

**UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
INGENIERÍA TÉCNICA INDUSTRIAL ELECTRÓNICA**



**PROYECTO FINAL DE CARRERA
DESARROLLO DE UNA DIANA REAL PARA EL GUIADO DE
UNA TUNELADORA**

AUTOR: RAFAEL PORTERO HERNÁNDEZ

TUTOR: ALBERTO JARDÓN HUETE

COTUTOR: CARLOS GONZÁLEZ DE LA VEGA

OCTUBRE DE 2009

AGRADECIMIENTOS

Este proyecto no solo es fruto de mi dedicación y esfuerzo. No hubiera sido posible sin la colaboración de muchas personas que, con su apoyo y ayuda incondicional, han puesto su granito de arena en la realización de este proyecto. Me gustaría mostrarles mi agradecimiento y dedicarles el resultado de estos duros meses de trabajo.

En primer lugar, quiero expresar mi profundo agradecimiento a mi familia, mis padres y hermanas, por todo su cariño, apoyo y comprensión durante estos años de carrera. Sin ellos, no hubiera sido posible mi desarrollo personal y profesional y nunca hubiera llegado hasta aquí.

A todos mis amigos, Héctor, Santi, Carlos, Álvaro, David, Cristóbal, Toni y tantos otros a los que sería imposible nombrar por falta de espacio, pero de los que no me olvido. Por los buenos momentos de los que hemos disfrutado y que espero continuemos pasando juntos. Por haber estado cerca y haberme apoyado cuando más lo he necesitado.

A todos mis profesores y compañeros de la Universidad. En especial, a Javier Romero, mi compañero de proyecto, por conseguir que las interminables horas que he dedicado a este proyecto fueran más llevaderas y agradables, y por sus ánimos cuando parecía que todo se venía abajo.

También, quisiera mostrar mi especial agradecimiento a cuantos han colaborado en este proyecto, con la aportación de sus conocimientos y cualificación técnica. A Julián Fortes, compañero y amigo, por su incalculable ayuda en la realización de la programación del microcontrolador. A Fernando Sandeogracias, técnico del departamento de Ingeniería de Sistemas y Automática, por sus consejos en las labores técnicas y de montaje. Y a Ernesto García, profesor y miembro del Departamento de Tecnología Electrónica, por su ayuda y sugerencias en los problemas surgidos con el filtro de línea láser.

Por último, y no por ello menos importante, quiero dar mi más sincero agradecimiento a mi tutor Alberto Jardón, por la oportunidad que me ha brindado para realizar este proyecto y por haber depositado su confianza en mí. Asimismo, quisiera agradecer a mi cotutor Carlos González, su paciencia ante las numerosas dudas que me han surgido, a lo largo de los duros meses de investigación en este proyecto.

A todos ellos, gracias.

RESUMEN

Este Proyecto Fin de Carrera titulado “Desarrollo de Prototipo de Diana para el Guiado de una Tuneladora” implica el diseño y construcción de un modelo de diana que permita mejorar en alcance, precisión y tiempo los actuales modelos disponibles para el control y cálculo de la trayectoria de una máquina tuneladora durante el proceso de perforación.

Esta diana dispone de dos paneles, uno translúcido y otro opaco, diseñados para permitir la incidencia de un haz de láser.

Para mejorar el sistema, se utiliza un láser de color verde en sustitución a los actuales láseres de color rojo, el cual tiene un alcance efectivo de 200 metros frente a los 100 metros de los de color rojo.

Para la detección del láser al incidir en los paneles de la diana, se utilizan dos cámaras que permiten obtener las imágenes a partir de las cuales se calcularán las posiciones de incidencia del láser, con lo que se obtendrán las desviaciones y ángulos que se producen en la máquina. Además, se requiere usar un inclinómetro que calcule el ángulo de giro en el eje de avance, puesto que no es visible para la diana.

Todo esto ha requerido del desarrollo de un software en el entorno de programación Matlab, que controla todos los dispositivos hardware y calcula la posición en un sistema de referencia global de la máquina tuneladora. Además, para el control del inclinómetro, se ha desarrollado un software en lenguaje C por medio del software MPlabIDE.

Por último, en este proyecto se ha desarrollado un método de comunicación con una interfaz gráfica de usuario (GUI) que permite la visualización de la trayectoria calculada en este proyecto, con el fin de conseguir una fácil interpretación por parte del operario que controla la tuneladora de los cálculos obtenidos.

ABSTRACT

The Final Project called “Developing a Target Prototype for a TBM or Tunnel Boring Machine (TBM) Guide” involves designing and building a type of target that improves the range, accuracy and time of the currently available models used to monitor and calculate a TBM’s journey during the drilling process.

This target has two panels, one is translucent and the other one is opaque, they are designed to produce a laser beam.

It uses a green coloured laser instead of the red lasers which are currently used to improve the system which has an effective range of 200 metres compared to that of the red laser range of 100 metres.

Two cameras are used to detect the laser when it shines on the target’s panels. They take pictures that are used to calculate the position of the laser beam incidence, from which the deflections and angles produced in the machine are then calculated. An inclinometer is also needed to calculate the pivoting angle in the axis of advance as the target cannot detect it.

Matlab development environment software is required to monitor all the hardware devices and calculate the position in the TBM’s whole reference system. C language software has also been developed using the MPlabIDE software to monitor the inclinometer.

Finally, communication is established in this project using a graphical user interface (GUI) which allows the calculated journey to be displayed so that the operator controlling the TBM can easily understand the calculations obtained.

ÍNDICE GENERAL

AGRADECIMIENTOS	2
RESUMEN	3
ABSTRACT	4
ÍNDICE GENERAL	5
LISTA DE FIGURAS	7
LISTA DE TABLAS	9
1. Introducción	10
1.1. Proyecto “La Ciudad Multidimensional”	11
1.1.1. Introducción	11
1.1.2. Marco de Trabajo.....	11
1.2. Objetivos del Proyecto	13
2. Estado del Arte	15
2.1. Introducción	16
2.2. 1ª Patente: Method and apparatus for controlling a tunneling machine	19
2.3. 2ª Patente: Method and system for controlling earth pressure in tunnel boring or shield machine	21
2.4. 3ª Patente: Apparatus for methos for detecting the position and orientation of a reference beam of light.....	23
2.5. 4ª Patente: Method for controlling the direction of a shield tunnelling machine and an apparatus therefor	26
2.6. 5ª Patente: Guidance System and Method for Keeping a tunnel boring machine cotinuously on a plan line	28
2.7. 6ª Patente: Automated Guided Apparatus Suitable for Toping Applications	30
2.8. Conclusiones	32
3. Especificaciones del Diseño	34
3.1. Requisitos del Sistema.....	35
3.2. Especificaciones del Diseño	36
3.2.1. Referencias del Sistema	37
3.2.2. Procesado de información	40
3.3. Requisitos e Innovaciones Funcionales del Proyecto.....	42
4. Arquitectura de la Aplicación	44
4.1. Introducción	45
4.2. Descripción de Sistema	45
4.2.1. Diana y Láser.....	45
4.2.2. Herramientas de Visión. Cámaras y láser	49

4.2.3. Inclinómetro	54
4.2.4. Conexión del Sistema	55
4.2.5. Interconexión con otros programas.....	56
4.3. Herramientas de Programación	63
4.3.1. MATLAB	63
4.3.2. MPLab IDE	65
4.4. Descripción de los Programas	66
4.4.1. Programas creados en Matlab	66
4.4.2. Programa inclinómetro	78
5. Manual de Uso.....	82
5.1. Manejo del Programa.....	83
5.2. Ajuste de las Cámaras.....	83
5.3. Calibración de las Cámaras	87
5.4. Manejo del Inclinómetro.....	93
6. Conclusiones	96
6.1. Análisis Crítico.....	97
6.2. Trabajos Futuros.....	98
6.3. Conclusiones	100
BIBLIOGRAFÍA.....	104
ANEXOS	107
ANEXO A. Programas Matlab.....	108
ANEXO B. Programa Inclinómetro.....	127
ANEXO C. Cámaras.....	130
ANEXO D. Láser.....	136
ANEXO E. Inclinómetro	140
ANEXO F. Microcontrolador.....	158
ANEXO G. Placa de Desarrollo	179

LISTA DE FIGURAS

Figura 1.- Perfil de Tuneladora (1ª Patente).....	20
Figura 2.- Variables en el límite de la tuneladora (1ª Patente)	20
Figura 3.- Planta de la tuneladora y sistema de guiado (1ª Patente).....	21
Figura 4.- Sección de perfil en tuneladora (2ª Patente).....	22
Figura 5.- Vista en perspectiva del detector de láser.....	24
Figura 6. - Vista en planta del detector con las desviaciones producidas	25
Figura 7.- Perfil de la máquina (Patente 4).....	26
Figura 8.- Vista en perspectiva de la máquina (Patente 5)	29
Figura 9.- Vista de los objetivos (Patente 5).....	30
Figura 10.- Planta del sistema (Patente 6)	31
Figura 11.- Vista en planta y alzado de los sistemas (Patente 6).....	32
Figura 12.- Esquema de perforación para máquina de hinca de tubos	35
Figura 13.- Vista en perspectiva del sistema de referencia global y sus sentidos de giro	38
Figura 14.- Sistemas de referencia para la Diana.....	38
Figura 15.- Sistemas de referencia para la GUI.....	39
Figura 16.- Sistema escalado para la representación de la GUI	39
Figura 17.- Esquema de las variables usadas para cada bloque del sistema	41
Figura 18.- Vista exterior de la Diana con dimensiones	46
Figura 19.- Vista del Láser con descripción de las funciones	48
Figura 20.- Vista interior de la Diana.....	51
Figura 21.- Esquema interior de la colocación de las cámaras e inclinómetro	52
Figura 22.- Vista del inclinómetro montado en la placa de desarrollo.....	54
Figura 23.- Esquema de sistema completo de guiado	56
Figura 24.- Diagrama de intercambio de variables para los tres sistemas de guiado ...	57
Figura 25.- Pantalla de Matlab Script Node en LabView.....	60
Figura 26.- Diagrama de flujo de las variables de los sistemas respecto a los archivos de intercambio	61
Figura 27.- Esquema de construcción de estructura de librería DLL	63
Figura 28.- Vista de la ventana de trabajo de Matlab	65
Figura 29.- Vista de ventana principal MPLabIDE	66
Figura 30.- Diagrama de la arquitectura de la aplicación	68

Figura 31.- Diagrama de flujo de las funciones principales del software.....	69
Figura 32.- Diagrama de flujo de la función “vision.m”	70
Figura 33.- Imágenes de ventana y fondo de la diana obtenidas por las cámaras.....	71
Figura 34.- Imágenes de las cámaras de la ventana y fondo de la diana binarizadas..	71
Figura 35.- Visualización de ejemplo de corrección de la distorsión.....	73
Figura 36.- Imagen binarizada en la que posteriormente se han eliminado los objetos menores a un tamaño definido	74
Figura 37.- Imágenes de ventana y fondo de la diana con los centroides marcados....	75
Figura 38.- Esquema del sistema de referencia del inclinómetro.....	76
Figura 39.- Imagen de corrección del ángulo gamma	76
Figura 40.- Esquema de colocación de la diana dentro de la tuneladora.....	78
Figura 41.- Diagrama de flujo del programa del inclinómetro	81
Figura 42.- Vista del hardware en el interior de la diana	84
Figura 43.- Imagen de la cámara instalada en la diana.....	84
Figura 44.- Vista de la colocación de las plantillas de obtención de los bordes de la imagen.....	85
Figura 45.- Imagen de la obtención de los valores pixel de los bordes de las imágenes de las cámaras.....	86
Figura 46.- Ventana de inicio de Toolbox Calibration Camera	88
Figura 47.- Menú principal de Toolbox Calibration Camera.....	88
Figura 48.- Ventanas de selección de bordes de Toolbox Calibration Camera	90
Figura 49.- Ventana de detección de cuadros en Toolbox Calibration Camera.....	91
Figura 50.- Proyección de los parámetros de calibración en imágenes en Toolbox Calibration Camera	92
Figura 51.- Obtención de la imagen sin distorsión.....	93
Figura 52.- Imagen de la placa de desarrollo y el inclinómetro	94
Figura 53.- Ventana de Propiedades del Sistema de Windows.....	95
Figura 54.- Vista del Administrador de dispositivos de Windows	95
Figura 55.- Propuesta futura de rediseño de la diana.....	99

LISTA DE TABLAS

Tabla 1.- Dimensiones de la Máquina Tuneladora de Hincas de Tubos37

Tabla 2.- Características de las dianas comerciales47

Tabla 3.- Características generales del filtro de la cámara53

Tabla 4.- Comandos de instrucción para el manejo del inclinómetro.....79

Tabla 5.- Tabla de comparación de precisiones entre versiones comerciales y el
prototipo del proyecto102

Tabla 6.- Comparación de Desviaciones y ángulos entre modelos de diana
comerciales y el prototipo realizado102

1. Introducción

En este primer apartado se va introducir el proyecto “La Ciudad Multidimensional”, comentando los objetivos y líneas de desarrollo que supone, para a continuación, analizar el marco de trabajo que engloba el desarrollo del presente proyecto.

Por último, se enumerarán los objetivos que se quieren alcanzar con la realización de este proyecto.

1.1. Proyecto “La Ciudad Multidimensional”

1.1.1. Introducción

El Proyecto Científico-Tecnológico singular de carácter Estratégico “La Ciudad Multidimensional” con referencia PSE – 380000 – 2006 – 6 es una iniciativa conjunta de la Industria, los Centros Tecnológicos, Universidades, Centros públicos y privados de investigación y otras organizaciones, para acometer el reto que se plantea en nuestra sociedad, con el fin de obtener unas ciudades más confortables, sostenibles y puestas al día con espacios para las personas, eliminando la mayor parte de las molestias del tráfico, promoviendo los transportes colectivos y sociales.

Esta iniciativa promueve la sostenibilidad de nuestro entorno, la seguridad de las personas durante las fases de construcción y uso de los servicios e infraestructuras (canalizaciones de agua, gas y electricidad, sistemas de recogida de basura subterránea, transporte subterráneo, etc.) generando espacios libres para el disfrute de los ciudadanos y transformando nuestro entorno urbano, congestionado, ruidoso y contaminado, en zonas agradables para vivir, donde las personas sean el centro de nuestro diseño y se imagine y construya la ciudad para el desarrollo integral del ser humano.

Esta iniciativa estratégica se dirige al acercamiento de los servicios anteriormente mencionados a todos los ciudadanos, teniendo especial cuidado con las personas discapacitadas, la gente con movilidad reducida, los ancianos y los niños, teniendo además presente la multiculturalidad de nuestro entorno.

1.1.2. Marco de Trabajo

Dentro del proyecto “La Ciudad Multidimensional”, se desarrolla este marco de trabajo, en el que se va a realizar el desarrollo teórico y práctico, encargado por

la empresa Eurohinca, con el fin de analizar los sistemas de guiado para máquinas tuneladoras y extraer conclusiones.

A continuación, se expone un análisis de las técnicas actuales para los sistemas de guiado.

Para la ejecución correcta de un túnel desde el punto de vista topográfico es necesario que se cumplan los siguientes requisitos:

- Seguir la alineación específica exigida por el cliente.
- Obtener una mayor exactitud en la realización del trabajo, lo cual implica una buena precisión en la salida de la tuneladora y, por tanto, en todo el trayecto de la misma.

Para ello es necesario disponer de un sistema de guiado que:

- Establezca unas especificaciones temporales para que el operador de la tuneladora pueda seguir como referencia en la perforación del túnel.
- Presente la información de tal manera que el operario de la tuneladora pueda conducirla lo más precisamente posible.

En la actualidad, debido a los grandes rendimientos que ofrecen las tuneladoras, (30 metros de perforación al día), es necesario:

- Disponer de sistemas de guiado que ofrezcan la información de la posición de la tuneladora al instante, para que el operario pueda conducirla lo más cerca posible del eje diseñado.
- Sistemas de guiado que incidan en menor medida en la actividad de los trabajadores, para evitar pérdidas de tiempo en el trabajo de los mismos.

Debido a todos los requisitos anteriormente mencionados, hace unos años que se descartó la topografía clásica para el replanteo de túneles realizados mediante tuneladoras, ya que los inconvenientes que presentaba eran:

- Lentitud en la presentación de la información.
- Falta de visibilidad para un replanteo eficaz.

Por este motivo, necesitamos un sistema de guiado que incorpore a los conceptos clásicos de replanteo los sistemas más modernos de cálculo por ordenadores, láseres, estaciones servomotorizadas y giróscopos, que nos permita conseguir sistemas de guiado que:

- Presenten información al instante de la posición de la tuneladora con relación al eje diseñado.
- Ofrezcan una precisión excelente, normalmente por encima de las capacidades de conducción de las tuneladoras, evitando los errores de varios centímetros con los que se realizan los túneles por medio de las actuales técnicas, debido a la imposibilidad de conducir la tuneladora exactamente por el eje indicado por el sistema de guiado.

1.2. Objetivos del Proyecto

Este proyecto, dentro de la necesidad de mejora de un sistema de guiado para máquinas tuneladoras, implica el desarrollo y construcción de un sistema de guiado por medio de una diana láser e integración de todos los componentes necesarios para obtener un sistema que aporte los datos necesarios.

Por tanto, los objetivos que persigue la realización de este proyecto son:

- Concepción y construcción de un nuevo prototipo de diana láser para el guiado.
- Realizar un sistema que permita la detección de un rayo láser verde, en lugar de los actuales láseres rojos de menor alcance, para aumentar la distancia a la que se es capaz de obtener la posición de la máquina tuneladora, mediante la utilización de cámaras en vez de pantallas fotosensibles.
- Realizar la integración y conexión de un inclinómetro que permita conocer los ángulos que no puedan ser obtenidos por medio de la diana.
- Realizar un software que controle todos los dispositivos y calcule, por medio de los datos adquiridos del hardware, los datos necesarios para que el operario de la máquina conozca su posición en cada instante.
- Mejorar la velocidad de refresco en que los datos son tomados y ofrecidos al operario, para posibilitar la mejora del guiado, al tener mayor cantidad de información en el tiempo.
- Conseguir que estos datos sean transmitidos a una interfaz gráfica de usuario GUI, y que permita la compresión de los mismos.

Todo esto supone la integración de componentes tanto hardware como software, bajo el paradigma de integración de sistemas, con el objetivo de desarrollar un prototipo mejorado con respecto a los actuales.

2. Estado del Arte

2.1. Introducción

Como introducción a las técnicas de sistemas de guiado de tuneladoras, se van a comentar diversas técnicas desarrolladas para el control y guiado de distintos tipos de tuneladoras, con métodos que se desarrollaron para conseguir un control de la máquina, atendiendo a dificultades tales como el terreno.

Se han elegido seis patentes americanas, algunas con similitudes a este proyecto por hacer uso de técnicas de guiado con láser, y otras que varían totalmente en el método, para tener una visión global de las posibilidades que se tienen a la hora de conseguir estimar los parámetros que facilitan el gobierno de estas máquinas.

A continuación, se explica los motivos que llevan a patentar una invención, atendiendo a las características que tiene que satisfacer y los beneficios que implican obtenerla. Se ha de tener en cuenta que en muchos casos, la tecnología que implica mayor competitividad solo está disponible en formato de patentes y no se suele divulgar en otros medios.

Una patente es un conjunto de derechos exclusivos garantizados por un gobierno o autoridad al inventor de un nuevo producto (material o inmaterial) susceptible de ser explotado industrialmente para el bien del solicitante de dicha invención durante un espacio limitado de tiempo, generalmente veinte años desde la fecha de solicitud. Después de la caducidad de la patente cualquier persona puede hacer uso de la tecnología de la patente sin la necesidad del consentimiento del titular de esta.

El titular de una patente puede ser una o varias personas nacionales o extranjeras, físicas o morales, combinadas de la manera que se especifique en la solicitud, en el porcentaje ahí mencionado. Los derechos de las patentes caen dentro de lo que se denomina propiedad industrial y, al igual que la propiedad inmobiliaria, estos derechos se pueden transferir por actos entre vivos o por vía sucesoria, pudiendo: rentarse, licenciarse, venderse, permutarse o heredarse. Por ello, las patentes deberían servir a los efectos de una difusión efectiva y rápida de las nuevas ideas en tecnología, mejorando el acceso a la tecnología.

Beneficios de una patente

Algunos de los argumentos habituales a favor de las patentes mantienen que los beneficios que una patente le otorga a un inventor son:

- Motiva la creatividad del inventor, ya que ahora tiene la garantía de que su actividad inventiva estará protegida durante 20 años y será el único en explotarla.
- Si la patente tiene buen éxito comercial o industrial, el inventor se beneficia con la licencia o licencias de explotación que decida otorgar a terceras personas.
- Evita el plagio de sus inventos.
- Debido a que la actividad inventiva aunque no se realizara la explotación industrial o sólo se utilizara para sí, se mantendrían los derechos sobre ella, el inventor siempre dará a conocer, publicitar y explicar los beneficios que su invento tiene.
-

Por su parte el Gobierno a través de la patente promueve la creación de invenciones de aplicación industrial, fomenta el desarrollo y explotación de la industria y el comercio, así como la transferencia de tecnología.

¿Qué es lo que no se considera invención?

- Los principios teóricos o científicos.
- Los descubrimientos que consistan en dar a conocer o revelar algo que ya existía en la naturaleza, aun cuando anteriormente fuese desconocido para el hombre.
- Los esquemas, planes, reglas y métodos para realizar actos mentales, juegos o negocios y los métodos matemáticos.
- Las formas de presentación de información.
- Las creaciones estéticas y las obras artísticas o literarias.
- Los métodos de tratamiento quirúrgico, terapéutico o de diagnóstico aplicables al cuerpo humano y los relativos a animales. Además, la yuxtaposición de invenciones conocidas o mezclas de productos conocidos, su variación de uso, de forma, de dimensiones o de materiales, salvo que en realidad se trate de su combinación o fusión de tal manera que no puedan funcionar separadamente o que las cualidades o funciones características de las mismas sean modificadas para obtener un resultado industrial o un uso no obvio para un técnico en la materia.

Lo que no se puede patentar

- Los procesos esencialmente biológicos para la producción, reproducción y/o propagación de plantas y animales.
- El material biológico y genético tal como se encuentran en la naturaleza.

- Las razas animales.
- El cuerpo humano y las partes vivas que lo componen.
- Las variedades vegetales y minerales.
-

Vigencia de una patente

La vigencia de las patentes depende de cada país. En España tienen una vigencia de 20 años improrrogables y los modelos de utilidad tienen una vigencia de 10 años igualmente improrrogables. Cuando la patente o modelo de utilidad expira, expira asimismo la protección, y la invención pasa a pertenecer al dominio público; es decir, el titular deja de tener derechos exclusivos sobre la invención, que pasa a estar disponible para la explotación comercial por terceros interesados.

El derecho exclusivo de explotación de la invención patentada confiere a su titular las siguientes prerrogativas:

- Si la materia objeto de la patente es un producto, el derecho de impedir a otras personas que fabriquen, usen, vendan, ofrezcan en venta o importen el producto patentado, sin consentimiento.
- Si la materia objeto de la patente es un proceso, el derecho de impedir a otras personas que utilicen ese proceso y que usen, vendan, ofrezcan en venta o importen el producto obtenido directamente de ese proceso, sin su consentimiento.

Prioridad

Cuando se solicite una patente después de hacerlo en otros países, se podrá reconocer como fecha de prioridad la de presentación en aquel que lo fue primero, siempre que se presente en España dentro de los plazos que determinen los Tratados Internacionales o, en su defecto, dentro de los doce meses siguientes a la solicitud de patente en el país de origen.

Derecho del inventor

En todos los casos, el o los inventores tienen este derecho:

- Reconocimiento al nombre
- Solicitar la patente

Derecho de explotación

El titular de la patente tiene lo que se denomina un derecho negativo sobre la tecnología patentada. Este derecho le permite impedir que terceros sin su consentimiento:

- Fabriquen, usen, vendan o importen el producto patentado.
- Usen el proceso patentado y usen, vendan o importen el producto obtenido de ese proceso.
- El titular puede permitir alguna de las actividades anteriores a determinada persona o empresa, otorgando una licencia y recibiendo un pago que se conoce como regalía. También puede transferir la titularidad de la patente o ceder su derecho, mediante un pago fijo. Después de esto, el titular original ya nada tiene que ver con la explotación de esa patente.

