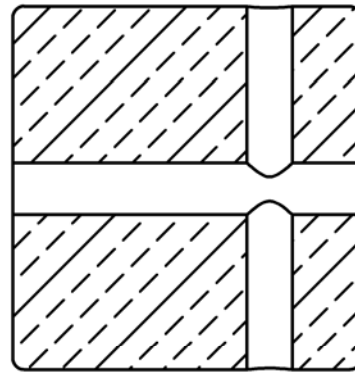
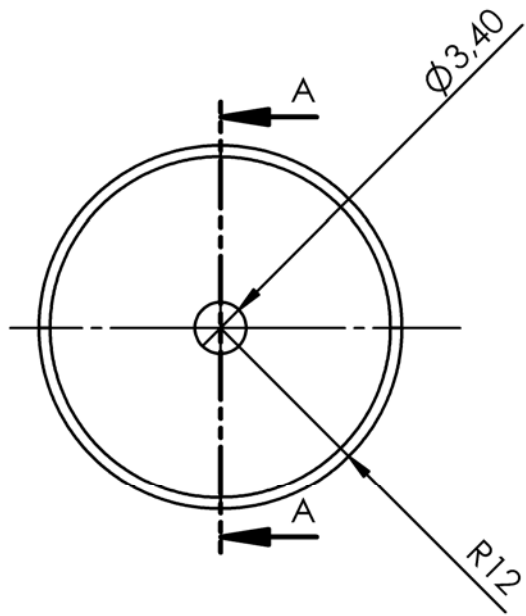
A4

ESCALA:1:5

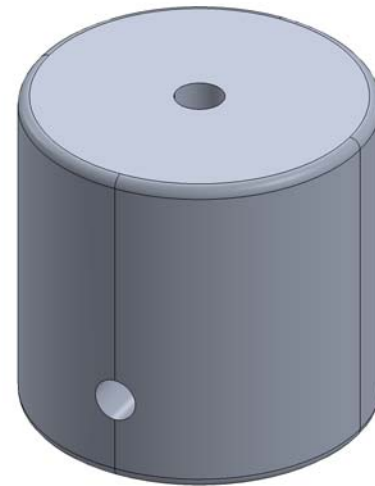
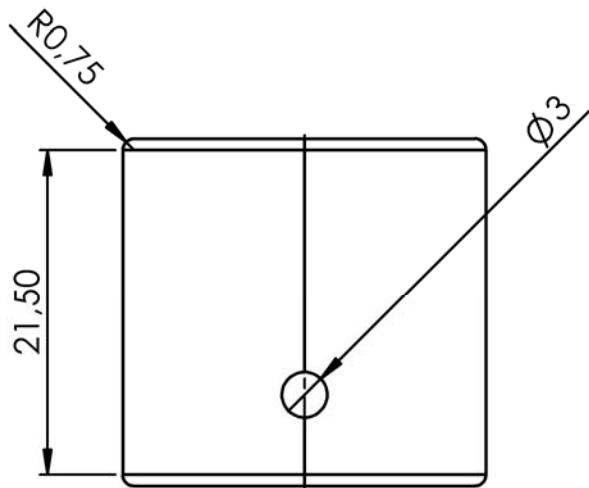
HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO: <h1>Soporte superior</h1>			
DIBUJ.		VERIF.		APROB.		MATERIAL:			
FABR.		CAUID.		PESO:		N.º DE DIBUJO		A4	
ESCALA:1:5		HOJA 1 DE 1							



SECCIÓN A-A



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA		
DIBUJ.					
VERIF.					
APROB.					
FABR.					
CAID.				MATERIAL:	
				PESO:	

TÍTULO:

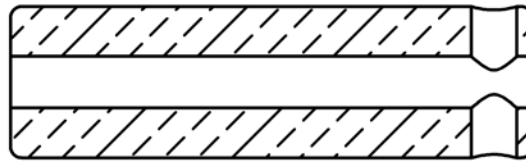
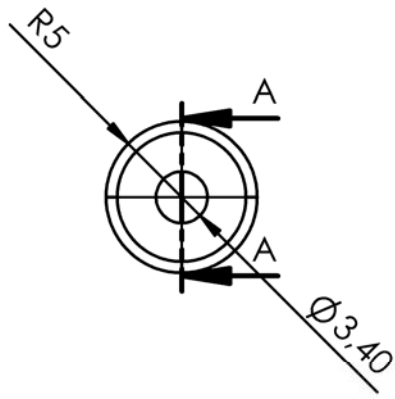
Soporte tornillo guía 1