2.2. 1ª Patente: Method and apparatus for controlling a tunneling machine

Número de patente: 4.023.861

Autor: Giinter Schnell

Fecha: 21 de Noviembre de 1977

La máquina tuneladora especificada para su guiado, se compone principalmente de un cuerpo motorizado, para permitir el avance de la máquina, además de una cabeza orientable en vertical y horizontal por medio de un sistema hidráulico. Además, se requiere de un sistema de recogida y eliminación de los residuos que se producen por el avance de la cabeza perforadora.

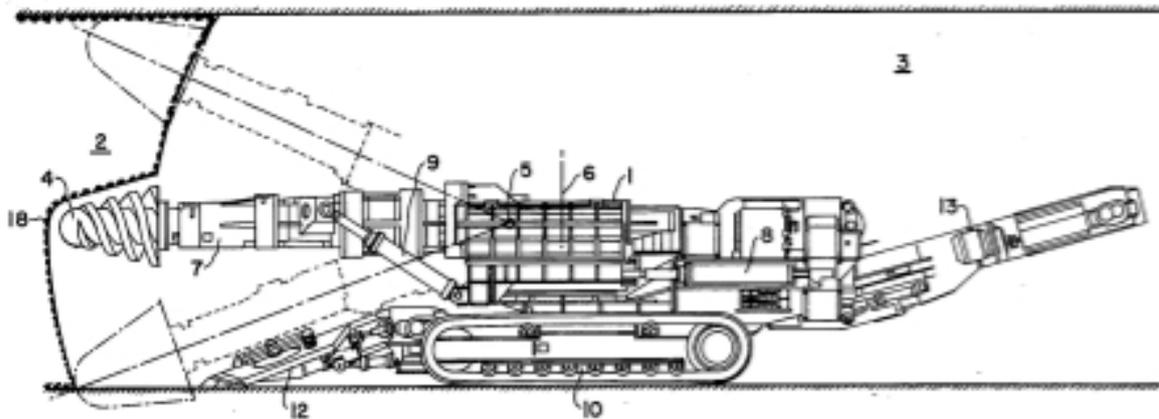


Figura 1.- Perfil de Tuneladora (1ª Patente)

El sistema de guiado hace uso de un láser, una plantilla y un detector para medir las desviaciones verticales y horizontales de la tuneladora, mientras se obtiene los giros de la máquina mediante el uso de dos giróscopos. La salida del sistema se realiza mediante impulsos eléctricos que van a parar a un ordenador, el cual se encarga, mediante el uso de un programa informático, del cálculo de la posición de la tuneladora.

El detector láser requiere de dos servomotores que permiten su desplazamiento, tanto horizontal como vertical, para estar en todo momento orientados con la línea del láser y permitir su detección.

El sistema se basa en que, una vez conocida la posición del cuerpo central de la tuneladora, podemos calcular los movimientos necesarios en la cabeza de corte, para obtener el perfil deseado del túnel. Además, una vez conocida la posición del cuerpo de la máquina, se pueden conocer las limitaciones en el movimiento de la herramienta de corte, tanto en horizontal como en vertical, y que se llevan a cabo por medio de actuadores hidráulicos.

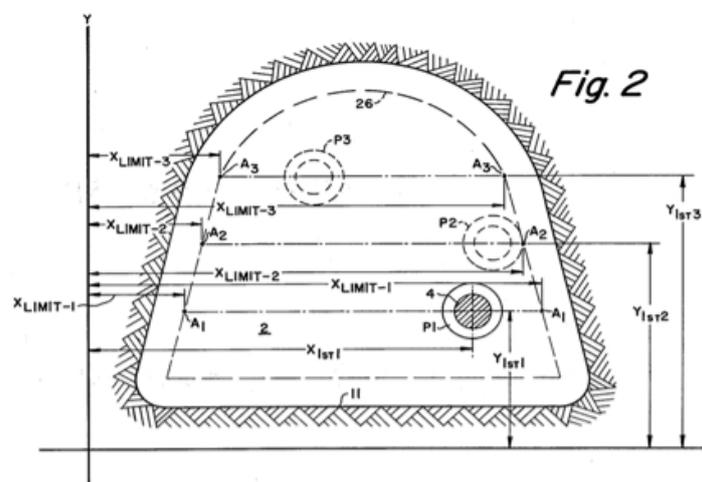


Figura 2.- Variables en el límite de la tuneladora (1ª Patente)

Las señales correspondientes a los valores del punto de referencia, se van corrigiendo por medio de otras señales correspondientes a las desviaciones de la máquina. La corrección de las señales en realidad corresponde a la posición de las desviaciones del receptor de rayo láser. Los valores de las coordenadas se corrigen, no sólo por las señales correspondientes a las desviaciones anteriores, sino también por la rotación de la plantilla, al poder ser orientada. De esta manera, el rango de movimiento de la herramienta de corte sigue siendo punto de referencia de la posición definida por el rayo láser. En este sistema de control, el receptor láser está alineado al detectar el haz de láser, mientras que la plantilla se ajusta únicamente por gravedad.

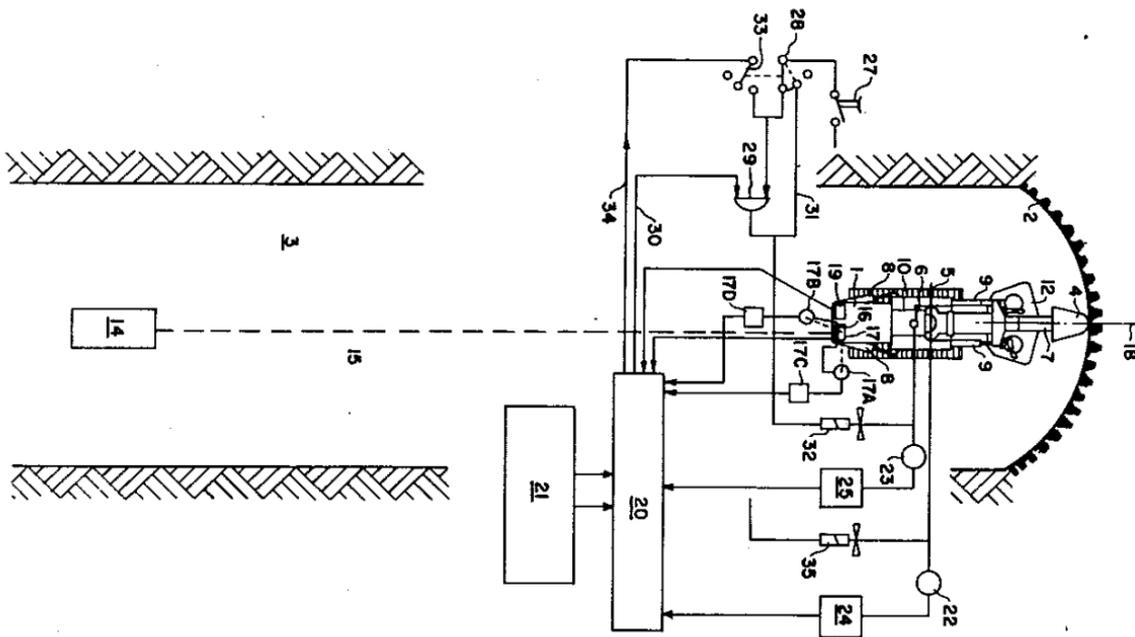


Figura 3.- Planta de la tuneladora y sistema de guiado (1ª Patente)

2.3. 2ª Patente: Method and system for controlling earth pressure in tunnel boring or shield machine

Número de Patente: 4.167.289

Autores: Kozo Ono; Masao Suda

Fecha: 11 de Septiembre de 1979

En esta patente se explica un método para el control en una tuneladora de tipo escudo, mediante el equilibrado de las presiones del frente o rueda de corte y la cámara de extracción. Este tipo de control se diseñó específicamente para el control de presiones en terrenos blandos, en los que se producen fáciles desprendimientos y emanaciones de agua.

En la siguiente imagen se muestra un perfil de la máquina, donde se pueden apreciar los elementos principales, como la cabeza de perforación, la cual está colocada en un bastidor e impulsada por un motor hidráulico y protegido por un mamparo de la propia cabeza y los residuos que desprende a gran velocidad; el sistema de extracción de residuos, mediante un tornillo sinfín y una compuerta de apertura por medio de un gato hidráulico; una campana que se dispone a lo largo de toda la circunferencia exterior de la maquina, para reforzar la estructura; y el sistema de empuje por gatos hidráulicos para el avance de la máquina:

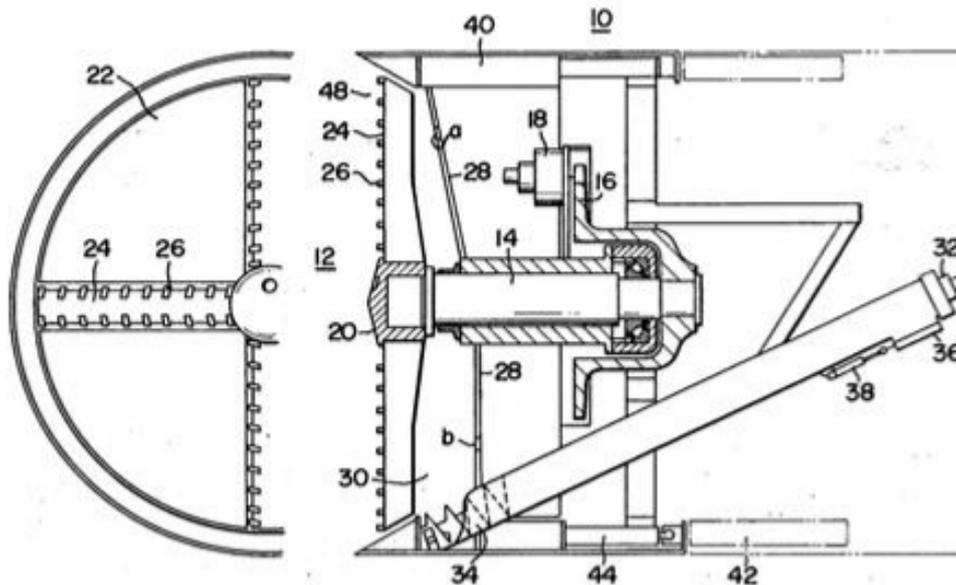


Figura 4.- Sección de perfil en tuneladora (2ª Patente)

El sistema se basa en la lectura de hasta 66 sensores de presión para comprobar que la presión sobre el terreno produce un adecuado volumen de residuos, sin llegar a provocar el desprendimiento de la pared. La mayoría de los sensores se colocan en la campana de protección, consiguiendo una monitorización de la mayor área posible de la pared sobre la que se está realizando el trabajo y evitando en todo momento un desprendimiento incontrolado o fuera de lo normal.

Gracias a esta monitorización, y mediante control por un programa informático, se llevan dos límites de presión, uno inferior, para prevenir el desprendimiento de la pared, y otro máximo de presión, para evitar el aumento de la tierra

perforada. También se encarga este control informático, de modificar el movimiento de rotación del tonillo encargado de extraer el material que ya ha sido perforado, para evitar en cualquier nivel de presión la obstrucción de la máquina, además de controlar el flujo de bentonita o material lubricante, necesario para una óptima rotación de corte.

Una característica importante de este sistema, es la capacidad de control automático de todo el proceso, sin llegar a necesitar de la intervención de ningún operario, los cuales se limitan a labores de inspección en condiciones normales de trabajo. En caso de llegar a una anomalía en el sistema, se ha previsto la conmutación automática a control manual, para situaciones en el que el control informático es incapaz de llegar a una solución para asignar los parámetros al sistema.

2.4. 3ª Patente: Apparatus and method for detecting the position and orientation of a reference beam of light

Número de Patente: 4.786.178

Autor: Ted L. Teach

Fecha: 22 de Noviembre de 1988

Este sistema de guiado permite mejorar los sistemas que se basan en el cálculo de la posición de una máquina tuneladora, mediante la detección de un haz de láser. Hasta la fecha, la mayoría de los sistemas de guiado se basaban en calcular que la máquina estaba en la trayectoria deseada manteniendo el láser en el centro de un detector, sabiendo que se había producido un error en caso de que saliera de ese centro. El problema con este tipo de sistemas es la dificultad de colocar en curso de nuevo a la máquina, sin tener una clara posición del error cometido, además de que en el caso de que se volviera a centrar la máquina, solo se tendría plena seguridad de que la parte trasera de la máquina estaba correctamente orientada puesto que es donde va colocado el detector de láser, no siendo así la parte frontal debido a la considerable longitud que tiene la máquina.

Para eliminar estas deficiencias, esta patente define un método basado en dos detectores para el haz de luz. El sistema incluye un primer detector, el cual es un espejo semi-translúcido, que desvía parte del rayo al lateral del aparato, mientras deja pasar otra parte del haz de luz. Un segundo detector se coloca a continuación, siguiendo el mismo método, desviando la luz de incidencia

también al lateral del invento. En los laterales se puede colocar cualquier detector habitual tales como, por ejemplo, una serie de células fotoeléctricas o un dispositivo de fotocélula de división, para obtener las posiciones de incidencia de los dos haces desviados. La colocación habitual de los espejos es de 45 grados, con lo cual, el haz de luz será desviado en 90 grados de su trayectoria original, para facilitar los cálculos a la hora de obtener la posición relativa.

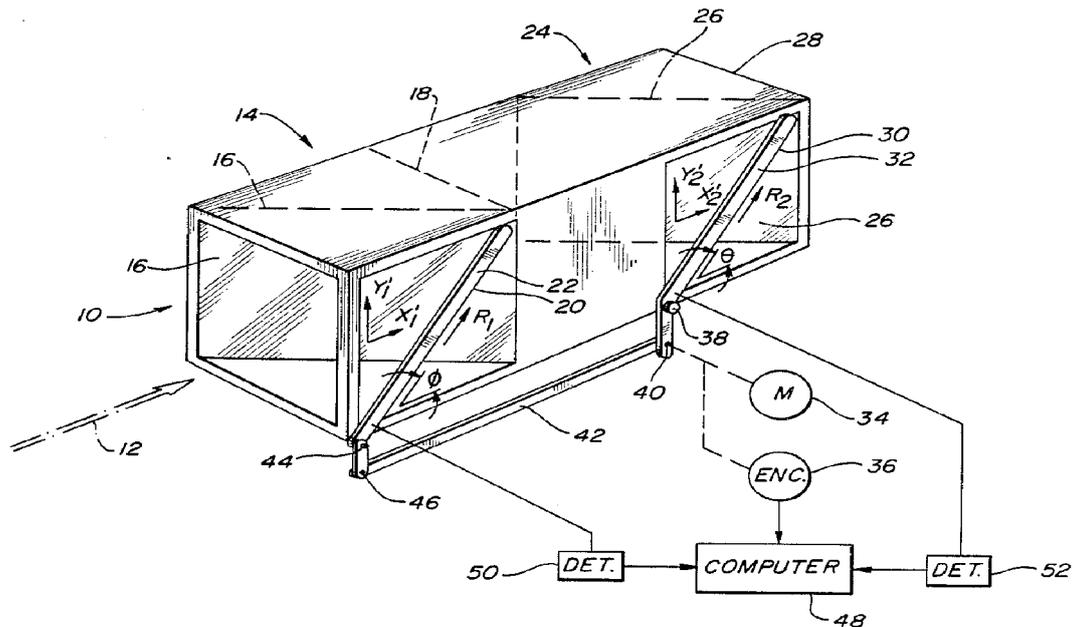


Figura 5.- Vista en perspectiva del detector de láser

Para realizar los cálculos matemáticos necesarios para la obtención de la corrección en la trayectoria de la máquina, el sistema incluye un ordenador de ámbito general, el cual se encarga de calcular la posición por medio de los puntos que han sido determinados en las dos proyecciones del haz, al incidir en los detectores de luz fotosensibles.

En la siguiente vista en planta, podemos apreciar las desviaciones del láser, y las medidas de desviaciones del haz que se miden:

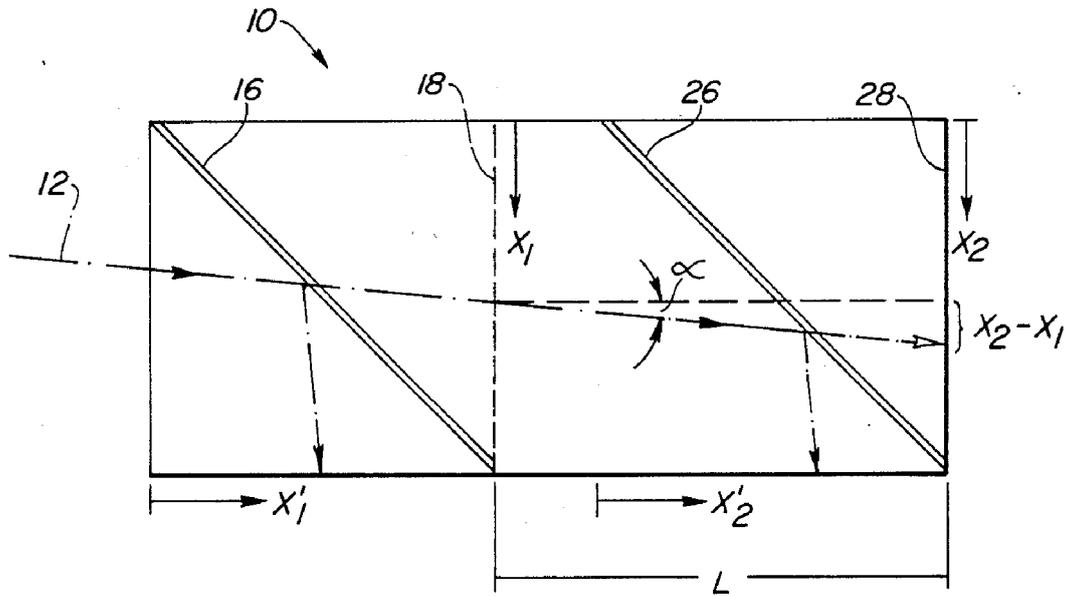


Figura 6. - Vista en planta del detector con las desviaciones producidas

Obteniendo las siguientes relaciones entre desviaciones y ángulos en el sistema, y siendo X1 e Y1 las coordenadas horizontal y vertical respectivamente en la incidencia con el primer plano, y X2 e Y2, las coordenadas horizontal y vertical respectivamente en la incidencia con el segundo plano:

$$\begin{aligned}
 X_1 &= X'_1 = R_1 \cos \phi. \\
 Y_1 &= Y'_1 = R_1 \sin \phi. \\
 X_2 &= X'_2 = R_2 \cos \theta \\
 Y_2 &= Y'_2 = R_2 \sin \theta
 \end{aligned}
 \quad \text{ec.(1)}$$

A partir de estas desviaciones, y por medio de cálculos trigonométricos básicos, se pueden determinar los ángulos del sistema por medio de las siguientes fórmulas:

$$\text{yaw } \alpha = \tan^{-1} \left(\frac{X_2 - X_1}{L} \right), \quad \text{pitch } \beta = \tan^{-1} \left(\frac{Y_2 - Y_1}{L} \right), \quad \text{ec.(2)}$$

De estas ecuaciones 1 y 2, y recordando que se han colocado los espejos de refracción con un ángulo de 45 grados respecto de la trayectoria original del haz de luz para un total de 90 grados en los detectores, resultan las fórmulas siguientes:

$$\text{yaw } \alpha = \tan^{-1} \frac{R_2 \cos \theta - R_1 \cos \phi}{L}$$

$$\text{pitch } \beta = \tan^{-1} \frac{R_2 \sin \theta - R_1 \sin \phi}{L}$$

ec.(3)

Otra característica del sistema, consiste en dotar a los paneles de incidencia de unas escobillas a modo de parabrisas, para permitir su limpieza, las cuales se mueven por medio de motores controlados por el mismo ordenador que se encarga de los cálculos de computación.

2.5. 4ª Patente: Method for controlling the direction of a shield tunnelling machine and an apparatus therefor

Número de Patente: 5.501.548

Autores: Mitsutoshi Hayashi; Akitaka Fujita; Shingo Nagashima; Isao Fujimori.

Fecha: 26 de Marzo de 1996

Esta patente se refiere a un método de control de la dirección de propulsión para el revestimiento de túneles y los aparatos que implica esta situación.

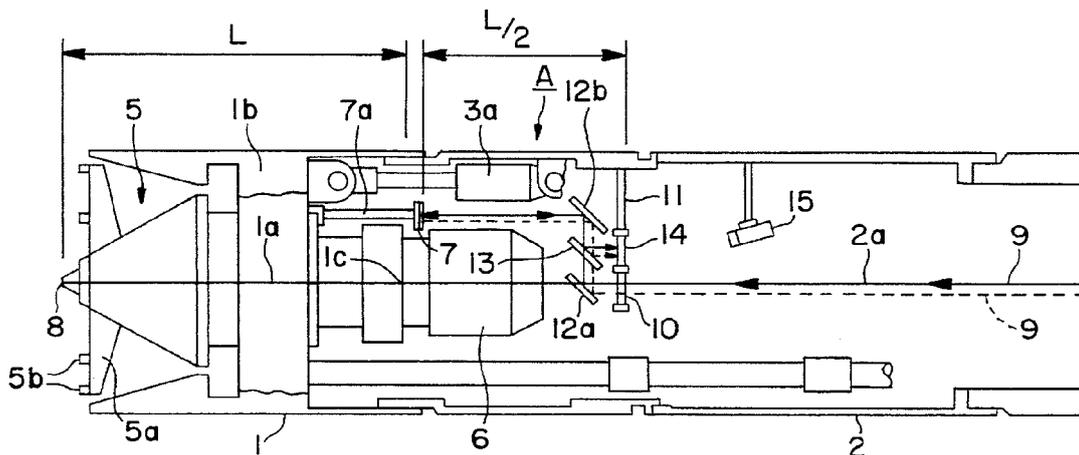


Figura 7.- Perfil de la máquina (Patente 4)

El objetivo del presente invento, es proveer de un método de control eficiente y automático, para el cálculo avanzado de la trayectoria de una máquina tuneladora. El segundo objetivo consiste en obtener un método que consiga

ajustar automáticamente la dirección del avance del escudo de la máquina tuneladora.

Para lograr el primer objetivo, la presente invención se caracteriza por tratarse de un método de control a partir de la dirección del revestimiento para una máquina tuneladora, que cuenta con un cabezal de corte para penetrar en el frente del terreno, y el cual tiene una pantalla conectada a través de la cola de control con la propia cabeza de corte. El método consiste en enviar un rayo de luz láser direccional (9) hacia la cabeza de corte, según la línea prevista para la colocación de un nuevo conducto, el cual generará una proyección sobre el objetivo colocado en el escudo de cola. A partir del momento en que se obtiene el reflejo del haz de luz de un reflector (12a), se toma una imagen (15) del objetivo, y cuando se observa que un punto de haz de luz proyectado sobre la pantalla se desplaza del punto original, mientras que el escudo tuneladora es impulsado, se puede controlar el impulso de la máquina para hacer coincidir el punto reflejado con el punto original que indica la dirección que se quiere tomar.

A fin de lograr el segundo objeto de la presente invención, se tienen los siguientes aparatos de control de dirección: cabezal de corte con un reflector para reflejar un haz de luz proveniente del láser; un escudo de cola articulado que cuenta con un control direccional y que está unido al cabezal de corte, y donde además está colocado un espejo para permitir la devolver el haz del láser proveniente tanto del cabezal de corte, como del láser; una cámara de televisión para obtener una foto del blanco originado en la pantalla; un procesador de imágenes para procesar la imagen tomada por una cámara de televisión, con el que se detecta la posición del punto en el objetivo y la pantalla; una memoria en la que se almacena la posición de los puntos iniciales proyectados, tomados como origen al comienzo del proceso de avance de la máquina; un comparador en el que una posición del punto actual de haz de luz proyectado sobre la pantalla, durante la propulsión del escudo de la máquina, se compara con la posición del punto original y cuando detecta que el lugar actual de haz de luz proyectada en la pantalla se desplaza del lugar original, genera una señal eléctrica que corresponde al desplazamiento originado.

De acuerdo con esto, al comenzar a impulsar el escudo de la máquina, un punto de proyección del haz de luz del rayo láser es proyectado en la pantalla y se define como el origen. Cuando el escudo de la máquina es desviado de la línea prevista, mientras que el escudo de la tuneladora está avanzando, el desplazamiento de la pantalla de la tuneladora está acompañado por la nueva proyección del haz de láser sobre la pantalla de desplazamiento fuera del origen. Por lo tanto, cuando es detectado un desplazamiento fuera del origen, el propulsor de dirección del escudo de la máquina puede ser ajustado por

medio del control direccional, para que el lugar de haz de luz proyectado sobre la pantalla coincida con el origen y, por tanto, con la dirección prevista.