N.º DE DIBUJO

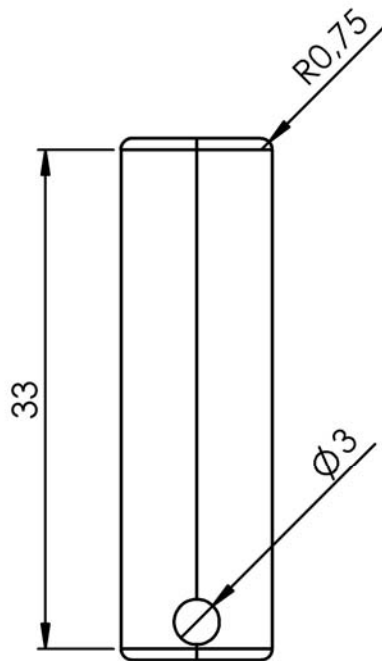
A4

ESCALA:2:1

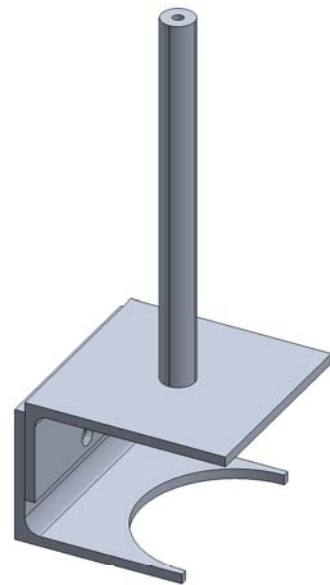
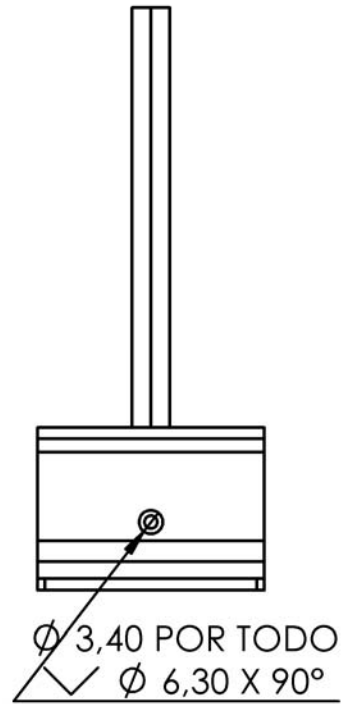
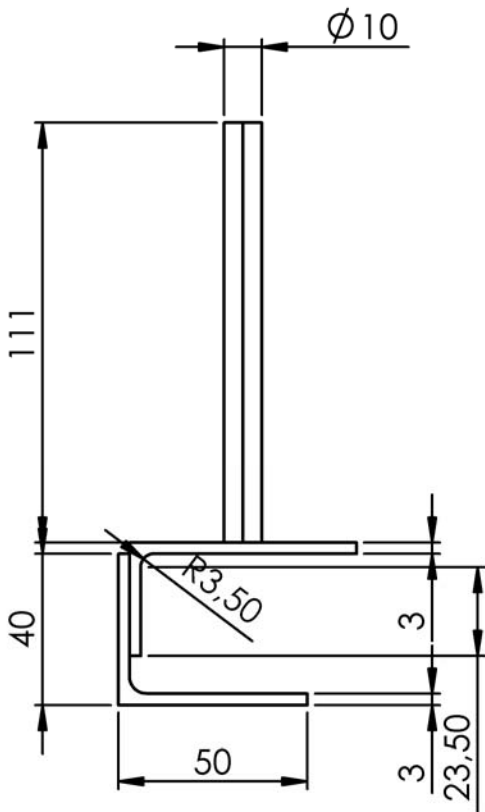
HOJA 1 DE 1



SECCIÓN A-A



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO: <h1>Soporte tornillo guía 2</h1>			
DIBUJ.		VERIF.		APROB.		MATERIAL:			
FABR.		CALID.		PESO:		ESCALA:2:1		HOJA 1 DE 1	
						N.º DE DIBUJO		A4	



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA	
DIBUJ.				
VERIF.				
APROB.				
FABR.				
CALID.				
				MATERIAL:
				PESO:

TÍTULO:

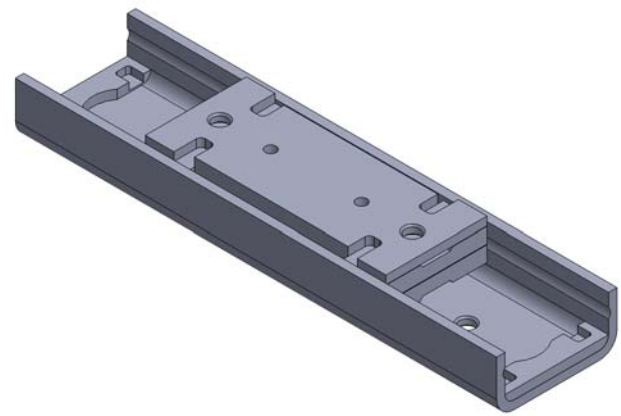
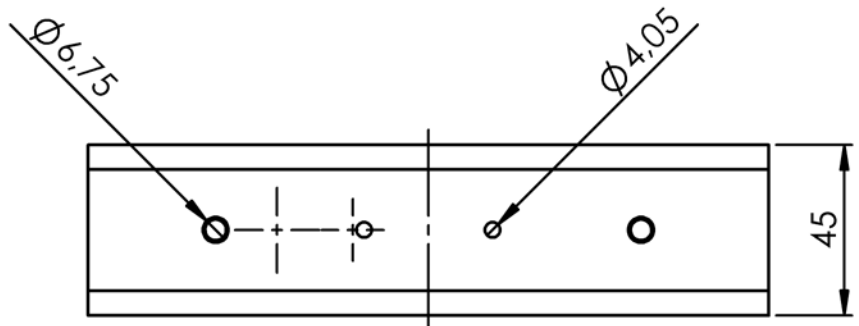
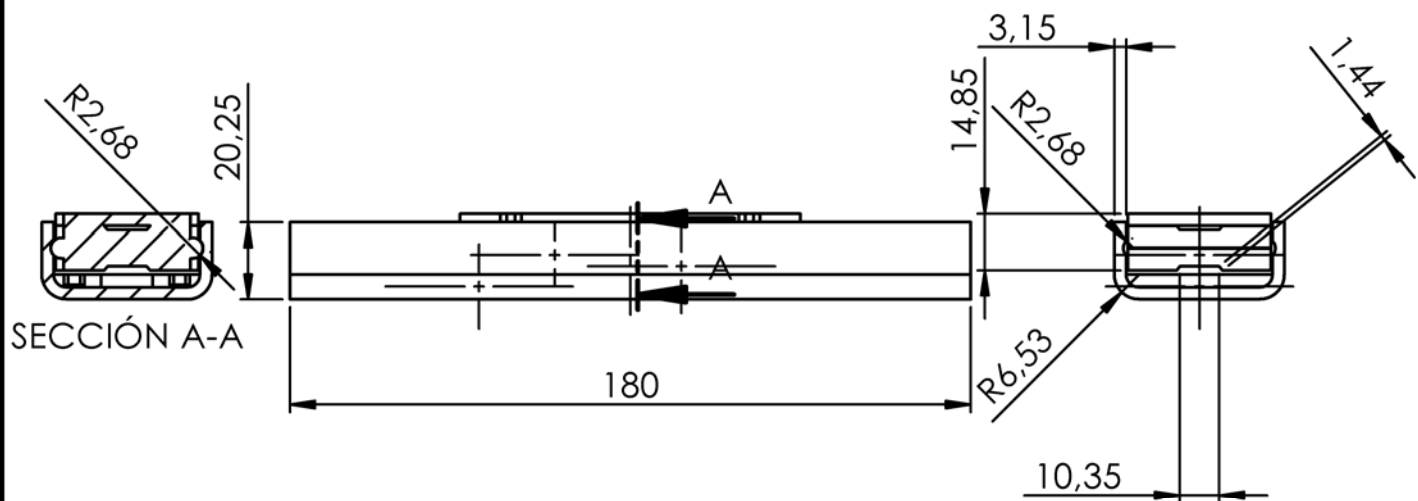
Soporte cámara

N.º DE DIBUJO

A4

ESCALA:1:2

HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:			ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN		
NOMBRE			FIRMA		FECHA		TÍTULO: <h1>Patín</h1>				
DIBUJ.			VERIF.		APROB.		MATERIAL:				
FABR.			CAUID.		PESO:		N.º DE DIBUJO				
ESCALA:1:2			HOJA 1 DE 1		A4						