2.6. 5ª Patente: Guidance System and Method for Keeping a tunnel boring machine continuously on a plan line

Número de Patente: 5.529.437

Autor: Mark S. Filipowski; Steven R. Wardwell; Michael L. Shank

Fecha: 13 de Septiembre de 1994

Esta invención se refiere a un sistema de guiado para máquinas tuneladoras, y más en particular, a un sistema para mantener en la trayectoria deseada la máquina, según una planificación previa. Este sistema de guiado, aun con un uso principal para la realización de túneles, también puede ser utilizado en la profundización de pozos y otras aplicaciones de minería.

En vista de lo anterior, es objeto primordial de la invención proporcionar un método rápido y fiable para el mantenimiento de la máquina en la línea prevista de perforación. El nuevo sistema elimina la necesidad de tener que realizar cálculos de posición por parte del operario que controla el guiado, lo que significa evitar errores y momentos de confusión a la hora de ajustar los parámetros relativos al movimiento de la máquina.

Otro objeto de la presente invención es la utilización de un ordenador de a bordo, el cual permite, a partir de los datos obtenidos, mostrar gráficas que simbolicen las posiciones anteriores y actual de la máquina tuneladora, desde una perspectiva vertical y horizontal de la misma.

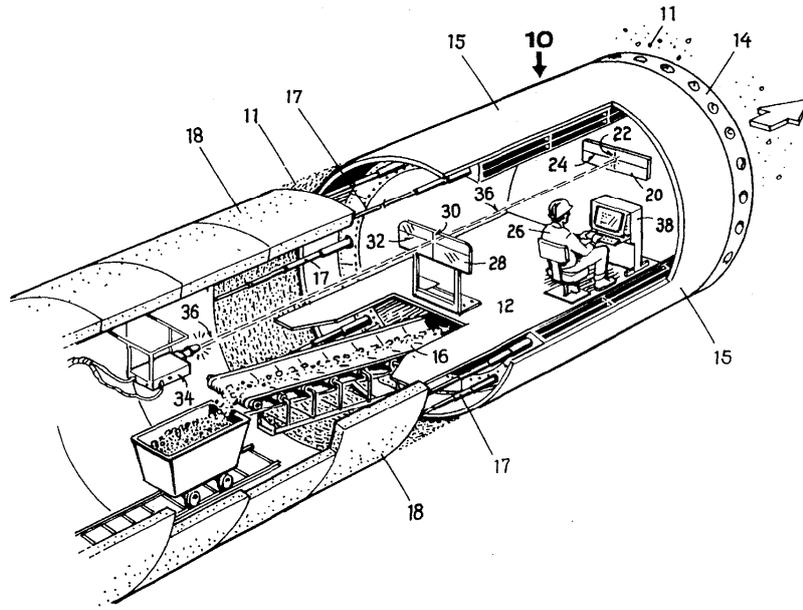


Figura 8.- Vista en perspectiva de la máquina (Patente 5)

Además de evitar errores en cálculos gracias a la automatización de los procesos de guiado, teniendo el operador que limitarse a realizar operaciones de inspección, esto supone un gran incremento en la velocidad de operación de la máquina en el proceso de perforación. Gracias a ello, la productividad se ve incrementada, eliminando los costes de tener la máquina fuera de funcionamiento.

Para llevar a cabo los cálculos de posicionamiento, se hace uso de un emisor de rayo láser, apuntando a dos blancos situados dentro de la máquina tuneladora, siendo el primero translúcido para dejar pasar al rayo hacia el segundo, el cual es de material opaco. Estos blancos se encuentran distanciados a un valor constante, normalmente de 4 metros (si bien pueden ser modificados en función de las dimensiones de la máquina usada).

A partir de las lecturas actuales de los puntos que se generan en los blancos por el rayo láser, y teniendo una primera referencia de valor en el origen, esos valores son enviados al ordenador para realizar el cálculo que permita obtener la nueva posición de la máquina, encargándose además de comprobar que se mantiene en la línea del plan proyectado. Además, otro objeto de esta invención, mediante el uso del ordenador, consiste en proporcionar los errores de alineación y verificar cuándo es necesario cambiar la ubicación del equipo emisor de rayo láser, al comprobar que se encuentra fuera de rango.

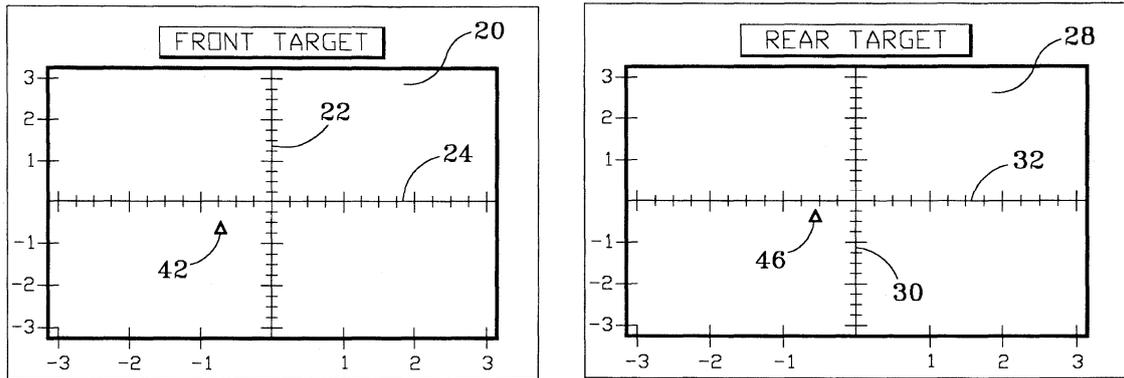


Figura 9.- Vista de los objetivos (Patente 5)

Por último, se muestran las gráficas generadas por el ordenador, estando disponibles tanto la actual como las anteriores calculadas, que servirán al operario para conocer las actuaciones que debe realizar sobre el control de la máquina, para mantener la dirección proyectada para el túnel.

2.7. 6ª Patente: Automated Guided Apparatus Suitable for Topping Applications

Número de Patente: 6.349.249

Autor: Peter D. Cunningham

Fecha: 20 de Febrero de 2000

Esta invención cuenta con un vehículo de guiado automático, capaz de determinar con precisión su posición dentro de un entorno de paredes, tales como una mina o un edificio. Cuenta con una unidad móvil, que incorpora una unidad de medición inercial y un sistema de procesamiento por visión en escala de grises, junto con una cámara y/o un puntero láser, siendo capaz de conocer su ubicación y actualizarla en el medio que se encuentre. El aparato está especialmente adaptado para la producción de túnel de vistas en planta y también para los equipos de orientación a través de estos entornos.

Debido a que el envío de equipos de trabajo de inspección a las excavaciones subterráneas puede ser peligroso y tedioso y, por otra parte, la contratación y utilización de un personal altamente cualificado contribuye al alto costo de la minería subterránea, hay un deseo permanente para automatizar los sistemas de minería. Las técnicas de minería robótica requieren de navegación

autónoma y sistemas de posicionamiento. Para la minería de superficie (así como otras actividades), el éxito cada vez mayor se ha encontrado con el uso de sistemas de posicionamiento global de navegación por satélite que utilizan en tierra. En las minas subterráneas y en las estructuras de cierta superficie, sin embargo, las señales de los satélites en órbita no pueden llegar al equipo, por lo que no se puede hacer uso de esta tecnología. En consecuencia, se utilizan técnicas típicas de sonido, medios electromagnéticos (láser, radar), giróscopos, o combinaciones de ellos de forma remota para operar con los equipos móviles.

El aparato incluye una unidad de medición inercial, la unidad central de procesamiento, telémetro láser, escáner láser, puntero láser, sistema de visión a escala de grises y una plataforma móvil. El aparato es capaz de inicializar su posición con precisión y fiabilidad de que atraviesa el lugar proyectado y, dado que recoge datos de topes para limitar su movimiento con precisión, la invención es especialmente útil para la realización de túneles y estudios de construcción similares.

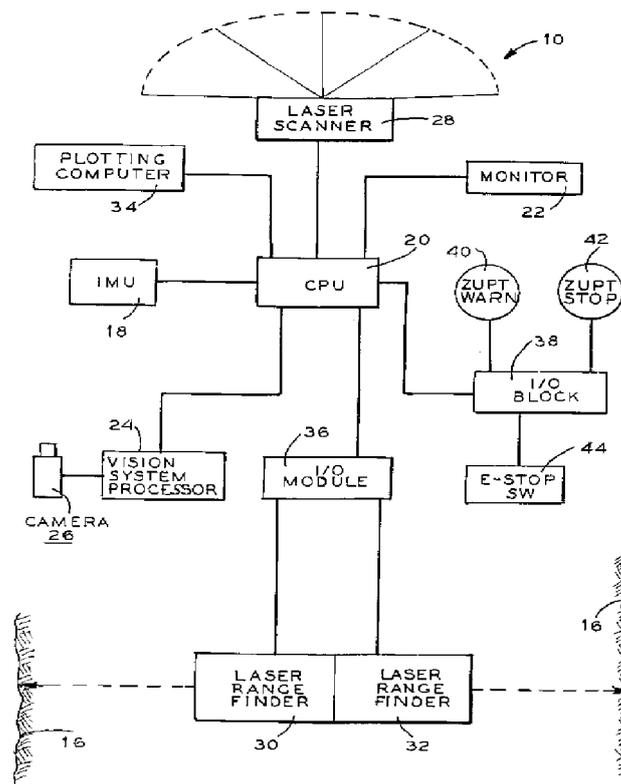


Figura 10.- Planta del sistema (Patente 6)

En el sistema propuesto, un giroscopio láser de anillo de haces opuestos se propagan en un recorrido triangular cerrado, respecto del eje de avance. Si el giroscopio está en reposo, los haces se anulan entre sí produciendo un estado nulo detectado por los sensores apropiados. Si se mueve el giroscopio, la oposición de los haces produce un cambio medible del diferencial de fase.

Utilizando un software para el giroscopio, los datos registrados sirven para medir el cambio en la posición en tres dimensiones, junto con el cambio en los ángulos de roll, pitch, yaw y azimut.

Dos sistemas de láser apuntan hacia los laterales del túnel, conteniendo un detector para el propio láser, que es reflejado sobre ellos al incidir sobre la pared, con que se registra la posición respecto al eje central del túnel.

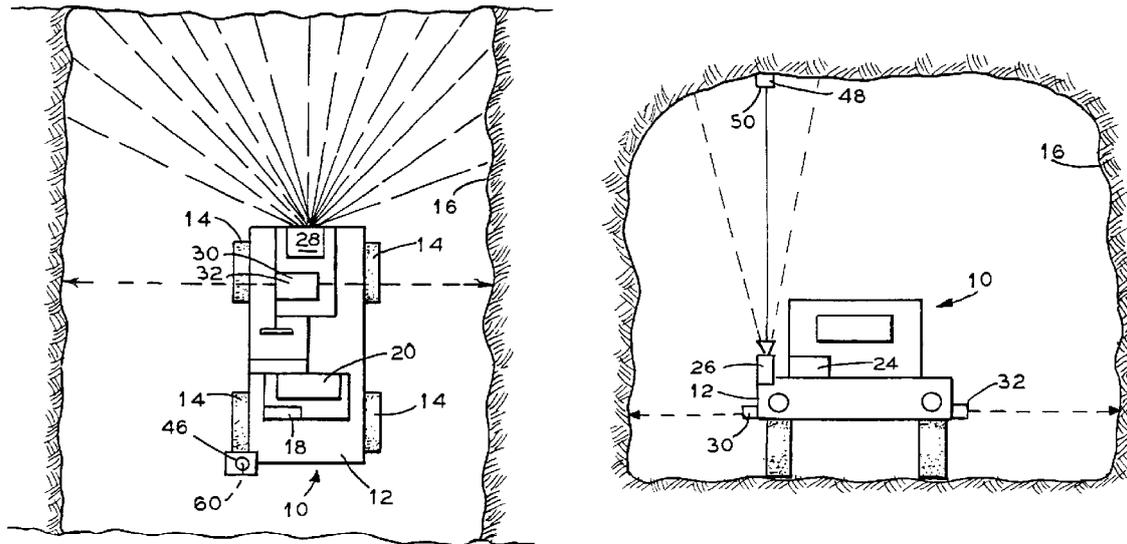


Figura 11.- Vista en planta y alzado de los sistemas (Patente 6)

Por último, se tiene un sistema de visión para conocer el avance del vehículo, compuesto por una cámara orientada hacia el techo del túnel, que obtiene fotografías de él. Para el conocimiento del avance es necesario colocar unos pernos en el techo del túnel, a una distancia conocida y equidistante. Por medio de un software de visión con capacidad a escala de grises de detectar bordes, se realizará la detección de los pernos, al ser distorsiones en la imagen sobre la superficie lisa del techo.

2.8. Conclusiones

Estos seis sistemas presentados tienen en común algunos aspectos con este proyecto, que son los siguientes:

- Uso de dianas o tarjetas para la detección de un láser para conocer las desviaciones de la máquina.

- Uso de cámaras para la detección de los puntos de incidencia del láser en los objetivos colocados para tal fin.
- Uso de un ordenador para computar tanto las desviaciones de la máquina como el avance de las mismas.

Sin embargo, todos presentan ciertas desventajas a la hora de su aplicación al guiado de tuneladoras, que se han querido resolver en este proyecto y son:

- Los láser usados son de color rojo, por lo que el alcance de los mismos es muy limitado, no sobrepasando los 100 metros, y teniendo en cuenta que en la actualidad se ha conseguido aumentar el avance diario de las máquinas tuneladoras, es el propio sistema de guiado el que limita en la productividad de perforación.
- Las señales eléctricas enviadas al ordenador de guiado son actualmente sustituidas por sistemas digitales en lugar de analógicas, permitiendo mayor precisión y sobretodo poder conocer si se producen errores en el envío de la señal, facilitando también la comprensión directa de las señales por el ordenador al no ser necesario añadir sistemas de instrumentación que traduzcan los valores.
- La evolución de las antiguas cámaras analógicas en favor de las actuales digitales, permite una mayor velocidad en el tratamiento de las imágenes obtenidas, consiguiendo extraer la información de las mismas en un intervalo de tiempo menor. Esto supone obtener un sistema de guiado con mayor refresco de datos y, por tanto, de mayor precisión.

3. Especificaciones del Diseño

3.1. Requisitos del Sistema

En general, una tuneladora es una máquina que permite perforar un túnel mediante un proceso continuo. En este proyecto se basará en el estudio para una tuneladora de tipo hincas de tubos. En este tipo de perforación, se utiliza el tubo como elemento definitivo del túnel y al mismo tiempo como elemento de empuje sobre la tuneladora. El avance se realiza gracias al empuje efectuado por un conjunto de cilindros de empuje instalados en el pozo de ataque sobre el tubo de hincas, el cual ha sido fabricado siguiendo unas normas estrictas, para poder soportar grandes esfuerzos longitudinales y transversales sin sufrir ningún deterioro. El tubo situado sobre el bastidor formará parte del túnel una vez concluida la hincas, cada tubo lleva instalada una junta en la boquilla, que debe garantizar la estanqueidad así como de una "sufridera" en la cola, que absorbe las posibles irregularidades del tubo y que permite que este no sufra al unir dos tubos y empujar hormigón con hormigón. En todas las hincas se construye un pozo de ataque con un muro de reacción que soportara las presiones de empuje de toda la tubería y un pozo de llegada donde se rescatara la maquina. Para hincas de gran longitud, se instalan unos elementos entre los tubos que se denominan "Estaciones Intermedias", y que permiten distribuir los esfuerzos entre varios tramos.

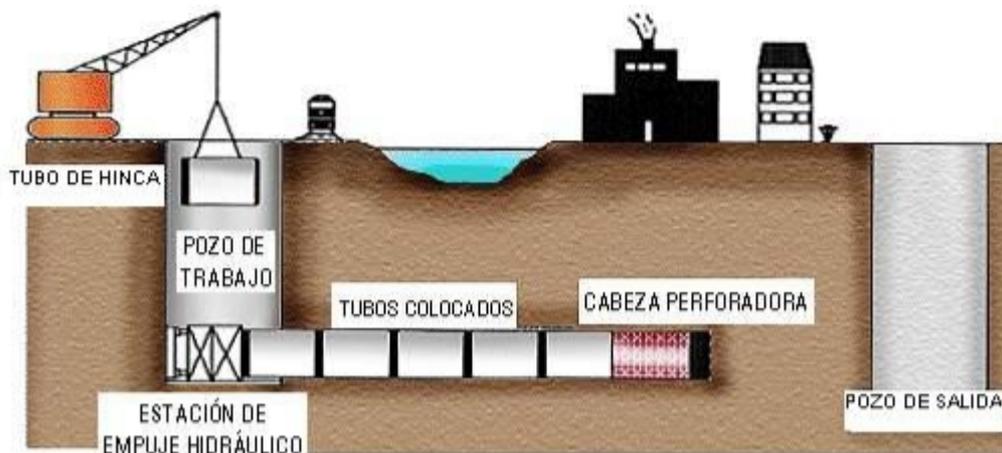


Figura 12.- Esquema de perforación para máquina de hincas de tubos

Hasta ahora, el guiado de una tuneladora, se realiza teleoperando desde una cabina de control, mediante la instrumentación de unos paneles de datos y mandos. El operario encargado del guiado, tiene la misión de regular el avance de la máquina dentro de unos límites óptimos prefijados.

Para realizar esta teleoperación, el operario necesita adquirir la experiencia suficiente para resolver cualquier situación que se produzca durante el uso de la máquina, adquiriendo una destreza en el manejo del control de los mandos,

además de obtener suficiente comprensión de los posibles problemas que puedan ocurrir en el transcurso de las operaciones de perforación. En estos problemas, tales como desprendimientos excesivos de tierra, atascos en la extracción del material o aumentos de presión entre otros, debe ser capaz de tomar las decisiones oportunas para solventarlos o, en el peor de los casos, minimizar los riesgos que puedan surgir para las personas y la maquinaria.

Los componentes esenciales que forman un sistema de guiado son:

- Un teodolito o estación láser de puntería.
- Una diana sobre la que incide el haz del láser, y que está colocada en el interior de la máquina.
- Inclinómetros o giróscopos capaces de medir los ángulos que no se puedan calcular a partir de la incidencia del láser.
- Un ordenador con software específico para el cálculo y procesamiento de los datos que definen la trayectoria de la máquina, a partir de los datos enviados por el resto de componentes, y que sea capaz de reproducirlos en pantalla, tanto de forma numérica como visual.

3.2. Especificaciones del Diseño

En vista de las necesidades que requiere un sistema de guiado para que sea completamente funcional y posea todas las características necesarias a la hora de trabajar en la realización de trabajos de perforación para la construcción de un túnel, mediante una máquina tuneladora, se ha previsto de la necesidad de desarrollar tres bloques diferenciados, pero a su vez conectados entre sí. Son los siguientes:

- Diana de puntería láser.
- Interfaz gráfica de usuario (GUI).
- Simulador de hincas de tubos.

La diana de puntería láser será la encargada de calcular y ofrecer los valores reales de la máquina tuneladora. El simulador hincas de tuberías tiene dos funciones primordiales: ser un sistema por el cual se puedan realizar las simulaciones de la obra antes de llevarla a cabo, para poder predecir las dificultades y errores que se tendrán a la hora de realizarla, así como buscar las soluciones en caso de que se den; y ser un sistema con el cual se pueda formar a nuevos operarios para el guiado de la máquina, con lo cual se evitaría los largos procesos de formación que actualmente se realizan en la propia obra bajo la supervisión de un operario cualificado y con experiencia. La interfaz

gráfica deberá ser capaz de registrar y mostrar en tiempo real los datos de ambos sistemas, con precisión y exactitud suficiente para poder guiar la máquina tuneladora.

Se ha elegido como modelo de referencia la máquina tuneladora para hincas de tubos AVND2000AB, ya que es el más veces se ha podido observar en obra en el proyecto "LCM". En la siguiente imagen se puede observar las dimensiones:

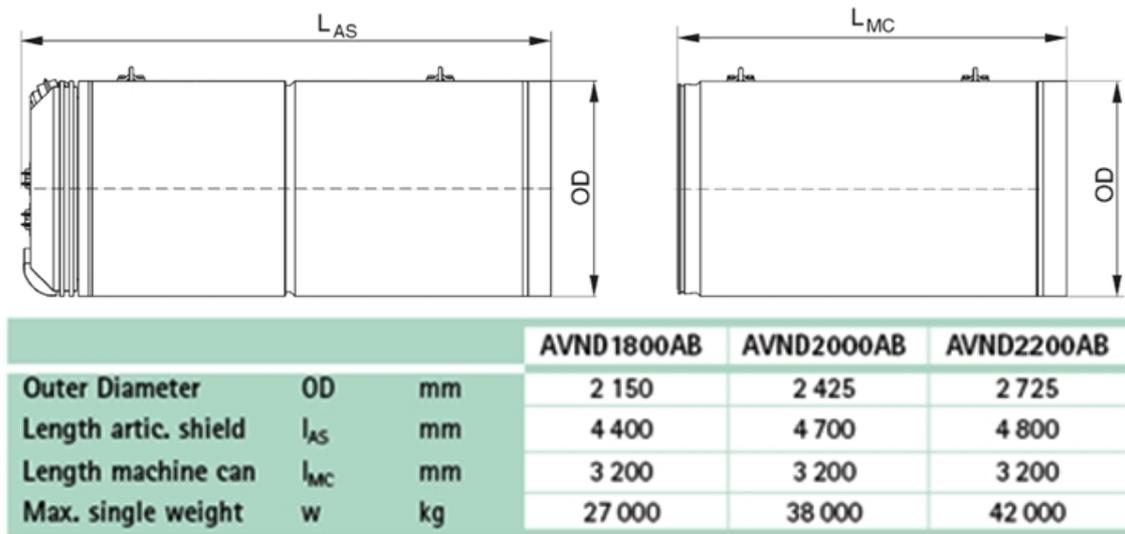


Tabla 1.- Dimensiones de la Máquina Tuneladora de Hincas de Tubos

3.2.1. Referencias del Sistema

En este apartado se detalla los sistemas de referencia que se han puesto en común entre todos los bloques que integran el sistema de guiado, para cumplir la característica comentada anteriormente de unificar criterios a la hora de obtener todos los parámetros necesarios.

El sistema de referencia global, común para todos los módulos, se ha emplazado a la entrada del túnel en el pozo de ataque, eligiendo como posición de origen el centro de la sección transversal del túnel. En cuanto a los ejes, se ha tomado el eje Z en la dirección de avance de la máquina, cumpliendo que junto al resto formen un sistema dextrógiro. Para los ángulos, se ha llegado al acuerdo de utilizar como referencia los ángulos de Euler, siendo el ángulo alfa el de "pitch" o cabeceo de la máquina, el ángulo beta el de "yaw" o guiñada, y gamma el de "roll" o alabeo. En la siguiente imagen se reproduce este sistema, para su mejor comprensión:

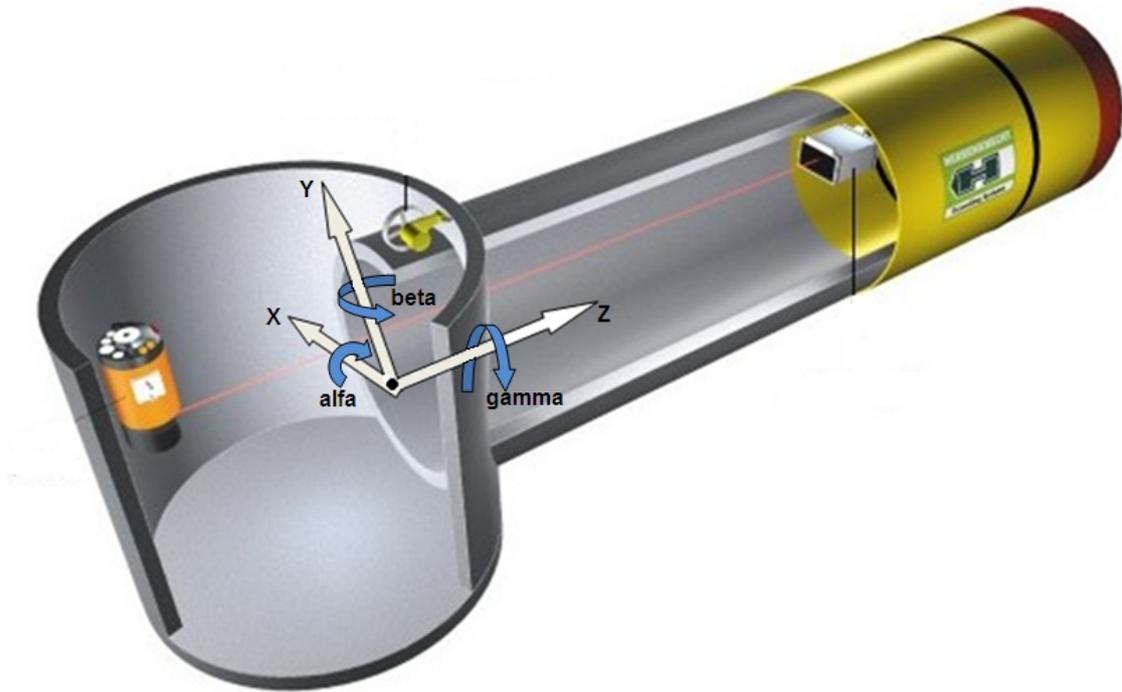


Figura 13.- Vista en perspectiva del sistema de referencia global y sus sentidos de giro

Para la diana, además, se llevan dos sistemas de referencia auxiliares, los cuales están ubicados en el centro de los dos paneles con los que cuenta, y con los ejes paralelos a los del sistema de referencia global. Con ellos se calculan los puntos de incidencia del láser para, posteriormente, calcular los ángulos y desviaciones realizados por la máquina tuneladora. Por último, se referencia al sistema global el punto ubicado bajo el panel trasero de la diana ortogonal al eje de simetría de la máquina, como punto de cálculo de la trayectoria.

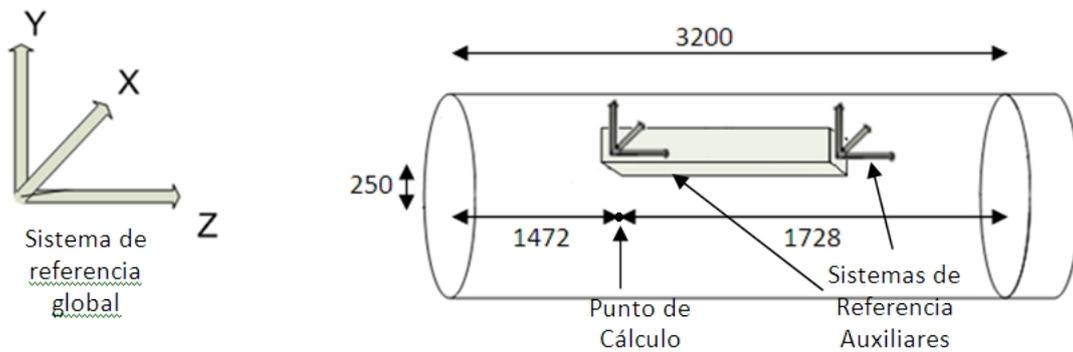


Figura 14.- Sistemas de referencia para la Diana

El simulador de hincas de tubos tiene una referencia al mismo punto de cálculo que la diana, y mantiene el sistema de referencia global, por lo que es equivalente a la imagen anterior.

Para finalizar, la interfaz gráfica de usuario GUI, transforma el punto de la trayectoria obtenido tanto por la diana como por el simulador, a los puntos trasero y delantero de la máquina, respecto del sistema de referencia global y apoyándose en un sistema de referencia móvil ubicado bajo la parte trasera de la diana, a la altura del eje de simetría de la máquina tuneladora.

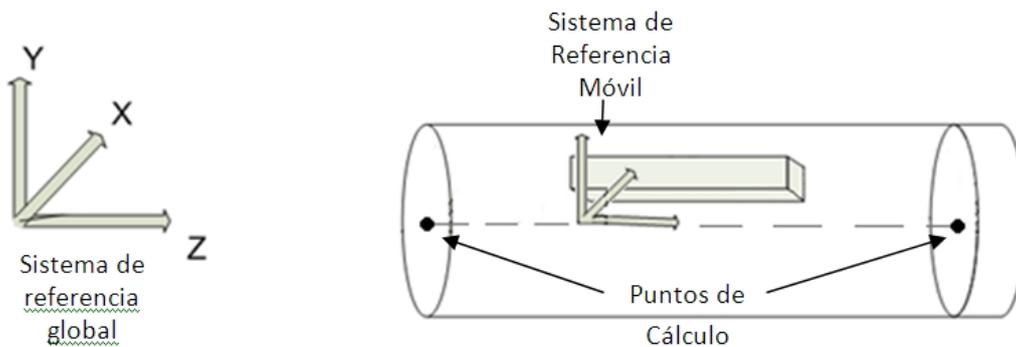


Figura 15.- Sistemas de referencia para la GUI

Sin embargo, aunque todos los cálculos para mostrar y registrar la información numérica se realizan como se ha indicado, no es así para el caso de los puntos en la representación visual, puesto que debido a las grandes dimensiones de la máquina tuneladora con respecto al pequeño avance de la misma, cada vez que se muestra una nueva representación, se verían solapadas las flechas que simbolizan la trayectoria de la tuneladora. Con el fin de resolver este problema, se ha realizado un escalado a la hora de representar la tuneladora gráficamente, según se muestra en la imagen a continuación:

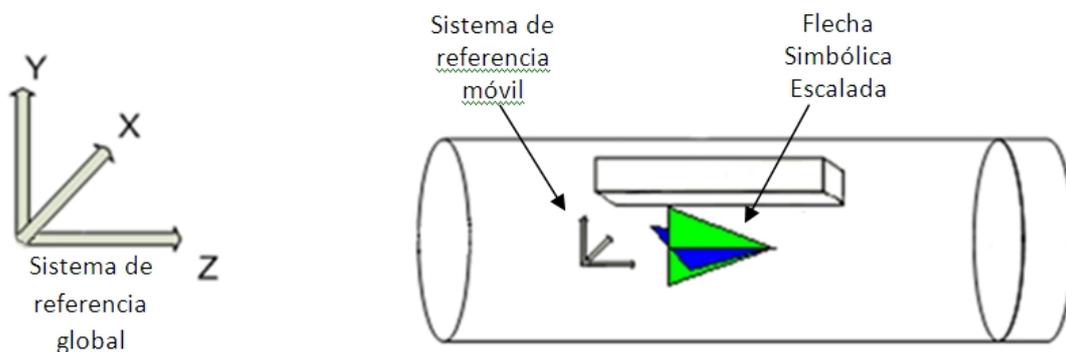


Figura 16.- Sistema escalado para la representación de la GUI

3.2.2. Procesado de información

En este apartado, se va a desarrollar la transición de flujo de la información de datos, procesados desde la obtención de los parámetros iniciales de referencia, hasta la visualización final de la posición espacial de la máquina tuneladora.

Como ya se ha comentado, para el sistema de guiado necesitamos tres bloques fundamentales, compuestos de una diana láser, un simulador y una interfaz gráfica externa (GUI). El sistema de guiado puede adquirir tanto datos reales procedentes de la diana, como datos procedentes del simulador. La interfaz gráfica debe ser capaz de procesar toda la información recibida por medio de la diana o del simulador para su respectiva monitorización.

Por este motivo, se establece un sistema de referencia global para todos los módulos que componen el sistema de guiado, con el fin de poder unificar un criterio común a la hora de especificar las coordenadas posicionales en las que se ubica la máquina tuneladora. Este sistema de referencia global está situado en la entrada del túnel y referenciado al eje de simetría de la máquina tuneladora.

La diana láser trabaja bajo dos sistemas de referencia auxiliares ubicados en el centro de ambos paneles de incidencia, donde recoge la posición de los puntos generados por el láser, a partir de la cual calcula los ángulos y las desviaciones producidas en la máquina para calcular la nueva situación espacial de la tuneladora. Estas coordenadas son transformadas al sistema de referencia global como salida del sistema hacia la interfaz gráfica.

El simulador genera tanto coordenadas de posición, como desviaciones angulares, a partir de la configuración de los parámetros del sistema empuje y cabezal de corte, que controlan la trayectoria ficticia que sigue la máquina tuneladora. El simulador ha adoptado el mismo punto de referencia que la diana, para obtener la trayectoria simulada con respecto al sistema de referencia global.

La interfaz gráfica recibe las coordenadas de la trayectoria, tanto reales como simuladas, y las transforma a las nuevas posiciones correspondientes de la parte delantera y trasera de la máquina tuneladora. Gracias a la unificación de criterios, tanto de los sistemas de referencia como del punto de cálculo por parte de la diana y el simulador, los cálculos realizados para la transformación de coordenadas son iguales para ambos módulos.

La monitorización paramétrica de los valores calculados para la obtención de las posiciones y desviaciones angulares de la parte delantera y trasera de la máquina tuneladora, se corresponden con las dimensiones reales de ésta.

Sin embargo, para la monitorización gráfica, tanto en dos como en tres dimensiones, se realiza un escalado reductor de las dimensiones totales de ésta, graficando una flecha simbólica que tipifica la máquina tuneladora, para conseguir representar un fragmento ampliado que permita visualizar con más precisión los cambios de trayectoria.

Del mismo modo, se realiza un escalamiento aumentativo de la desviación angular del eje de avance longitudinal, para visualizar exaltadamente los cambios angulares que resultan inapreciables.

A continuación, se va explicar detalladamente las variables con las que trabaja cada uno de los tres módulos y que comparten entre ellos.

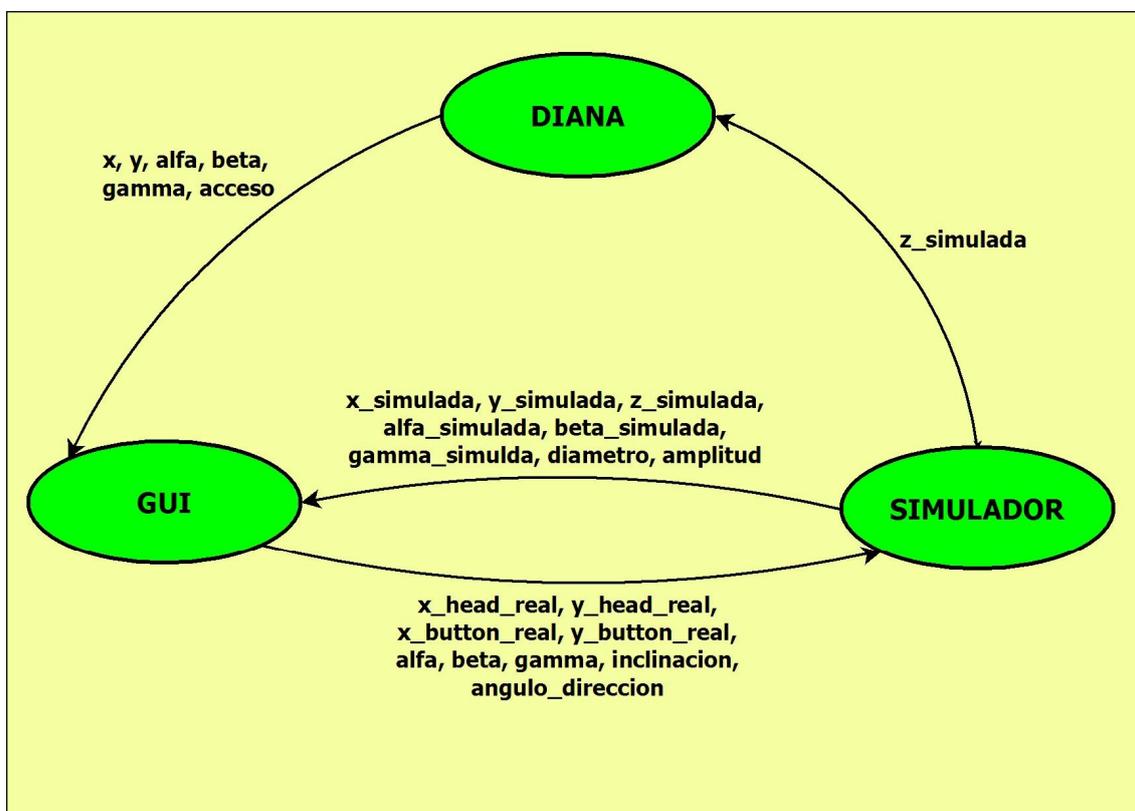


Figura 17.- Esquema de las variables usadas para cada bloque del sistema

En primer lugar, la diana láser calcula las variables respecto a los dos sistemas de referencia auxiliares, “x1”, “y1”, “x2” e “y2”, que son las desviaciones de los puntos producidos por la incidencia del láser en los paneles delantero y trasero de la diana. A partir de los mismos, se calculan las variables finales “x”, “y”, “alfa”, “beta” y “gamma” respecto al sistema de referencia global. La unidad elegida para las variables de posición es el milímetro, mientras que las variables angulares se calculan en radianes.

En segundo lugar, el simulador de hincas de tubos realiza el cálculo de variables rigiéndose en una librería dinámica, programada en C++, que obtiene las variables simuladas de posición “x_simulada”, “y_simulada”, “z_simulada”, angulares “alfa_simulada”, “beta_simulada”, “gamma_simulada”, y parámetros del láser “amplitud” y “diametro”. La unidad elegida para las variables de posición es el milímetro, mientras que las variables angulares se calculan en grados, y los parámetros del láser en tanto por ciento y milímetros respectivamente.

En tercer lugar, la interfaz gráfica calcula los puntos situados en la parte trasera y delantera de la máquina tuneladora, a partir tanto de los puntos reales, obtenidos por la diana, como de los ofrecidos por el simulador. Además, para realizar una comprobación, en el caso de estar usando el sistema de diana real y simulador al mismo tiempo, hará entrega de los valores reales calculados para los puntos de la parte trasera y delantera de la máquina al simulador.

Como variables de salida hacia los otros módulos por parte de la GUI, tenemos:

- Las variables correspondientes a los valores reales de la posición de la máquina: “x_head_real”, “y_head_real”, “x_button_real” e “y_button_real”.
- Las variables correspondientes al cálculo de tendencias que son: “inclinacion” y “angulo_direccion”.

Las unidades utilizadas tanto para el cálculo de parámetros reales como simulados son milímetros.

3.3. Requisitos e Innovaciones Funcionales del Proyecto

En vista de los objetivos marcados en el punto 1.2 de este proyecto, y pretendiendo cumplir las especificaciones señaladas en este mismo capítulo, en relación a las características que se buscan para mejorar y desarrollar un sistema de guiado, se ha llegado a la conclusión de que se necesita cumplir unos requisitos que se irán comentando a continuación.

Las actuales dianas láser son de reducidas dimensiones, debido a que los paneles de puntería, que son realizados mediante paneles de células fotosensibles, tienen un alto coste. Además, se basan en el uso de un láser de color rojo, cuyo rango de alcance no supera los cien metros.

Como solución a esto, se va a desarrollar un prototipo de diana de mayores dimensiones, que posibilite la obtención de la dirección del láser sin tener que realizar cambios tan frecuentes en su dirección, con el tiempo, dificultad y coste que esto supone. También, se cambiará el láser rojo por uno de color verde, con un alcance cercano a los 200 metros, lo cual va más acorde a los tiempos de perforación, en torno a los 30 metros diarios, para las máquinas tuneladoras actuales. Por último, y debido a la imposibilidad de conocer la posición de un láser verde con los paneles de células fotosensibles, se cambiará el método de detección por un sistema de visión, compuesto por dos cámaras, cada una de las cuales apuntará hacia un panel, siendo un sistema más económico que los actuales. El panel de primer alcance del láser estará compuesto por dos placas de metacrilato transparente con un mallado entre ambos, el cual permitirá el paso del láser hacia el siguiente panel, pero sin admitir una excesiva entrada de luz ambiental, que imposibilite la visión del punto láser generado en el panel. Por el contrario, el segundo panel será totalmente opaco, para beneficiar la generación del punto láser.

Debido a la colocación de la diana en el interior de la máquina tuneladora, el sistema de visión será incapaz de conocer el ángulo "roll" o alabeo, producido por el giro sobre el eje Z de avance de la máquina. Por tanto, en la diana se colocará un inclinómetro que permitirá conocer este ángulo, para tener totalmente definida la trayectoria de la tuneladora.

Como último requisito, se generarán las variables de salida, compuestas por las posiciones "X" e "Y" del punto de cálculo de la trayectoria, ubicado en el eje de simetría de la máquina a la altura de la parte trasera de la diana, con respecto al sistema de referencia global propuesto, además de los tres ángulos "alfa", "beta" y "gamma" producidos sobre los tres ejes del sistema. Estas variables serán calculadas en milímetros y radianes, y tendrán que ser comunicadas a una interfaz gráfica de usuario (GUI) para su posterior visualización.

4. Arquitectura de la Aplicación

4.1. Introducción

En este capítulo, se quiere realizar una visión de las herramientas usadas para la realización de este proyecto, además de realizar una explicación detallada de los programas elaborados.

Por una parte, se comentarán los aspectos y cualidades que han llevado a usar las distintas herramientas de hardware y software, buscando llevar al mejor fin posible este proyecto.

En cuanto al software, se comentan las características generales de los programas de desarrollo usados, así como se resaltan las necesidades que se buscaban para conseguirlo. Como adelanto, decir que se han usado los programas Matlab y MPLab IDE.

Del hardware se detallan las características principales que tiene cada componente, además de comentar los problemas que han surgido en su uso y como se han solventado. Dentro de este apartado se comentan las siguientes herramientas: Diana láser, láser, cámaras, filtro de línea láser e inclinómetro.

En la explicación de los programas realizados, se comenta cada función por separado, haciendo hincapié en los principales comandos, describiendo tanto su función, como los parámetros definibles e importantes en este proyecto.

Por último, se describe las conexiones y configuración necesarias para hacer funcionar correctamente todas las partes del proyecto, así como la interconexión que se realiza con otros programas desarrollados e implementados con éste.

4.2. Descripción de Sistema

4.2.1. Diana y Láser

La diana es el elemento donde incidirá el láser y permitirá realizar el cálculo de la trayectoria, por la cual quedará definida la posición de la máquina tuneladora. La diana de incidencia láser va colocada en el interior de la tuneladora, siendo el elemento móvil, mientras el láser mantiene su posición fija, a la entrada del túnel, con lo cual sirve como referencia para calcular las desviaciones a lo largo de la trayectoria.

La siguiente imagen muestra el prototipo de diana construido para la realización de este proyecto, así como las dimensiones exteriores del mismo en milímetros:

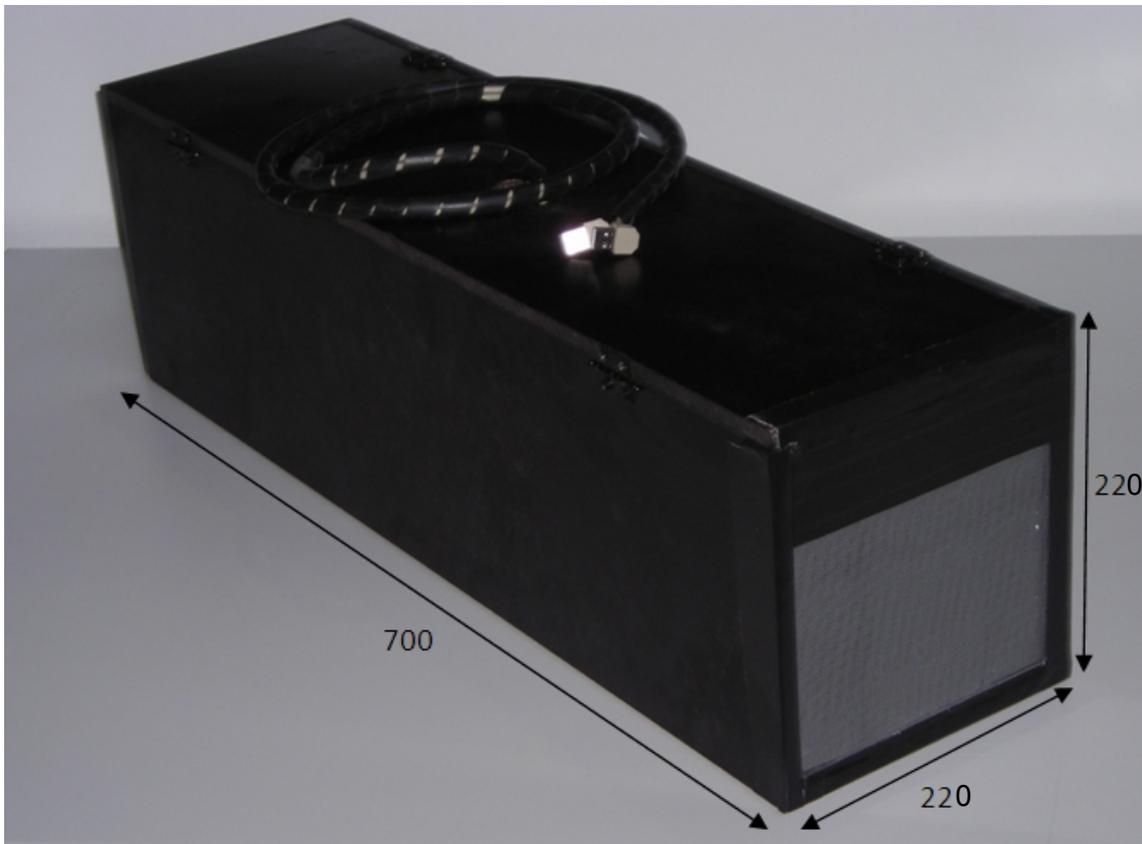


Figura 18.- Vista exterior de la Diana con dimensiones

Uno de los problemas que se han querido solventar, ha sido las reducidas dimensiones de las dianas actuales para láser rojo, los cuales constan de paneles fotosensibles, con un alto coste que implica que las dimensiones efectivas de las dianas con esta tecnología sean del tamaño menor posible. Ello, sin embargo, supone que las desviaciones para las que puede trabajar sean menores, al suponer que la máquina quedará fuera del rango del láser en muchas ocasiones, provocando la necesidad de tener que recolocar el láser en línea con la diana. Estas colocaciones que se producen durante el proceso de tunelación, suponen tener no sólo que recalcular la posición relativa del láser con respecto del origen de coordenadas con el que se esté trabajando, sino también la necesidad de parar la ejecución de la obra, y por tanto, de mayor tiempo en finalizar los trabajos.

La diana consta de dos paneles, uno traslúcido a la entrada de la misma, y un panel final opaco, en los cuales se producirán dos puntos en los que se refleja el láser. Las dimensiones útiles de los paneles son de 120 x 160 mm, lo cual supone en resolución de la cámara a 450 x 600 píxeles. Por ello, la resolución

que tenemos para detectar las posiciones es de 0,267 mm tanto para las distancias en horizontal como en vertical.

La distancia interior entre ventanas es de 680 mm, por lo que atendiendo al tamaño de paneles, se pueden obtener los ángulos de desviación máximos, para estas ventanas:

$$\text{Desplazamiento vertical máximo} = \arctg \frac{120}{680} = 10^\circ$$

$$\text{Desplazamiento horizontal máximo} = \arctg \frac{160}{680} = 13,24^\circ$$

En la siguiente figura se muestra las especificaciones de dos modelos de diana comerciales similares a las del prototipo desarrollado en este proyecto, que pueden ayudar como guía para cumplir las especificaciones buscadas.

6 Características técnicas de ELS1 y ELS2

Carcasa	ELS1	ELS2
Dimensiones (sin dispositivo de fijación)	330 x 135 x 80 mm	330 x 168 x 108 mm
Material	aleación de aluminio altamente resistente	
Peso	aprox. 5,2kg	aprox. 8,7kg
Modo de protección	IP68, impermeable al agua hasta 5 m de profundidad, llenado en seco con nitrógeno	
Posibilidad de fijación	agujeros roscados M6 en el beafidor frontal, mandril de guía en el dorso de la carcasa	suspensión mediante 3 pernos, perno trasero además con rosca M8
Medio ambiente		
Temperatura del almacén	-30 a +70 °C	
Temperatura de trabajo	-10 a +50 °C	
Carga de impacto	100 g	
Óptica		
Láser a utilizar	1 a 3 mW	
Longitud de onda	633 nm (láser HeNe) / 635nm (láser de diodo)	
Modo	TEM00	
Distancia del láser	depende del láser y del medio ambiente (contenido de polvo)	
Gamas de medida		
Desplazamiento horizontal:		
Gama	115 mm	148 mm
Precisión	1 mm	
Desplazamiento vertical:		
Gama	60 mm	88 mm
Precisión	1 mm	
Ángulos de dirección (eje de guiada):		
Gama	±5°	
Precisión	1 mrad (1σ) = 0'05"	
Ángulo de inclinación longitudinal:		
Gama	±15°	
Precisión	1 mrad (0,1 mrad estabilidad de corta duración) = 0'05"	
Ángulo de balanceo:		
Gama	±40° (±5°)	
Precisión	0,1° (0,05")	
Ciclo de medición:	cada 600 ms	
Calibrado/Mantenimiento		
Calibrado	en planta, datos de calibrado almacenados en el aparato	
Mantenimiento	Intervalo recomendado: 3000 horas de servicio	
Contador de las horas de servicio	asistente	

Tabla 2.- Características de las dianas comerciales

Como se puede comprobar a partir de los resultados, estamos mejorando la precisión en la detección de los desplazamientos multiplicándola por cuatro, al pasar de 1 mm a 0,267 mm, mientras que estamos prácticamente cumpliendo los rangos mínimos para el cálculo de giros, que son de $\pm 5^\circ$ en vertical y de $\pm 15^\circ$ en horizontal. Se han propuesto mejoras como trabajos futuros (ver apartado 6.2) en la realización de un nuevo prototipo de diana, que no solo las cumpla, si no que consiga superarlas.

El láser ha sido cedido por la empresa Eurohinca de la marca Topcon modelo "TP-L4G", siendo un láser verde con una longitud de onda de 532 nm (para conocer las características detalladas, ver anexo D).



Figura 19.- Vista del Láser con descripción de las funciones

4.2.2. Herramientas de Visión. Cámaras y láser.

Existen distintos parámetros a la hora de la elección de una cámara que cumpla los requisitos de una aplicación en concreto. Se puede destacar las siguientes características, a la hora de elegir una cámara:

- **Tipo de sensor de la cámara:** Hay dos tipos principales de sensores, CCD y CMOS. El sensor CCD se define principalmente por obtener más calidad de imagen a costa de un tamaño mayor y consumo más elevado. Por su lado CMOS es superior en integración y bajo consumo a costa de perder calidad de imagen en situaciones de poca luz. Se comparan sus características de una forma más detallada a continuación:

- **Responsividad:** Se define como el nivel de señal que ofrece el sensor por unidad de energía óptica incidente. Lógicamente interesa que el sensor tenga una responsividad elevada, o sea que con poca luz nos dé una señal aceptable. En este caso los CMOS son superiores a los CCD, debido a la mayor sencillez de integrar elementos amplificadores en un CMOS. Además por construcción CMOS permite alta amplificación con bajo consumo, mientras que en CCD la amplificación supone un consumo más elevado.
- **Rango Dinámico:** Es el cociente entre el nivel de saturación de los píxeles y el umbral por debajo del cual no captan señal. En este aspecto CCD supera a CMOS, ya que típicamente el rango dinámico de un sensor CCD es del orden del doble que un CMOS.
- **Ruido:** Los CCD aventajan a los CMOS en términos de ruido, dado que por construcción todo el procesado de señal se da fuera del CCD, con lo que se podrá aprovechar de convertidores A/D más rápidos o mejores procesadores, por su lado los CMOS al realizar la gran mayoría de las funciones dentro del sensor (Amplificación, Conversión A/D) pierden espacio para el fotodiodo en sí, y pagan esto en términos de ruido.
- **Respuesta Uniforme:** Se espera que un píxel, ante el mismo nivel de excitación de luz, no presente cambios apreciables respecto a sus vecinos, en este aspecto la individualidad de cada píxel en CMOS lo hace más sensible a sufrir fallos, siendo mayor la uniformidad en CCD. No obstante mediante circuitos con realimentación se ha conseguido paliar este problema en los CMOS, siguiendo los CCD con una ligera ventaja.
- **Velocidad:** La velocidad en CMOS es bastante superior a los CCD, debido a que muchas funciones, como la propia conversión analógico-digital son realizadas en el propio sensor. Por ahora esta ventaja es ligera pero se espera que aumente con el tiempo.
- **Enventanado:** Es sabido que los CCD funcionan a su máxima resolución en la gran mayoría de dispositivos, en cambio los CMOS dado su carácter individualizado de cada píxel, permiten seleccionar más

fácilmente porciones de la imagen, esto permite mas refresco en modos de video o bien tomar imágenes a menor resolución sin interpolación.

- **Blooming:** Es el conocido fenómeno por el cual un píxel que ha alcanzado la saturación empieza a 'contagiar' a sus vecinos, creando efectos indeseados. Esto se da en CCD, necesitando trucos de construcción cuando se quiere evitar, en cambio por su construcción los CMOS no sufre este defecto.

- **Resolución:** Se mide por la cantidad de puntos o pixeles que componen la imagen, tanto en vertical como horizontal. Siendo a mayor cantidad de píxeles, mayor resolución y, por tanto, mayor cantidad de información en la imagen. También se debe tener en cuenta que a mayor resolución, aumenta el coste computacional para el procesado de imágenes.

- **Tipo de lente u objetivo:** La lente de la cámara puede ser fija o intercambiable. Una cámara que permita el uso de diferentes medidas de lente, permite adecuarse a mayor cantidad de situaciones, que van desde un gran angular, con una distancia hasta la imagen muy pequeña, hasta un teleobjetivo, donde se busca en detalle objetos muy alejados de la cámara.

- **Autoenfoco:** En la actualidad, muchas cámaras incluyen un control automático tanto para la medida del enfoque, como para el balance de blancos de la imagen, permitiendo obtener imágenes con características adecuadas, sin necesitar de un ajuste manual.

- **Lux o mínima cantidad de luz:** Es la mínima cantidad de luz que las cámaras necesitan para poder funcionar. Es un parámetro que indica la sensibilidad de la cámara. Asimismo, existen cámaras infrarrojos que operan con 0 lux, y que nos permiten ver en total oscuridad.

- **Alimentación:** La alimentación de la cámara puede ser externa, comúnmente de 12V, o bien, alimentado por el propio cable de conexión de datos al ordenador, como es el caso de muchas de las actuales cámaras por conexión USB.

El sistema de visión comprende el uso de dos cámaras según se ha comentado en la descripción del sistema para la detección de los puntos generados en la diana por el láser. Las cámaras van colocadas en el interior de la diana en uno de los paneles laterales, enfocadas una hacia la ventana y otra hacia el fondo de la diana, como se puede observar en la siguiente imagen:

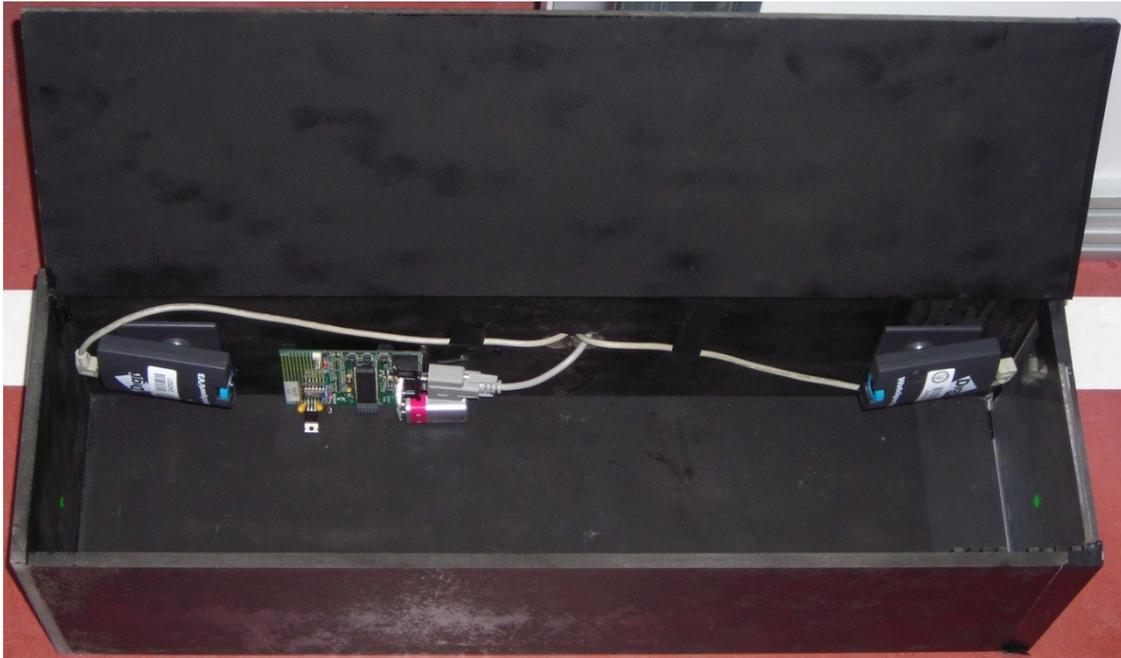


Figura 20.- Vista interior de la Diana

Esto limita el área útil dentro del interior de la diana para el láser, puesto que se debe evitar que el láser apunte hacia ellas y se impida la generación del punto en el fondo de la diana.

Las cámaras usadas para tal fin son de la empresa Digi, modelo "Watchport V3" con las siguientes características principales (ver Anexo C, para más información):

- Sensor CCD de resolución 320x240 sin interpolar.
- Distancia focal F:2.0 ajustable.
- Lente de 8mm.

Uno de los principales problemas a la hora de obtener una posición óptima para las cámaras, es el uso de un filtro de línea láser, el cual necesita estar lo más centrado en la dirección del láser para su detección, sin que se llegue a producir una pérdida de la visión, puesto que debido a su ancho de banda de 2nm, llegaría a filtrar el láser en la imagen para ángulos de incidencia muy grandes, al cambiar su longitud de onda. La modificación de la longitud de onda respecto al ángulo de incidencia se puede calcular mediante la ley de Snell, como se muestra en la fórmula siguiente:

$$n = \frac{c}{v} = \frac{\lambda_0 \cdot f}{\lambda \cdot f} = \frac{\lambda_0}{\lambda}$$

Por lo tanto, se debe buscar la mayor distancia posible entre la cámara y el plano de la imagen a obtener, para minimizar el ángulo de incidencia entre el láser y la cámara, con el fin de evitar la pérdida de visión para cualquier posible ángulo de trabajo.

Con la colocación actual de las cámaras, se aprovecha toda la longitud de la diana, al estar en los extremos opuestos de la imagen a obtener. Además, al estar las dos cámaras en el mismo panel de la diana, siendo en este caso el panel superior interior, sólo perdemos como área de visión hasta la altura de las cámaras, unos 6 cm, siendo la forma alargada y baja de ellas una de las características tenidas en cuenta a la hora de elegir las, así como la posibilidad de cambiar de lente.

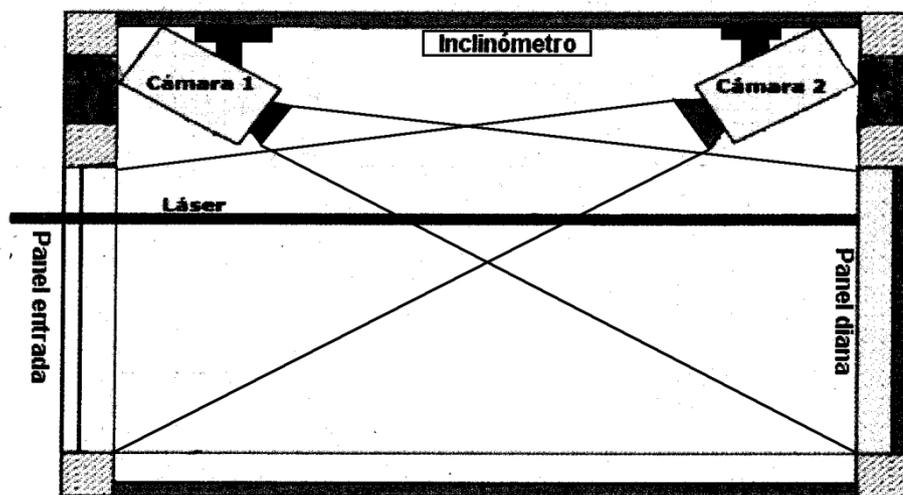


Figura 21.- Esquema interior de la colocación de las cámaras e inclinómetro

La lente de 8mm ha sido elegida debido a que se aprovecha un área útil de imagen máxima que pertenece tanto a la ventana como al fondo de la diana. Para otras ópticas probadas, como de 6 mm, el área útil se ve reducida en aproximadamente un 30% respecto a la de 8 mm, por lo que no es una opción a tener en cuenta, ya que para tener una imagen similar a la actual, sería necesario acercar las cámaras a sus planos de imagen unos 15 cm más, pero las pruebas realizadas determinan que con el uso del filtro, se pierde la visión del láser en el panel de la ventana de la diana, para ángulos de incidencia altos o en caso de que el láser se aproxime a los bordes de la imagen.

El filtro elegido es de la compañía Edmund Optics modelo "High Performance Laser-Line Filter 12.5mm Dia", con las siguientes características:

- Longitud de onda de 532 nm.
- Ancho de banda de 2 nm.
- Bloqueo de longitud de onda de 447-527nm & 537-699nm.

El filtro, colocado únicamente en la cámara que obtiene la imagen de la

ventana de la diana, nos resuelve el problema de obtener una imagen con excesiva luz exterior, que puede deberse a la luz natural, para una posición de la tuneladora muy cercana al pozo de ataque en el inicio de la obra, o a labores de inspección de los operarios en la tuneladora, haciendo uso de luz artificial. Gracias a él, la intensidad luminosa de la luz será muy baja en comparación a la intensidad del haz del láser, por lo que un proceso de binarización en la imagen, permitirá obtener su localización sin mucha dificultad.

Otras opciones de filtros que se han tenido en cuenta son los que tienen mayor ancho de banda, de hasta 10 nm, lo que en principio reduciría problemas a la hora de perder por filtrado la visión del láser con ángulos de incidencia mayores. Pero aunque mejorara éste problema, aumentaría la dificultad a la hora de identificar el punto generado por el láser con una alta luminosidad exterior.

Como se ha comentado anteriormente, el filtro está colocado exclusivamente en la cámara que recoge la imagen en la ventana de la diana. Ello es debido a que en el fondo de la diana la luminosidad es menor en gran medida, en parte por la característica de usar un color negro para la construcción de la diana, lo que impide que se produzcan reflejos dentro de ella, obteniendo una imagen de fondo prácticamente negro incluso para situaciones de luminosidad exterior muy altas.

La siguiente tabla muestra las características que definen este filtro de línea láser:

Dimensions	12.5mm Dia. and 25.0mm Dia. +0.0/-0.1mm
Thickness	3.5 ± 0.1mm
Full Width Half Maximum, Typical	0.38% of Laser Wavelength
Full Width Half Maximum, Max.	0.7% of Laser Wavelength
Transmission	>90% (>85% for 363.8nm, >80% for 325nm, >55% for 266nm)
Blocking	OD > 5.0 Range except Laser Wavelength ±1.0% OD > 6.0 Range except Laser Wavelength ±1.5%
Angle of Incidence	0.0° ± 2.0°
Substrate	Low Fluorescence N-BK7 or equivalent
Beam Deviation	<11 arcsec
Transmitted Wavefront	1/4 λ RMS @ 633nm
Surface Quality	60-40
Coating	"Hard" Ion-beam sputtered
Temperature Dependence	<5ppm/°C
Environmental Durability	MIL-STD-810F
Physical Durability	MIL-C-48497A

Tabla 3.- Características generales del filtro de la cámara

4.2.3. Inclinómetro

El inclinómetro es el componente con el cual podemos conocer el ángulo de “roll” o alabeo de la máquina tuneladora, producido por el giro sobre el eje Z en el sentido de avance de la máquina.

En la siguiente imagen, se puede apreciar la placa de desarrollo usada para el montaje del inclinómetro, el propio inclinómetro, el microcontrolador que incluye la placa, el regulador lineal incorporado para permitir la correcta alimentación del inclinómetro, la pila de alimentación de la placa y la conexión por puerto serie para la entrada/salida de datos.

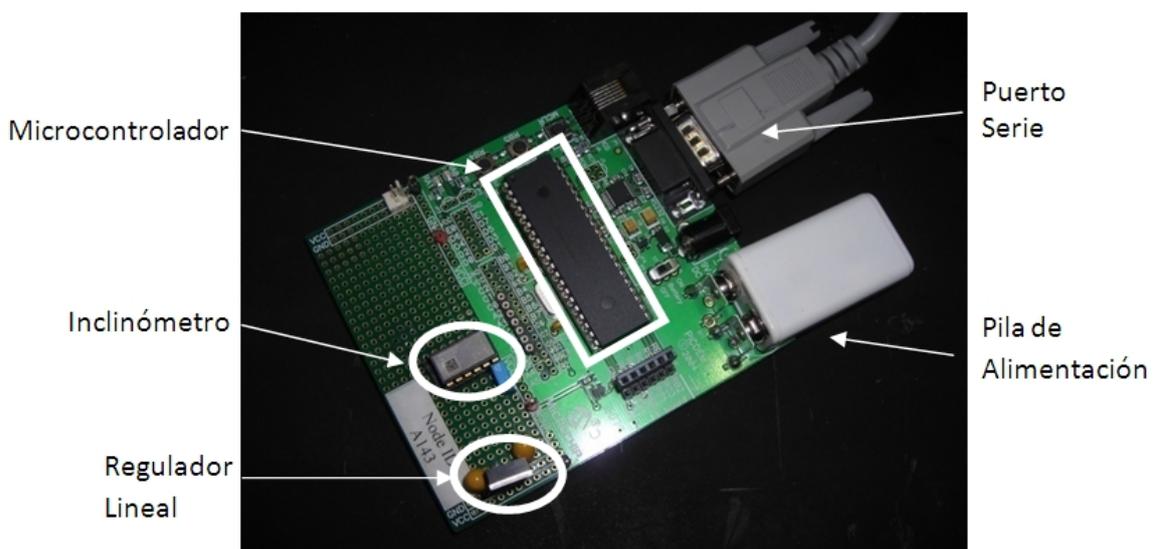


Figura 22.- Vista del inclinómetro montado en la placa de desarrollo

El chip elegido para conseguir tal fin, es de la marca VTI Technologies, modelo “SCA 100T-D01”, con las siguientes características principales (ver anexo E para más información):

- Inclinómetro de 2 ejes.
- Rango de medida de ángulos de ± 30 grados.
- Diseño robusto y resistente a los golpes (20.000g).
- Alimentación a 5V DC.
- Posibilidad de salida de datos analógica por tensión, o digital por conexión SPI.

Como indican las características, se puede optar entre calcular el ángulo de salida por medio de una relación entre tensiones usando la salida analógica del chip, u obtener la inclinación de forma digital, gracias a la conexión por SPI con la que cuenta. Para este proyecto, se ha querido llevar a cabo la última opción, aprovechando la salida digital.

Para comprobar si realmente este inclinómetro cuenta con las características necesarias para servir en la medida del ángulo, hay que fijarse en dos aspectos. El primero, es tener un rango de medida suficiente, lo cual no conlleva mayor problema en este caso, ya que el ángulo de alabeo permitido no excede en más de ± 3 grados. La otra característica a cumplir sería la resolución mínima detectable en un giro, que a partir de la fórmula de cálculo para el ángulo obtenido es de:

$$\gamma = \arcsin\left(\frac{1 - 1024}{1638}\right) = 0,016^\circ$$

Resultando que se ha incrementado esta precisión, al ser de $0,05^\circ$ en los modelos de dianas actuales.

Para el montaje del inclinómetro, se optó por realizarlo en una placa de desarrollo de la empresa Microchip modelo "Picdem Z", el cual integra un microcontrolador "18LF4620" (ver anexos G y F respectivamente), conexión a partir de un puerto serie y la posibilidad de añadir dispositivos realizando la conexión SPI digital, lo que cumple con las necesidades del proyecto. Para la alimentación se dan dos opciones, alimentarlo por medio de un transformador de 9V DC o mediante una pila de 9 V. Lleva implementado un regulador lineal de 9V a 3,3 V para permitir la correcta alimentación del microcontrolador, sin embargo, no es suficiente para el inclinómetro al necesitar 5V DC si se quiere hacer uso de la conexión digital SPI. Para solucionar este problema, se ha incorporado a la placa un regulador lineal "7805" con salida a 5V para obtener esta tensión directamente de la pila.

4.2.4. Conexión del Sistema

La realización de la conexión del sistema se basa en la conexión de las dos cámaras y el inclinómetro, al ordenador. Esta conexión debe ser realizada previa a la ejecución del programa, para realizar la identificación de los puertos de cada dispositivo en las sentencias de inicialización.

Uno de los problemas que pueden surgir en la actualidad es no tener disponible una conexión para puerto serie en el ordenador. Para resolverlo, hay disponibles unos cables adaptadores que, con sólo la instalación de unos drivers, simula a un puerto USB como serie.

Por otro lado, se debe realizar la instalación de los drivers de las cámaras la primera vez que se conecten al ordenador, eligiendo los drivers "Direct Show" en el CD que acompaña a las cámaras, frente a los "TWAIN", más orientados a aplicaciones del propio fabricante. Además, para comprobar el correcto

funcionamiento de las cámaras, el fabricante acompaña a los drivers con un software llamado “Watchportviewer”.

4.2.5. Interconexión con otros programas

Este proyecto, que comprende el desarrollo de hardware y software de una diana real de puntería láser, es una parte de la realizada para la obtención de un sistema total de guiado y simulación, que incluye a la propia diana real, una GUI de guiado para la visualización tanto de datos reales como de simulaciones y un simulador de hinca de tuberías.

Este conjunto se ha concebido con un sistema que permita el desarrollo de todas las situaciones que se vayan a producir durante el manejo de una máquina tuneladora. Con el simulador se pretende poder dar una idea clara de los problemas o situaciones que pueden surgir durante la realización de la obra de tunelación, además de servir como herramienta de aprendizaje para los distintos operarios que forman parte del control y guiado en la obra. Con la diana real se intenta conseguir un sistema que obtenga todos los datos necesarios para conocer la posición de la tuneladora en cada instante, durante el desarrollo de una obra real. Por último, la GUI facilita de una forma visual, tanto en una situación real trabajando con la diana, como en una simulación por medio del simulador, conocer la posición de la tuneladora, así como de los datos numéricos obtenidos y necesarios para que un operario conozca el estado actual y futuro de la máquina.

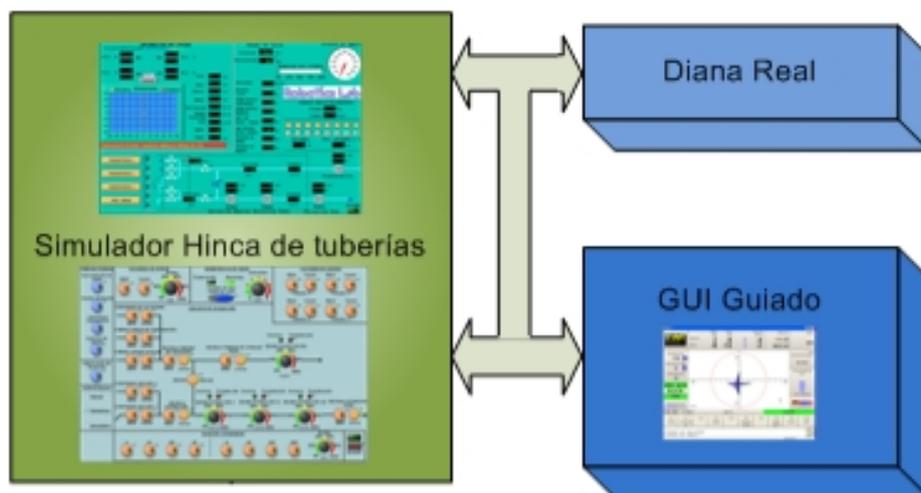


Figura 23.- Esquema de sistema completo de guiado

Debido a ello, el sistema necesita de la comunicación entre sus partes, para poder hacer uso del mismo y, por esa razón, se ha desarrollado una

comunicación entre los programas donde se comparte la información que necesite cada aplicación en cualquier momento. En vista de la necesidad de cada aplicación de obtener valores de otros, se ha llegado a la conclusión de compartir las variables que se recogen en el siguiente gráfico:

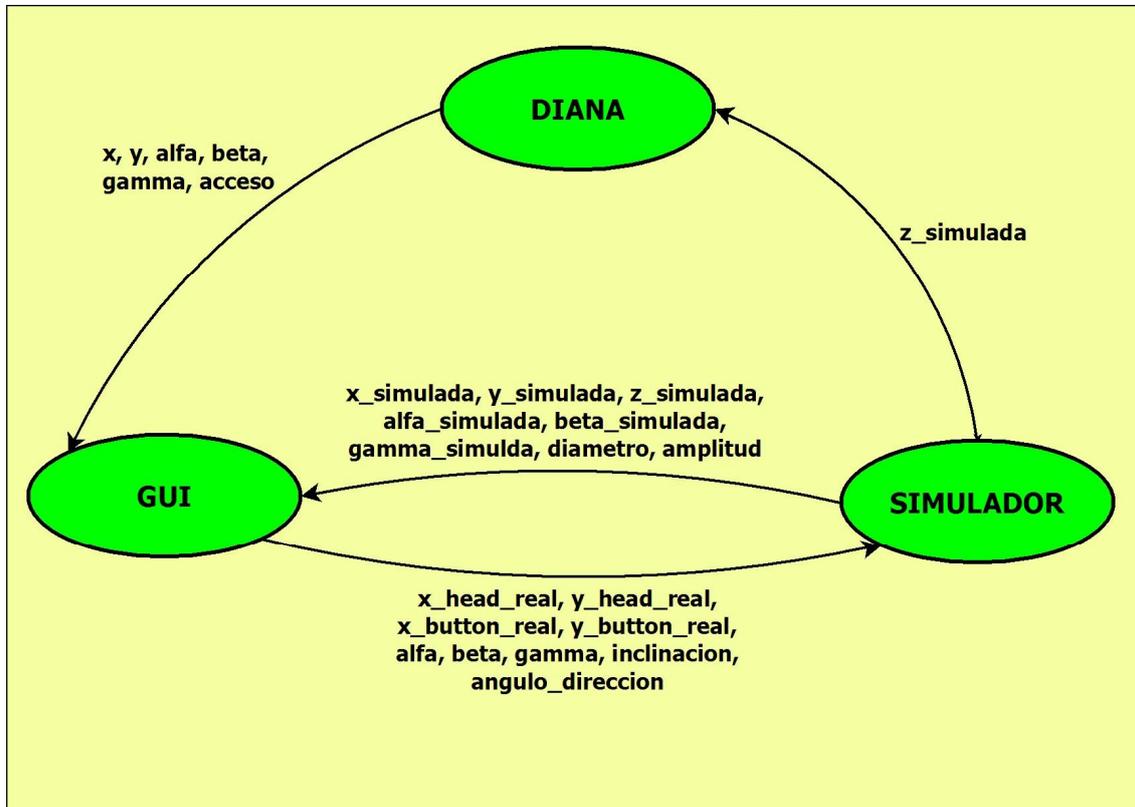


Figura 24.- Diagrama de intercambio de variables para los tres sistemas de guiado

Tanto la diana como la GUI han sido desarrollados mediante el programa Matlab, por lo que se han barajado distintas formas de comunicación comunes de este software. El simulador está siendo desarrollado en el programa Labview, lo que supone mayores problemas a la hora de obtener una forma rápida y fácil de comunicación.

Los métodos considerados para conseguir una comunicación han sido:

- Variables globales entre programas.
- Realización de un archivo de intercambio de datos.
- Comunicación por Socket.
- Realización de una librería de Enlace Dinámico (DLL).

A continuación, se detallan los métodos de comunicación y los aspectos que se han tenido en cuenta para elegirlo.

1. Variables globales

El método más directo para compartir valores entre las variables de los distintos programas, es mediante el uso de variables globales, las cuales serían definidas anteponiendo “global” al nombre de la variable en la definición de la misma, y usando el mismo nombre en todos los programas. Sin embargo, son varios los problemas que han llevado a desestimar el uso de esta forma de comunicación.

Al tener todos los programas el mismo nombre de la variable, se debe tener especial atención a la hora de modificar su valor, puesto que podrían hacer uso de ella al mismo tiempo y concurriría en errores en el cálculo difícilmente detectables. Además, al ser varios los programas comunicados, pero independientes en su ejecución y uso, el método de trabajo de Matlab para asignar las direcciones de memoria para las variables hace que no estén disponibles de un programa a otro, aun declarando la variable como global y con el mismo nombre. Esta falta de comunicación se produce al asignar Matlab un espacio de trabajo distinto para cada programa, aun corriendo en la misma máquina y al mismo tiempo.

Además, hay que discriminar entre los archivos Matlab definidos como función y como script. La diferencia fundamental entre ambos es que mientras en los script se trabaja con variables globales, ubicadas en un espacio de trabajo conjunto para ellos, las funciones tienen por defecto variables locales, las cuales no comparten con otras funciones ni aun declarándolas con el mismo nombre. En caso de obligar en una función a tener una variable de tipo global, si se puede tener acceso a la variable desde las distintas funciones, pero se concurriría en un error al llevarlo a cabo, puesto que habría que poner especial interés a la hora de hacer un mismo uso de nombres para las variables, pudiendo sobrescribir las de otros programas.

Las pruebas de distintos programas en los que se creó el archivo ejecutable “.exe”, para permitir tener varios programas a la vez, puesto que Matlab solo permite la ejecución uno, creándolos tanto como script como funciones y con las variables a compartir definidas como globales, determinó que ni aun así los programas compartían los espacios de trabajo en los que eran guardadas las variables de cada programa, lo que hizo imposible realizar la comunicación por este método. Además, ya se desestimó anteriormente, por los problemas que pueden surgir por un uso excesivo de las variables globales, comentados antes.

2. Archivo de intercambio de datos

Este método de comunicación, finalmente elegido para realizar la comunicación, se basa en la creación y uso de un archivo que permita leer y

escribir tanto el nombre de la variable como su valor, siendo accesible en cualquier momento por cualquier programa.

No presenta el problema de tener varios programas en diferentes carpetas de trabajo, puesto que se puede hacer la llamada al archivo indicando previamente la ruta o path en la que está ubicado, por lo que beneficia en la independencia de cada programa.

Además, Matlab cuenta con un tipo de archivo, de extensión “.mat”, el cual no sólo tiene comandos específicos para su escritura y lectura, sino que además lo hace manteniendo el formato que tengan las variables a la hora de realizarlo, lo que beneficia enormemente al no tener que especificarlo junto con el valor, ni tener que dar otro formato al leerlo. Estas ventajas no se obtienen en otros tipos de archivos comúnmente utilizados, como los archivos de texto plano “.txt” o de tabla “.xls”, por lo que ha sido el adoptado para realizar la comunicación.

El comando “save” es el que nos permite guardar el valor de todas las variables usadas por el programa, pero también nos permite hacer una selección de las que queramos que sean guardadas, tan sólo especificando el nombre, creándolas en el archivo en caso de que no lo haya sido hecho anteriormente o sobrescribiendo el valor. Además, cabe destacar el uso de la instrucción “-append” en este comando, que evita que se borren otras variables guardadas en el archivo de intercambio, en caso de que no se les dé valor en ese momento, ya que no todos los programas harán uso de las mismas variables del archivo.

```
save datos.mat x y alfa beta gamma acceso -append
```

El comando “load” lee el valor de las variables y las guarda en la variable con el mismo nombre en caso de que ya esté siendo usada por el programa, o las crea si no han sido declaradas. Al igual que el comando “save”, nos permite hacer una lectura selectiva de las variables que queramos, sin la necesidad de tener que cargar todas las variables, ocupando espacio en memoria innecesario, y evitando posibles pérdidas de valores.

```
load datos.mat z_simulada
```

Una dificultad añadida a la hora de compartir un mismo archivo para guardar las variables de intercambio, es la necesidad de regular el tráfico de los programas al acceder por medio de lectura o escritura al archivo. En el caso de la comunicación Diana – GUI, se ha tomado una variable común llamada “acceso”, la cual es puesta por la diana a valor lógico “1” cuando se realiza algún nuevo cálculo de la situación de la tuneladora y, al ser leída por la GUI le indicará que puede realizar la representación de una nueva posición de la tuneladora.

En cuanto a la comunicación Matlab – LabView, que en principio puede parecer imposible por no poder usar este tipo de archivos en LabView, se ha podido llevar a cabo gracias a la inclusión en la últimas versiones de labView del bloque de programación llamado “Matlab Script Node”, con el cual se puede insertar directamente código fuente de Matlab siendo reconocido y ejecutado, por lo que el método llevado en LabView es exactamente igual que en los programas de la Diana y GUI realizados en Matlab.

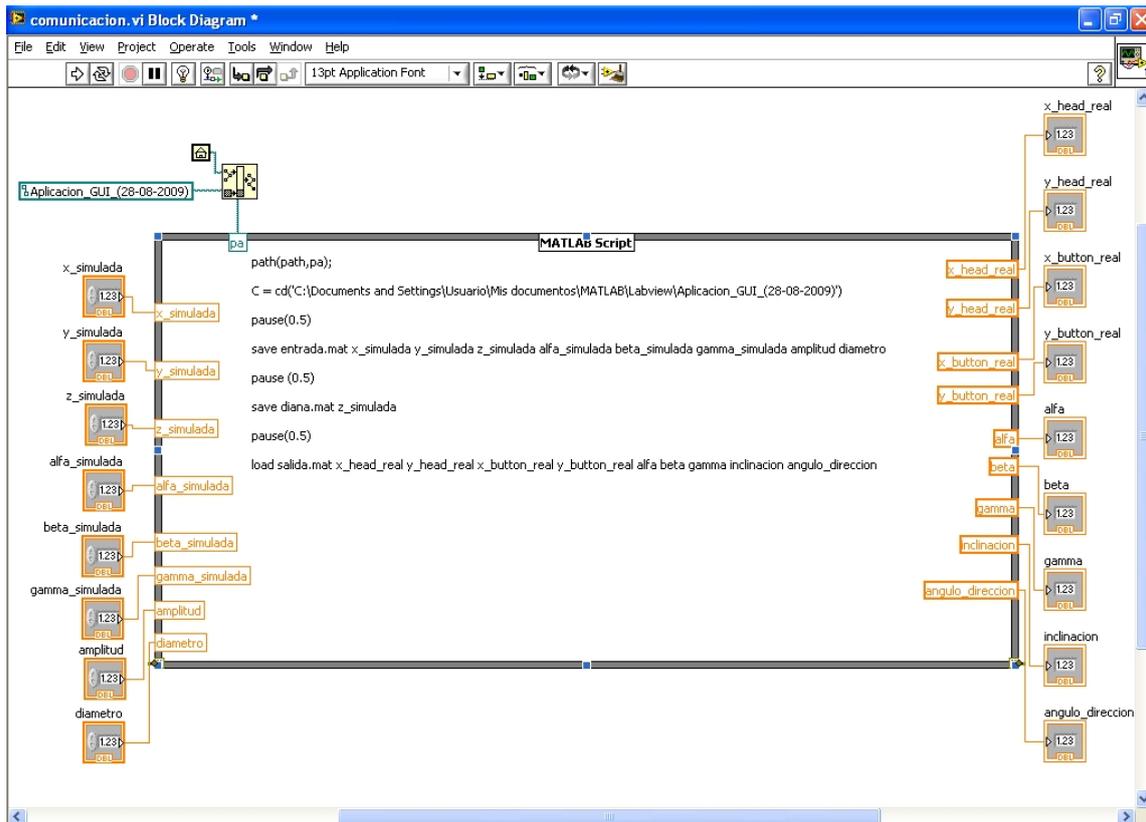


Figura 25.- Pantalla de Matlab Script Node en LabView

Por último, indicar que se han creado hasta cuatro archivos “.mat” para conseguir la total comunicación de los programas (ver Figura.- 26). La finalidad de utilizar varios archivos para la comunicación, es evitar que varios programas realicen a la vez la escritura o lectura de un mismo archivo, lo que llevaría a un error por parte de uno o varios programas por imposibilidad de obtener o guardar los datos. Los archivos son: “salida.mat”, “entrada.mat”, “diana.mat” y “gui.mat”. En el siguiente diagrama se puede observar las variables que contiene cada uno:

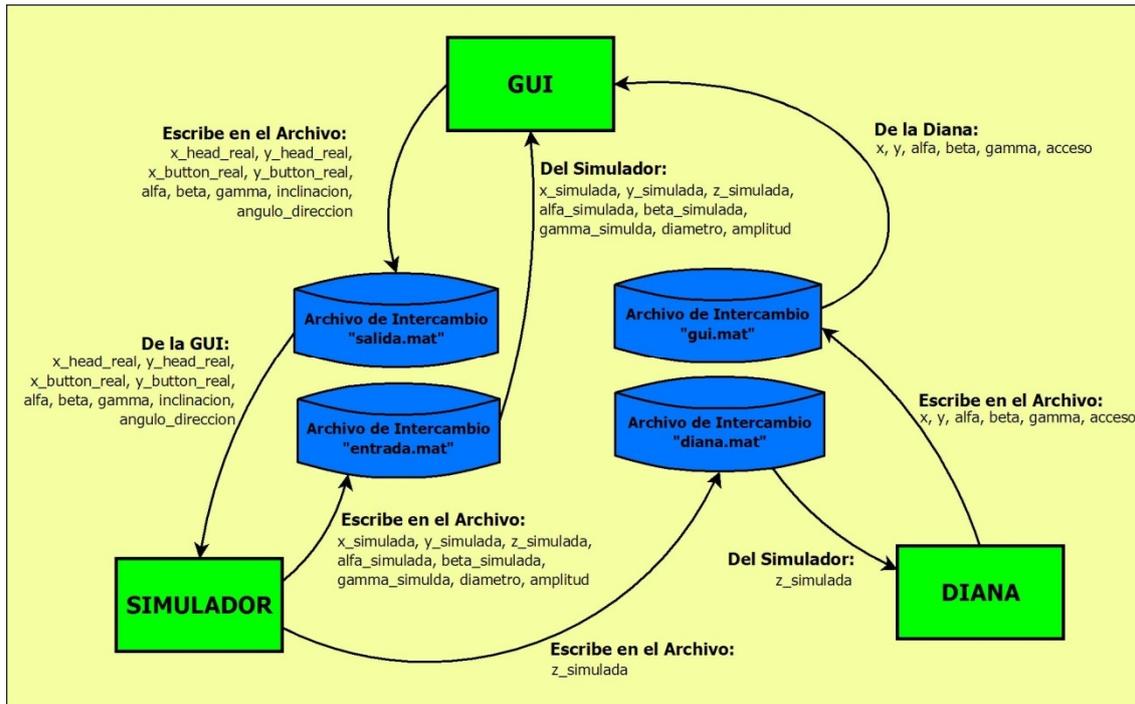


Figura 26.- Diagrama de flujo de las variables de los sistemas respecto a los archivos de intercambio

Mostrándose los cuatro archivos de intercambio entre los tres sistemas, y las variables que se escriben y leen en cada uno, indicando con las flechas el sentido que toman.

3. Comunicación por Socket

La comunicación por Socket es comúnmente usada para realizar el intercambio de datos en una red Ethernet bajo el tipo cliente-servidor. Más exactamente, un socket es un punto de comunicación por el cual un proceso puede emitir o recibir información.

Este tipo de comunicación necesita ser configurado mediante una dirección IP, un protocolo de transporte y un número de puerto, por lo que está muy orientado a la comunicación para programas que se ejecuten en diferentes máquinas. En el caso de nuestro sistema no es necesario, puesto que se ejecutarán todos los programas en el mismo ordenador, por lo que no es a priori una ventaja frente a otros métodos puesto que además obliga a realizar la configuración de los puertos de red.

El inconveniente que supone tener que dar formato a los valores obtenidos en esta comunicación, en función del dato que se vaya a recibir, ha sido fundamental a la hora de desestimar este método de comunicación frente al

elegido. Sin embargo, hay que tener presente este método de comunicación en el caso de que se realizase la integración de un ordenador industrial en la propia diana (ver apartado 6.2. Trabajos Futuros), con lo que tendría muchas posibilidades de ser el más adecuado, teniendo en cuenta que se realizaría la transmisión de datos desde la diana al ordenador central por medio de conexión de red.

4. Librería de Enlace Dinámico DLL

Dynamic Link Library o más comúnmente DLL es el término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo.

Un archivo DLL contiene funciones que el programa ejecutable puede llamar durante la ejecución, es decir, un archivo DLL es una biblioteca de funciones que el programa puede vincular dinámicamente. Un vínculo puede ser estático o dinámico. Toda la información de dirección necesaria por el programa para tener acceso a la función de biblioteca es fija cuando el archivo ejecutable se crea y permanece sin cambios durante la ejecución.

Matlab permite usar las funciones de estas librerías que están programadas en C++ a través de una interfaz de línea de comandos. Esta interfaz ofrece la posibilidad de cargar una librería externa en Matlab y acceder a cualquiera de las funciones definidas en dicha librería. Aunque los tipos de datos son diferentes en Matlab y en C++, en muchos casos es posible pasar los tipos de datos de Matlab a C++ sin tenerse que preocupar de la conversión porque Matlab la realiza de forma automática.

Para crear un librería dinámica (*.dll) y poder mantener comunicación e intercambio de datos con otras aplicaciones externas (LabVIEW), en primer lugar, es necesario crear una librería dinámica, que contenga todos los archivos de funciones “*.m” de nuestra carpeta específica de trabajo de Matlab utilizando para ello el compilador interno de MATLAB.

Acto seguido, se genera la función “wrapper” mediante el programa Visual Studio. La función “wrapper” es una función que crea un envoltorio alrededor de la librería dinámica anteriormente generada por Matlab. El objetivo que tiene la creación de dicho envoltorio es la programación de una serie de declaraciones para poder acceder a las funciones de la librería dinámica compilada anteriormente desde Matlab y a sus respectivas variables tanto de entrada como de salida, para que otras aplicaciones externas puedan intercambiar datos, comunicarse y acceder a las funciones de dicha librería dinámica.

El último paso es generar el archivo de librería dinámica definitivo, de extensión “.dll” utilizando el compilador propio de Visual Studio para compilar la función “wrapper”. A continuación, se adjunta la librería dinámica generada por Matlab a la nueva librería dinámica de la función “wrapper” para que ésta última pueda acceder a las funciones de la librería de nuestro programa y de esta manera mantener intercambio de variables con otras aplicaciones externas, en este caso, LabVIEW.

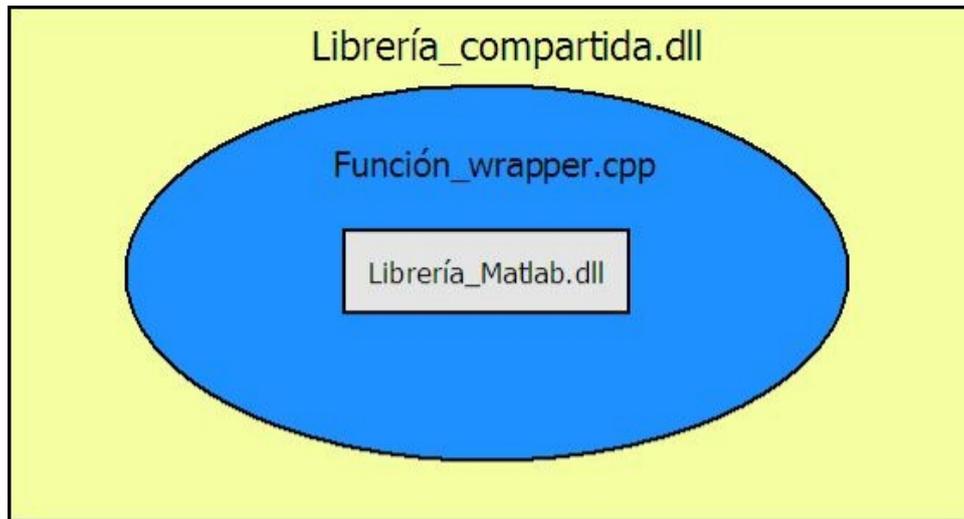


Figura 27.- Esquema de construcción de estructura de librería DLL

Por último, comentar que el sistema de comunicación entre LabVIEW y Matlab mediante librería de enlace dinámico se ha descartado debido a un error de compatibilidad producido por alternar el software de Matlab y Visual Studio de diferentes versiones. El error consistía en que una vez generada la librería dinámica de forma correcta mediante el compilador de Visual Studio, no se podía acceder a la función que inicializa la librería dinámica generada por Matlab, resultando imposible que se ejecutara cualquier instrucción de código interno de funcionamiento de la librería dinámica.

4.3. Herramientas de Programación

4.3.1. MATLAB

MATLAB (abreviatura de MATrix LABoratory, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las *cajas de herramientas (toolboxes)*; y las de Simulink con los *paquetes de bloques (blocksets)*.

Características principales

- Lenguaje de alto nivel para cálculo técnico.
- Entorno de desarrollo para la gestión de código, archivos y datos.
- Herramientas interactivas para exploración, diseño y resolución de problemas iterativos.
- Funciones matemáticas para álgebra lineal, estadística, análisis de Fourier, filtraje, optimización e integración numérica.
- Funciones gráficas bidimensionales y tridimensionales para visualización de datos.
- Herramientas para crear interfaces gráficas de usuario personalizadas.
- Funciones para integrar los algoritmos basados en MATLAB con aplicaciones y lenguajes externos, tales como C/C++, FORTRAN, Java, COM y Microsoft Excel.

Con el lenguaje de MATLAB, se puede programar y desarrollar algoritmos rápidamente puesto que no hay que realizar tareas administrativas de bajo nivel, tales como declarar variables, especificar tipos de datos y asignar memoria. En muchos casos, MATLAB elimina la necesidad de bucles "for" y otros bucles de alto coste computacional, por medio de funciones implementadas.

Al mismo tiempo, MATLAB ofrece todas las características de los lenguajes de programación tradicionales, que incluyen operadores aritméticos, control de flujo, estructuras de datos, tipos de datos, programación orientada a objetos (OOP) y depuración.

Herramientas de desarrollo

MATLAB incluye herramientas de desarrollo que ayudan a implementar algoritmos eficientemente. Las siguientes son las principales:

- **MATLAB Editor:** Funciones de edición y depuración estándar, como establecimiento de puntos de interrupción y simulaciones paso a paso.
- **M-Lint Code Checker:** Analiza el código y recomienda modificaciones para mejorar el rendimiento y mantenimiento.
- **MATLAB Profiler:** Registra el tiempo que tarda en ejecutarse cada línea de código.
- **Directory Reports:** Explora todos los archivos de un directorio y crea informes sobre la eficiencia del código, las diferencias entre los archivos, las dependencias de los archivos y la cobertura del código.

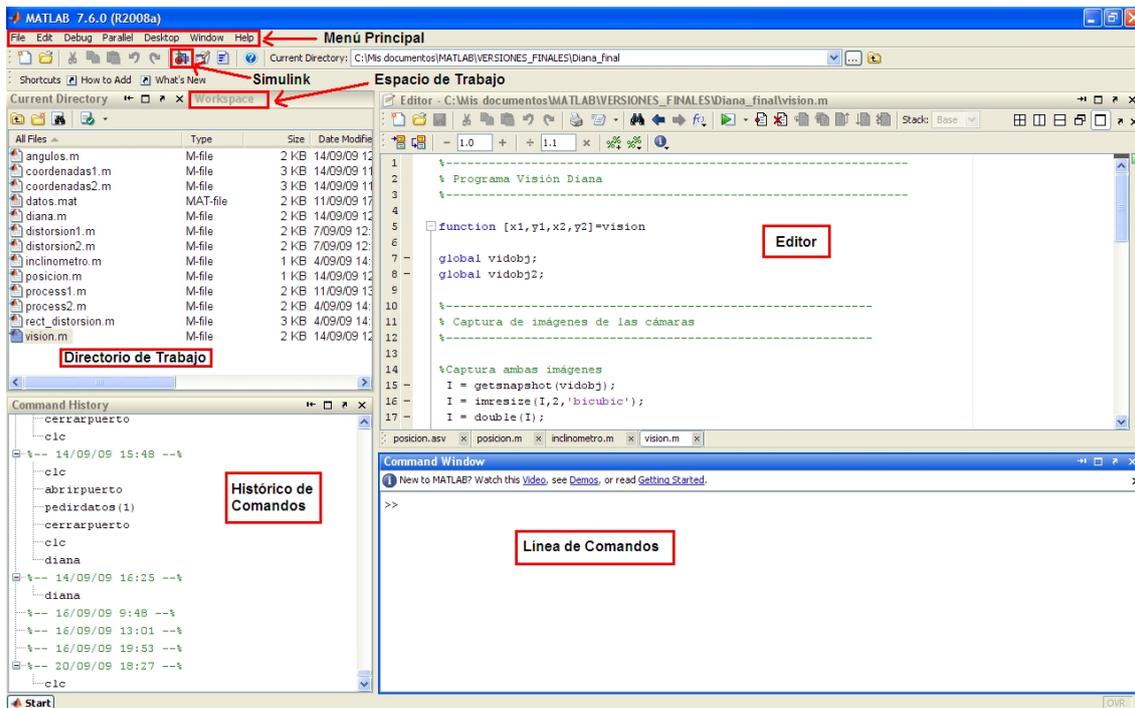


Figura 28.- Vista de la ventana de trabajo de Matlab

4.3.2. MPLab IDE

MPLAB es un editor IDE gratuito, destinado a productos de la marca Microchip. Este editor es modular, permite seleccionar los distintos microprocesadores soportados, además de permitir la grabación de estos circuitos integrados directamente al programador.

En adición a esta herramienta, se puede integrar una herramienta de compilación para lenguaje C, llamada “MCC 18”, permitiendo realizar la depuración y programación del dispositivo en este lenguaje.

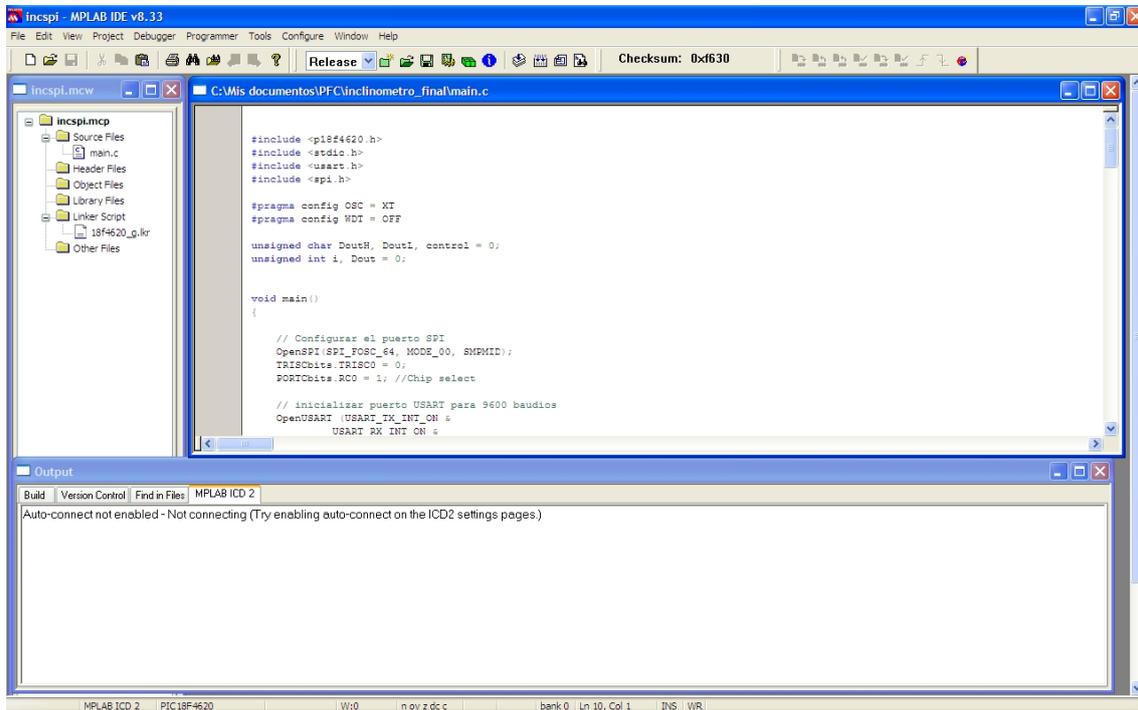


Figura 29.- Vista de ventana principal MPLabIDE

Tenemos las dos herramientas principales siguientes:

- **Debugger:** Permite realizar la compilación y depuración del programa, para comprobar los cambios realizados en el código sin tener que programarlo. Se puede realizar una ejecución paso a paso, con el uso de “breakpoints”.
- **Programer:** Realiza la programación del dispositivo, una vez que el programa ha pasado todas las pruebas pertinentes de compilación y depuración, quedando el programa en la memoria interna no volátil del dispositivo.

4.4. Descripción de los Programas

4.4.1. Programas creados en Matlab

La estructura software del sistema para la diana se ha llevado a cabo en el programa de desarrollo Matlab. Una de las principales razones para realizarlo

con este software además de las comentadas en las herramientas de programación (apartado 4.3.1), es la fácil integración de componentes hardware en la programación por medio de puestos, tanto USB como serie, por lo que es muy adecuado para este caso, al hacer uso de dos cámaras de conexión USB, además de un inclinómetro de conexión por puerto serie.

También se debe destacar, que se ha intentado realizar los programas que componen este proyecto de una forma modular, es decir, intentando que cada función y posibles subfunciones permitan la solución a cada problema del sistema por separado, pudiendo ser sustituidas por otras sin tener que hacer un replanteamiento global del sistema. Esto es así, debido a la posible revisión para un cambio o mejora de alguna de las partes de este proyecto, al estar enfocado a un desarrollo de innovación, además de contar con un hardware, como es la diana o el inclinómetro, que se encuentra en una fase de prototipo, por lo que cambios futuros en su estructura podrían necesitar de modificaciones o mejoras del software.

La siguiente lista comprende los nombres de todas las funciones y subfunciones realizadas, y que se irán explicando a continuación:

- diana.m
- vision.m
- distorsion1.m
- distorsion2.m
- rect_distorsion.m
- process1.m
- process2.m
- coordenadas1.m
- coordenadas2.m
- angulos.m
- inclinometro.m
- posicion.m
- parametros.m

Las funciones se podrían clasificar según la aplicación a la que se refiere, en tres grandes bloques, de la siguiente manera:

- Programa principal del sistema con la función “diana.m” y la función de parámetros definibles “parámetros.m”.
- Sistema de visión, que engloba a las funciones “visión.m”, “distorsion1.m”, “distorsion2.m”, “rect_distorsion.m”, “process1.m” y “process2.m”.
- Cálculo de coordenadas, con las funciones “coordenadas1.m”, “coordenadas2.m”, “angulos.m”, “inclinometro.m” y “posicion.m”.

En el siguiente diagrama de flujo puede apreciarse el orden a las llamadas a las funciones, en cada ejecución del programa:

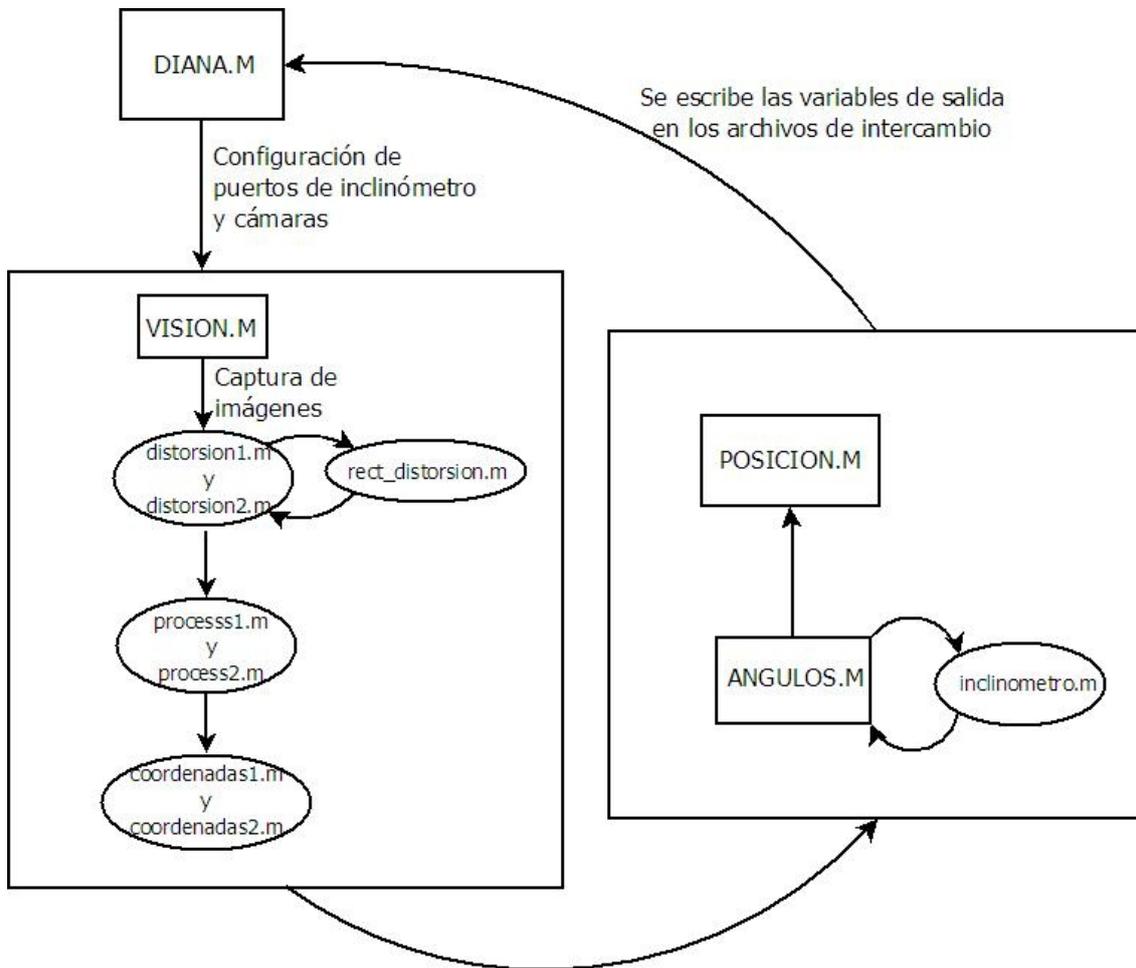


Figura 30.- Diagrama de la arquitectura de la aplicación

A continuación se explicará cada uno de los archivos o programas que intervienen en el diagrama.

Diana.m

El programa diana.m es la función principal del software, el cual controla el proceso, tanto de ejecución/espera del programa, como la secuencia de llamadas a las funciones principales.

Este programa se puede estructurar en tres partes fundamentales: sentencias de inicialización, cuerpo principal y sentencias de finalización.

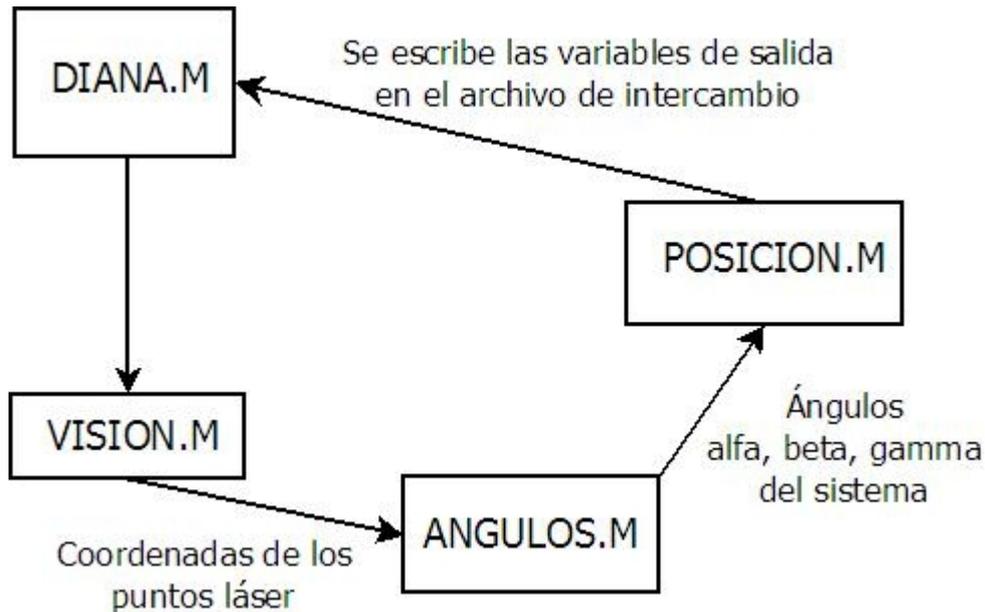


Figura 31.- Diagrama de flujo de las funciones principales del software

Dentro de las sentencias de inicialización mediante la llamada a la función “parámetros.m”, nos encontramos con la inicialización y configuración del puerto serie, para la comunicación del programa con el inclinómetro con el que está dotada la diana. La inicialización se realiza previo al uso del puerto, debido al alto tiempo de espera que se necesita para ello de unos 3 segundos, por lo que se asigna al puerto una variable de tipo global, para poder hacer uso de él en la función “inclinometro.m”, y se toma una lectura inicial del inclinómetro, para tomarlos como valores en el origen. También se lleva a cabo la inicialización de las variables constantes

Además se lleva a cabo la inicialización de las variables principales del programa, las cuales serán las que se considerarán como salida final del programa y son: x, y, alfa, beta, gamma y acceso.

El cuerpo principal del programa se caracteriza por el uso de un bucle while infinito, el cual discriminará entre si debe llevar a cabo la ejecución del programa o permanecer en espera mediante la variable “z_simulada”. Esta variable es la que corresponde al avance de la tuneladora, y es leída del archivo de intercambio de variables. Actualmente se obtiene del simulador, no siendo un dato real, aunque en una situación normal de funcionamiento sería un dato directo, puesto que se obtiene de una rueda de medida longitudinal colocada a la entrada del pozo de ataque del túnel, del cual no se podía disponer en estos momentos. Si no se ve ningún cambio en ella desde la última lectura, se realizará una pausa de 0.5 segundos para evitar estar leyendo el archivo de intercambio constantemente, y se repetirá el proceso. En caso de que haya una diferencia de valor entre la lectura anterior y la nueva, se llevará

a cabo el flujo principal, para el cálculo y obtención de las nuevas variables de salida.

Por último, en el cuerpo principal del programa, cabe destacar que las variables de salida son guardadas en el archivo de intercambio “gui.mat”, para permitir la comunicación y uso de ellas en una GUI desarrollada para mostrar los puntos de la trayectoria de la tuneladora.

En cuanto a las sentencias de finalización del programa, sirven para liberar el puerto serie asignado al inclinómetro de su uso, para que en posteriores ejecuciones del programa no se lleven errores debido al uso del puerto.

Vision.m

Esta función lleva a cabo la captura de nuevas imágenes de las cámaras de la diana, así como las llamadas a las funciones que permiten calcular las nuevas coordenadas relativas (x1,y1,x2,y2), para conocer la nueva posición de la tuneladora.

En el siguiente diagrama de flujo, se detalla la secuencia de las funciones realizadas:

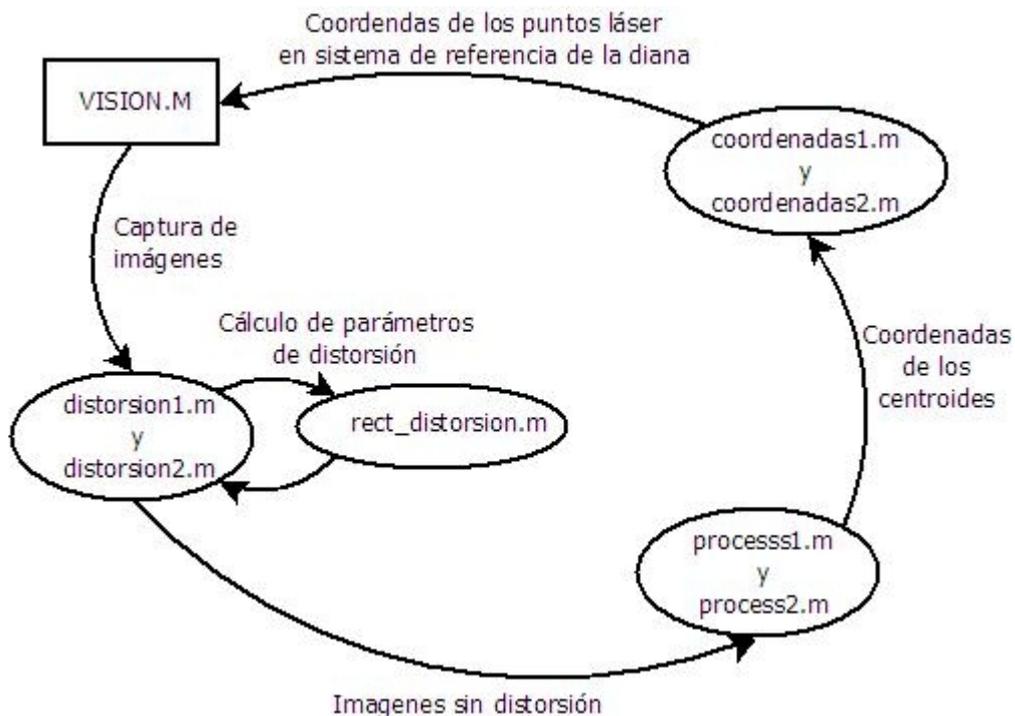


Figura 32.- Diagrama de flujo de la función “vision.m”

Inicialmente, se lleva a cabo la captura de las imágenes, siendo guardadas como variables para su posterior tratamiento. Además, debido a la resolución no excesivamente alta, y con el fin de obtener mayor variación en la detección de los puntos que genera el láser en la diana, se lleva a cabo un redimensionamiento del tamaño de ambas imágenes, duplicándose hasta 640x480 píxeles, y aplicando un filtro de interpolación bicúbico, mediante el siguiente comando:

```
I = imresize(I,2,'bicubic');
```

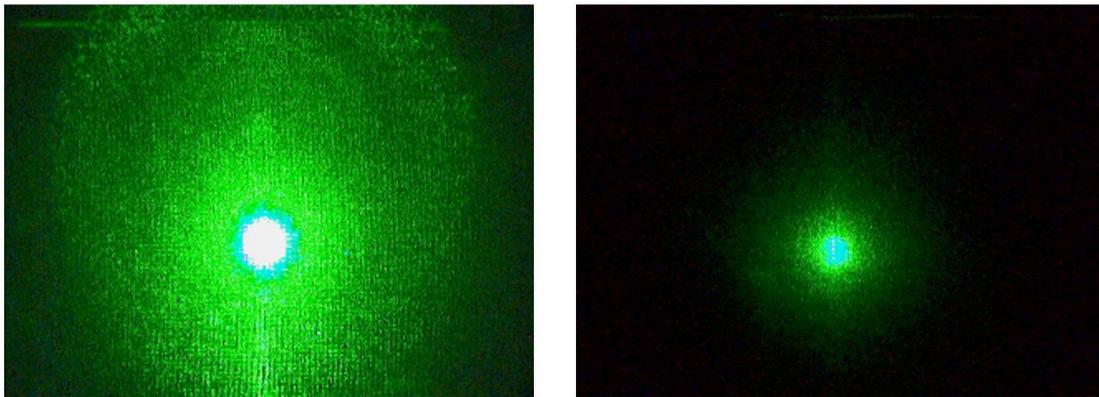


Figura 33.- Imágenes de ventana y fondo de la diana obtenidas por las cámaras

Además, se realiza un primer tratamiento de la imagen, que consiste en binarizar ambas imágenes, y realizar los giros necesarios para tener las imágenes desde un punto de vista igual al que obtendría un observador frente a la ventana de la diana. En la imagen de la ventana de la diana, se realiza un giro respecto a un eje horizontal, mientras que en la imagen del fondo se realizan dos giros, uno con respecto a un eje vertical y otro respecto a un eje horizontal.



Figura 34.- Imágenes de las cámaras de la ventana y fondo de la diana binarizadas

En la binarización, cabe destacar que el comando de Matlab correspondiente, realiza inicialmente la conversión de la imagen a escala de grises automáticamente, y posteriormente asigna los valores de cada pixel a blanco o negro, en función de un parámetro umbral que se especifica manualmente en la función, y con el que podemos ajustar la correcta binarización de la imagen, tal y como se muestra en la siguiente línea:

```
I_binarizada=im2bw(I_original, umbral);
```

A continuación, se realiza la llamada a dos funciones, primero processX.m, la cual calcula las coordenadas en pixel de los puntos generados por el laser en la diana, y la función coordenadasX.m, donde se transforman las posiciones de los puntos de pixel a milímetros.

Distorsion1.m y Distorsion2.m

Existen dos funciones distorsión, debido a la necesidad de trabajar con valores diferentes para algunas de las variables definidas, entre la imagen obtenida de la ventana y del fondo de la diana. Esto es debido a que son las variables que definen las componentes intrínsecas de cada cámara y, por tanto, diferentes en cada caso, y calculadas mediante la calibración de cada cámara (ver Anexo).

Las variables obtenidas en la calibración y definidas en este programa son: distancia focal, punto principal, coeficiente de apuntamiento, y la matriz de coeficientes de distorsión radial y tangencial.

Estos parámetros modifican la imagen para eliminar la distorsión producida por la lente, en forma de barrilete, mediante la subfunción "rect_distorsión". Esta subfunción se basa en el modelo de Heikkilä (ver apartado 5.3 y Bibliografía), para la eliminación de la distorsión, y obtenemos una imagen en la cual tenemos una relación lineal entre los puntos de la imagen, lo cual nos permite calcular las distancias entre el punto superior izquierdo de la imagen, que toma Matlab como referencia de la imagen, y el punto centroide del láser, que posteriormente calcularemos.

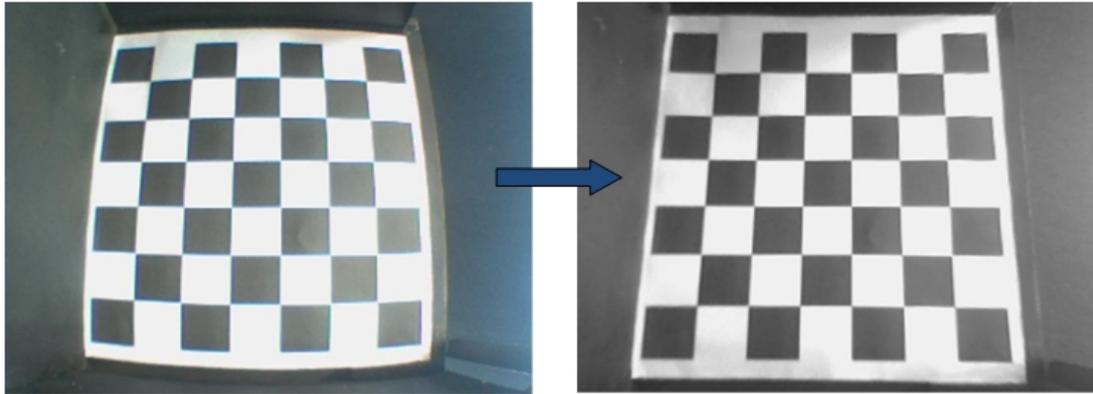


Figura 35.- Visualización de ejemplo de corrección de la distorsión

Aunque necesitamos dos funciones para definir los parámetros de cada cámara, que varían sobre todo en el punto focal, debido a una diferencia de enfoque al tener la cámara que apunta hacia la ventana de la diana colocado un filtro, aunque comparten ambas funciones la subfunción “rect_distorsión”, ya que se corrige la distorsión con el mismo método y, por tanto, las mismas ecuaciones.

Process1.m y Process2.m

Ambas funciones son llamadas desde la función visión.m, y en ellas se obtiene las coordenadas x e y de los centroides de los puntos del láser en la diana. Debido a que se deben definir un parámetro que difiere entre las dos cámaras de la diana, se debe discriminar entre ellas a la hora de ejecutarlas, siendo process1 la encargada del cálculo en la cámara de la ventana, y process2 en la cámara que apunta al fondo de la diana.

La secuencia que realiza el código de esta función sería: Obtención de la relación entre píxeles vecinos mediante el comando “strel”; Transformación morfológica, aplicando el método de opening seguido de closing (apertura + cerradura), para en primer lugar rellenar los huecos blancos de los objetos en la imagen y en segundo lugar, eliminar ruido en los bordes de los objetos, con lo que se aumentará la exactitud en las medidas que se realicen en la imagen; Etiquetado de los objetos que se detecten en la imagen, mediante el comando “bwboundaries”, reconociendo como objeto los regiones negras sin píxeles blancos en su interior, al añadirle la condición ‘noholes’; Cálculo de los centroides de los objetos etiquetados, con la función “regionprops (L, 'centroid')” formando un array con sus coordenadas, para posteriormente asignárselos a las variables (x, y) de salida de la función.

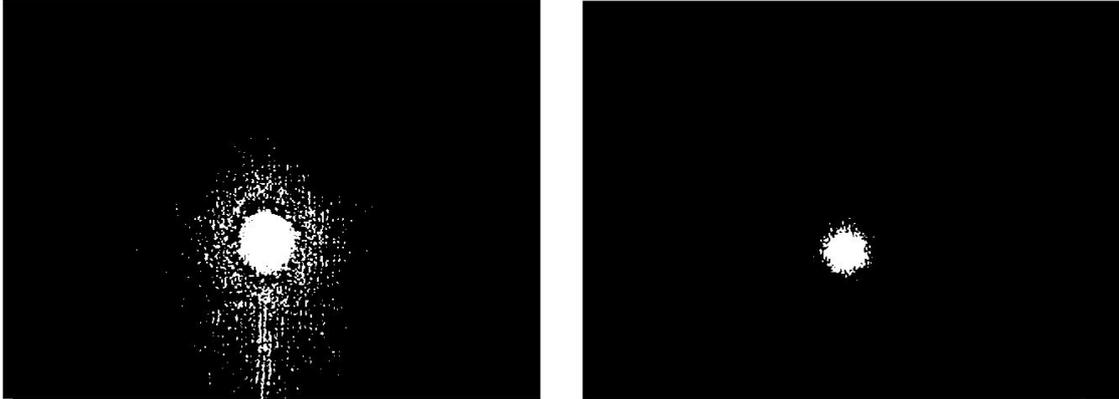


Figura 36.- Imagen binarizada en la que posteriormente se han eliminado los objetos menores a un tamaño definido

Cabe destacar el parámetro 'tamaño_objeto' definido en la función:

```
Imagen_etiquetada = bwareaopen(imagen_original,tamaño_objeto);
```

El cual es el que difiere en ambas funciones, y el que permite eliminar los objetos de la imagen con un número de píxeles menores a él, con lo que se consigue una imagen libre de ruido, obtenidos tanto por errores de la lente de la cámara, como por brillos del propio láser o de la iluminación exterior. Si este parámetro es elegido correctamente, lo cual se puede llevar a cabo mediante sucesivas pruebas, deberíamos obtener un solo objeto en la imagen, que se corresponde con el generado por el láser. De no ser así, podría darse el caso de identificar la posición de ruido en la imagen como la posición del punto del láser, y por ello la importancia de este parámetro.

Coordenadas1.m y Coordenadas2.m

Ambos programas se encargan de calcular las posiciones relativas de los puntos generados por el láser en la diana, y recogidas por ambas cámaras. La función "coordenadas1.m" se encarga del proceso de la cámara de la ventana, mientras la función "coordenadas2.m" se encarga de la del fondo. Al igual que en anteriores funciones, la distinción entre una u otra función dependiendo de la cámara que haya tomado la imagen, se debe a la necesidad de definir variables distintas para cada caso. En esta aplicación, se tienen que definir los puntos en píxeles que definen las esquinas exteriores de las imágenes y, debido a pequeñas variaciones en cuanto a su colocación y orientación, estas posiciones son distintas. Además, también se definirá el tamaño real de la imagen recogida en milímetros, puesto que la función de estos programas es calcular la correspondencia entre las coordenadas pixel, calculadas

anteriormente, y devolver la posición en milímetros, siendo esta posición relativa al punto origen que se encuentra todavía en la esquina superior izquierda de la imagen, y que posteriormente se referenciará al centro de la misma.

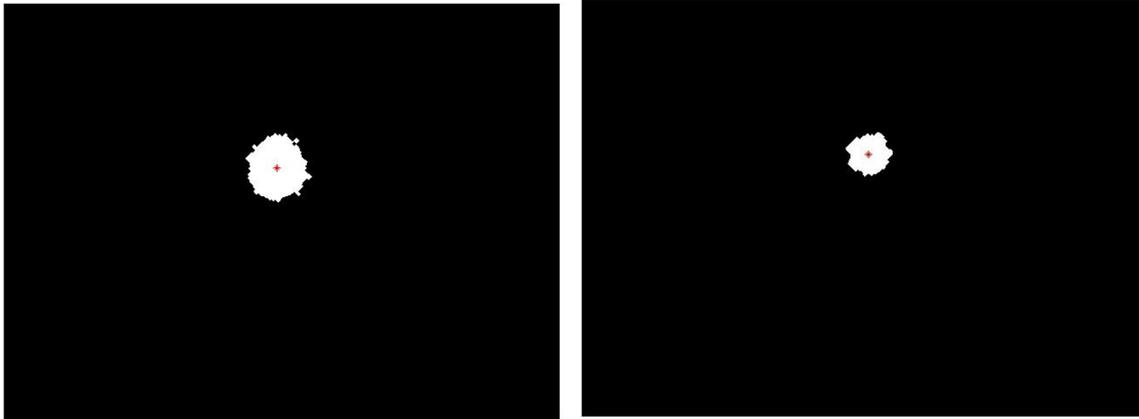


Figura 37.- Imágenes de ventana y fondo de la diana con los centroides marcados

Una vez definidas estas coordenadas exteriores de la imagen, tanto en pixeles como en milímetros, se lleva a cabo una función con el objetivo de calcular una relación lineal entre posición pixel y posición real (o mundo). La solución al problema se lleva a cabo por medio de la resolución de un sistema de ecuaciones, calculando primero los autovalores del sistema, para formar los autovectores a partir de ellos. A partir de los autovectores, creamos una matriz H de la transformación mundo-cámara, o lo que es lo mismo, entre posición pixel y la posición en milímetros, con la que se multiplican las coordenadas pixel introducidas en la función, para obtener como salida las coordenadas en milímetros de el punto generado por el láser.

Por último, debido a que Matlab trabaja por defecto referenciando las distancias de puntos de una imagen a la esquina superior izquierda, se realiza un cambio de variable por medio de dos estructuras if-else, una para cada coordenada obtenida, a una referencia en las distancias en el centro de la imagen. Se lleva a cabo gracias a que se conocen las dimensiones de la imagen en la realidad, y se puede delimitar a qué cuadrante debería de pertenecer cada punto.

Angulos.m

Esta función se encarga de calcular los ángulos alfa y beta, correspondientes a los ejes X e Y, a partir de las coordenadas x_1 , y_1 , x_2 e y_2 , que son las coordenadas en el fondo y la ventana de la diana del láser, siendo parámetros de entrada de la función. Además, dentro de esta función, se hace la petición del ángulo gamma al inclinómetro, puesto que no puede ser calculado por el

sistema de visión que comprende el láser y las cámaras, además de pedir también el valor del ángulo alfa, dato que podemos conocer al haber utilizado un inclinómetro biaxial, y que nos permitirá tener una medida extra para compararlo con el calculado por la trayectoria del láser, como forma de asegurarnos de la veracidad de los cálculos. Se considerará que el ángulo alfa de la diana es un valor aceptable, en el caso de que la diferencia entre ambos valores no sobrepase la precisión en el cálculo del ángulo en la diana, que es de 0.5 grados.

Para la petición de datos al inclinómetro, se ha desarrollado una subfunción llamada "inclinometro.m", la cual hará una petición de los dos ángulos en los que está orientado cada vez que se llame a ella.

En la siguiente imagen, se muestra un esquema en el que se puede observar el sistema de coordenadas usado, así como los sentidos de referencia que tienen los tres ángulos del sistema:

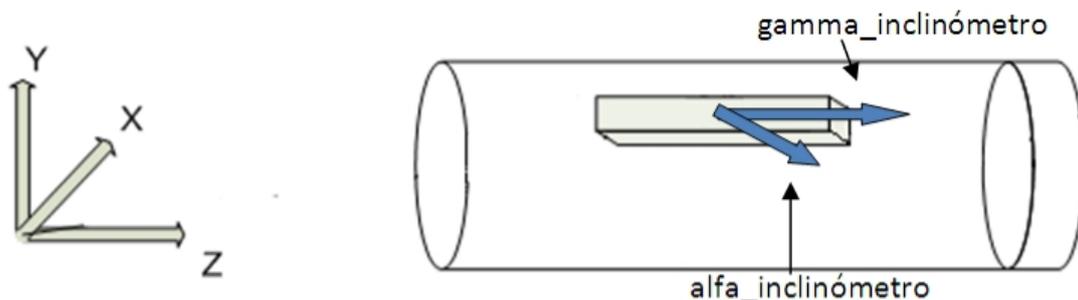


Figura 38.- Esquema del sistema de referencia del inclinómetro

Para el cálculo de los ángulos, lo primero a tener en cuenta es si se ha producido un giro en el eje Z de avance, si se tiene un ángulo gamma en la petición al inclinómetro. En caso de que gamma sea distinto de cero, lo primero a realizar es realizar ese mismo giro en los ejes X e Y, puesto que se tienen girados los ejes de referencia de los paneles de la diana. En la siguiente imagen, se muestra como se lleva a cabo:

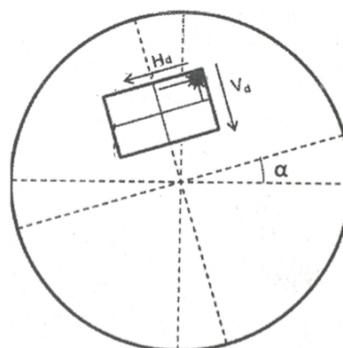


Figura 39.- Imagen de corrección del ángulo gamma
Por lo que realizamos las siguientes operaciones para ambos ejes X e Y:

```
delta1=atan2(y1,x1);  
d= sqrt((y1^2)+(x1^2));  
x1g= d*cos(gamma+delta1);  
y1g= d*sin(gamma+delta1);
```

Una vez referenciados los ejes x,y relativos que hemos tomado en los dos paneles de la diana, procedemos a calcular los ángulos alfa y beta, correspondientes a los ejes X e Y de el sistema de referencia general, por medio de las siguientes ecuaciones, y teniendo en cuenta que la distancia entre paneles de ventana y fondo de la diana es constante e igual a 680 mm:

```
%Giro en X (se genera un ángulo alfa):  
dy=(y2g-y1g);  
alfa=atan2(dy,distancia);  
  
%Giro en Y (se genera un ángulo beta):  
dx=(x2g-x1g);  
beta=atan2(dx,distancia);
```

Por tanto, se obtiene la salida de la función, que serían los valores de los tres ángulos, alfa, beta y gamma, referenciados al sistema de coordenadas global, con lo que ya se está en disposición de calcular las nuevas coordenadas de la tuneladora.

Posicion.m

En la última función del sistema, se realiza el cálculo de las coordenadas X e Y de la nueva posición de la tuneladora. Debido a que el fin del cálculo de estas coordenadas es la representación en una GUI de guiado de la tuneladora, se ha llegado al convenio de tomar como punto de salida, el ubicado en el centro del panel de la ventana de la diana.

Al comienzo de la función, se buscará en el archivo de intercambio “datos.mat” la variable “z_simulada” asignada al avance en el eje Z de la máquina. A continuación, se ha hecho uso de una toolbox para Matlab llamada “Robotics Toolbox for Matlab”, la cual nos facilitará el trabajo del cálculo de que se realizará por medio de coordenadas homogéneas. De esta toolbox se usan los comandos “transl” y “rpy2tr”, siendo el primero para el cálculo de una matriz homogénea de traslación, en la cual introduciremos la variable de avance en el eje Z como parámetro, y la segunda una función que permite obtener una matriz de coordenadas homogéneas a partir de los ángulos de Euler alfa, beta y gamma, obtenidos en la función “angulos.m”. Multiplicando la matriz de rotación por la de traslación, obtenemos una matriz de coordenadas homogéneas en la cual se produce una traslación seguida de una rotación, al

multiplicarla por un vector de coordenadas, según se expresa en el siguiente cálculo matricial:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \alpha \cos \beta - \sin \alpha \sin \beta \cos \gamma & -\cos \alpha \sin \beta \cos \gamma & \sin \alpha \sin \beta & 0 \\ \sin \alpha \cos \beta + \cos \alpha \sin \beta \cos \gamma & -\sin \alpha \sin \beta + \cos \alpha \cos \beta \cos \gamma & -\cos \alpha \sin \beta & 0 \\ \sin \gamma \sin \beta & \sin \gamma \cos \beta & \cos \gamma & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Además, al haber tomado el origen de referencia como el centro del eje de simetría de la tuneladora y a la entrada del pozo de ataque, y se busca obtener las coordenadas del punto del centro de la ventana de la diana, se realiza un desplazamiento en el eje Z, previo a la multiplicación por la matriz homogénea, sabiendo que es debido a la colocación de la diana en el interior de la máquina tuneladora y, por tanto, constante e iguales a -1728 mm sobre el eje Z.

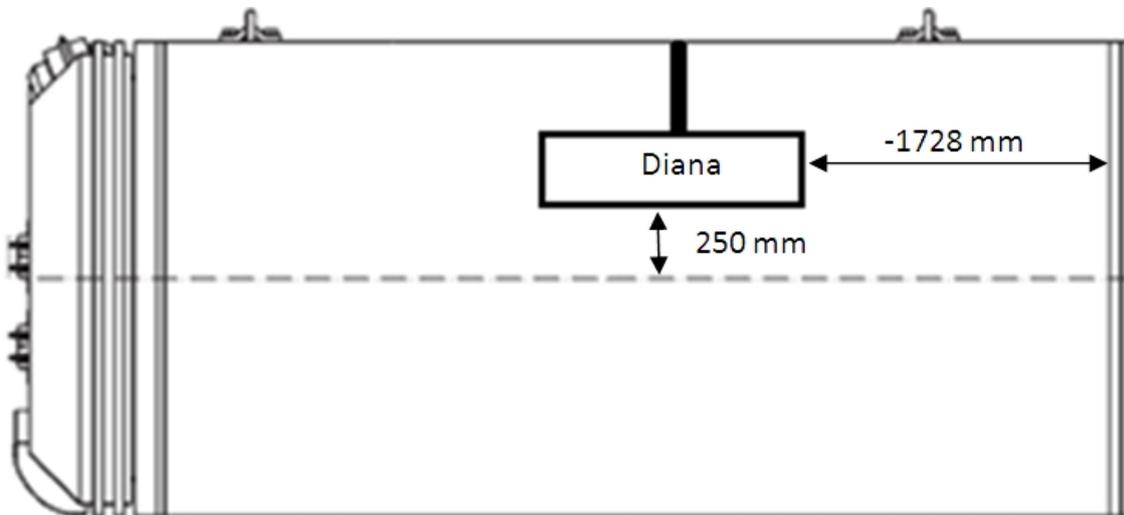


Figura 40.- Esquema de colocación de la diana dentro de la tuneladora

4.4.2. Programa inclinómetro

MAIN.C

Este programa se ejecuta en la tarjeta de desarrollo del inclinómetro, el cual es el dispositivo que permite en la diana calcular el ángulo gamma, sobre el eje Z o de avance de la tuneladora. Además, al disponer de un inclinómetro de dos ejes, se ha colocado de tal forma que permita la lectura del ángulo alfa, que se produce en el eje X, para obtener ese dato al mismo tiempo que se calcula por la proyección del láser en la diana, para estimar que los cálculos son correctos.

El sistema consta fundamentalmente de un micro con salida por puerto serie, además de un inclinómetro, conectado por la entrada SPI del micro.

En la cabecera del código se muestran las inclusiones de librerías necesarias para el modelo de micro usado, los puertos y la librería estándar de entrada/salida, además de la configuración de parámetros del reloj, para 4MHz, y desactivación del “Watch Dog Timer”:

```
#include <p18f4620.h>
#include <stdio.h>
#include <usart.h>
#include <spi.h>

#pragma config OSC = XT
#pragma config WDT = OFF
```

Posteriormente, se realiza la inicialización de las variables que van a ser usadas:

- “Dout” es la salida del valor por el inclinómetro en valor decimal, obtenido del inclinómetro hacia el puerto serie.
- “DoutH” y “DoutL” son las variables (parte alta y baja respectivamente) obtenido directamente del puerto SPI, siendo un vector de 11 bits que se recibe primero la parte alta de 3 bits y, a continuación, la parte baja de 8bits, hasta completar el dato.
- “i” es una variable auxiliar para permitir hacer un contador para realizar pausas entre escritura o lectura, tanto en el puerto serie, como en el SPI.
- “Control” se utiliza como un carácter de entrada que nos permite discriminar entre la función a realizar por el inclinómetro, siendo dato de entrada por el puerto serie y escrito por el programa de Matlab “inclinometro.m” en la petición de datos. En la siguiente tabla se observa los posibles valores y comandos que se pueden usar:

Command name	Command format (Decimal)	Description:
MEAS	00000000 (0)	Measure mode (normal operation mode after power on)
RWTR	00001000 (8)	Read and write temperature data register
RDSR	00001010 (10)	Read status register
RLOAD	00001011 (11)	Reload NV data to memory output register
STX	00001110 (14)	Activate Self test for X-channel
STY	00001111 (15)	Activate Self test for Y-channel
RDAX	00010000 (16)	Read X-channel acceleration through SPI
RDAY	00010001 (17)	Read Y-channel acceleration through SPI

Tabla 4.- Comandos de instrucción para el manejo del inclinómetro

A continuación, se realiza la configuración de los puertos SPI y serie. De la configuración del puerto SPI hay que destacar que se usa el puerto del micro RC0, como “chip select” del inclinómetro. Mientras, en la configuración del

puerto serie se configura con las interrupciones activas de entrada y salida, en modo asíncrono y con dato de 8 bits. También se configuran las interrupciones serie como alta prioridad.

El flujo principal del programa se mantiene en un bucle while infinito, en el que primeramente se realiza una lectura de la entrada del puerto serie, para comprobar si hay petición de datos por parte del programa Matlab. Esta petición se determina si es escrito el comando correspondiente a la acción a realizar, siendo el carácter "A" para lectura del ángulo alfa, y "B" para lectura del ángulo gamma. Además, se ha implementado una estructura, a la que se puede llamar para su ejecución con el carácter "C", para realizar un "Self Test" del inclinómetro. Esta ejecución realiza un chequeo de cada eje, por medio de los comandos "14" y "15" escritos en el puerto SPI de forma secuencial, y dejando de funcionar el inclinómetro para evitar fallos en caso de error. Aunque puede ser útil en labores de inspección y buen funcionamiento, no se ha hecho uso de ella debido a que el propio inclinómetro realiza internamente un modo de "Self Test" y "Failure Detection" durante una ejecución normal, según especifica el fabricante.

En caso de realizarse una petición de uno u otro ángulo, se lleva a cabo la siguiente secuencia de programa, que es igual para ambos canales:

- Puesta a cero de la variable de salida "Dout".
- Se activa a nivel bajo el "Chip Select" del inclinómetro, con la salida del puerto del micro "RC0".
- Se escribe en el puerto SPI el comando para la petición de datos de uno de los ángulos, siendo el valor decimal "16" para el ángulo alfa del sistema (eje X del inclinómetro), y el valor decimal "17" para el ángulo gamma (eje Y del inclinómetro).
- Lectura del puerto SPI, primero se realiza la lectura de los 3 bits de nivel alto y se guardan en la variable "DoutH", y a continuación de los 8 bits restantes, guardados en la variable "DoutL", para completar la lectura total del dato.
- Se desactiva el "Chip Select" del inclinómetro, poniéndolo a nivel alto.
- Se concatena los dos valores de "DoutH" y "DoutL", para guardar el dato total en la variable "Dout" de salida.
- Por último, se escribe en el puerto serie el valor de "Dout", para que pueda ser leído por el programa de Matlab desde donde se realizó la petición de datos al inclinómetro ("inclinómetro.m"), y se realiza una pausa por medio de un bucle "for", para dar tiempo al puerto de ser leído y refrescado.

Además, y aunque no se ha hecho uso de ella, también se ha realizado una estructura de petición de temperatura. Se puede incluir en la llamada de datos de ángulos o realizarse a parte de una petición, sirviendo para la compensación de temperatura del inclinómetro. Sin embargo, al usar la comunicación digital por medio de SPI, en vez de una lectura analógica del dato, no es realmente necesario.

Por último, se muestra un diagrama de flujo donde se especifica la ejecución normal de una llamada de datos por parte del programa principal al micro conectado al inclinómetro:

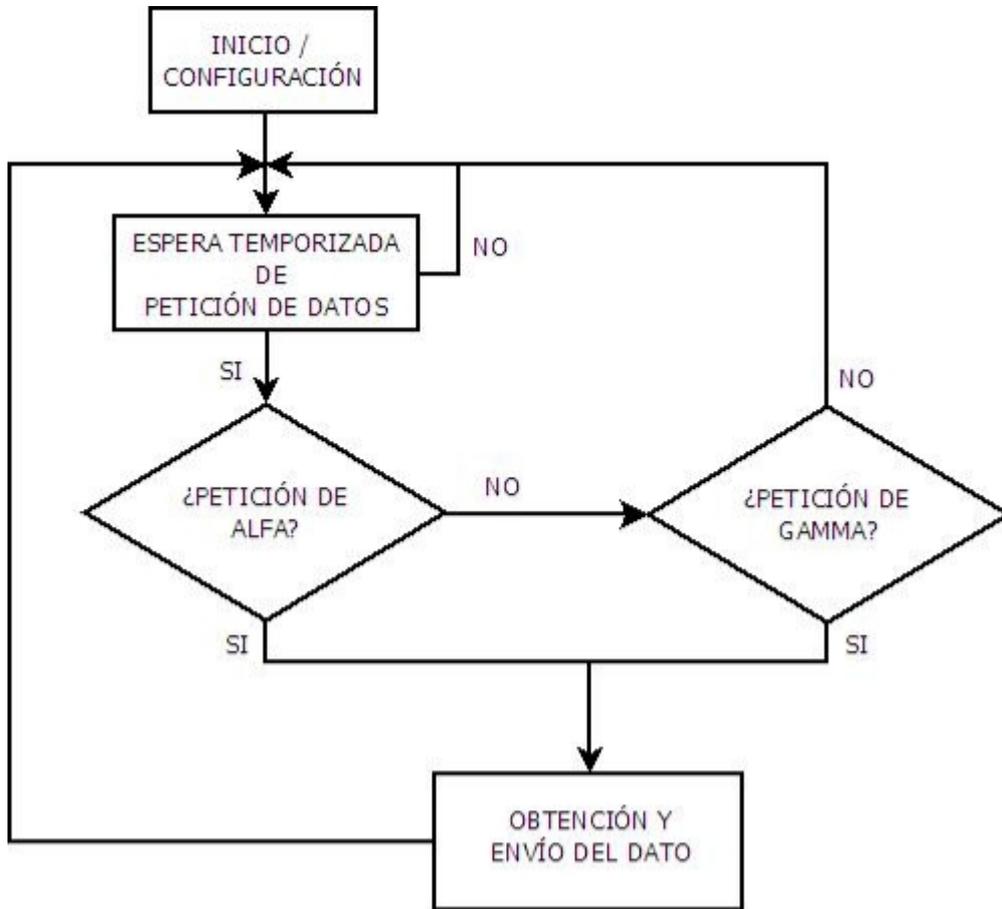


Figura 41.- Diagrama de flujo del programa del inclinómetro

5. Manual de Uso

5.1. Manejo del Programa

El programa se presenta como un archivo ejecutable por línea de comandos de MS-DOS, para la ejecución del mismo. Además, para el caso de tener que realizar alguna modificación previa, como la configuración de los puertos para el inclinómetro o las cámaras (ver apartado 4.2.3 referido a la conexión del sistema), se incluyen los archivos de Matlab que componen todo el proyecto. Si bien desde el mismo programa Matlab se puede realizar la ejecución del programa, para hacer uso de todo el sistema global, que se compone de Diana, GUI y Simulador, es necesario para la ejecución de todas las partes realizar a continuación de las modificaciones el archivo ejecutable, ya que Matlab sólo permite la ejecución de un programa a la vez. El ejecutable se realiza con el comando “mcc” y crea el archivo “diana.exe”:

```
mcc -m diana.m
```

O bien mediante la herramienta “Deploytool” de Matlab, la cual se ejecuta desde la línea de comandos escribiendo “deploytool”, para posteriormente crear un proyecto de compilación donde se elige el archivo “.m” principal por un lado y el resto de funciones que compongan el proyecto por otro, para terminar pulsando el botón “Build the project” para obtener el archivo ejecutable. Además, con este método se tiene la posibilidad de realizar el “Stand Alone” del proyecto sin más que pulsar la tecla “Package the project” una vez compilado.

Para realizar la correcta ejecución del programa, se deberá escribir el comando “diana.exe”, en una ventana de “Símbolo de Sistema” de Windows. Previo a esta ejecución, se deben de llevar a cabo todas las conexiones del sistema, como se explica en el apartado 4.2.4, puesto que en caso de no tener alguna conexión, el programa mostrará un fallo de ejecución que requerirá la reinicialización de su ejecución.

5.2. Ajuste de las Cámaras

Para conocer la relación entre los píxeles de la imagen capturada por cada cámara del sistema, frente a las dimensiones reales, hay que realizar unos pasos previos al uso del programa, si se quiere realizar alguna modificación tanto en la posición de las cámaras en la diana, como en el área de captura de las imágenes.

Antes de esto, se debe situar las cámaras en su colocación final, ajustando los bordes de la imagen al área efectiva que se quiera capturar, en este caso los paneles del fondo y la ventana de la diana, puesto que en caso de moverlas, se debe volver a realizar los pasos que se indican a continuación.



Figura 42.- Vista del hardware en el interior de la diana

Debido a que las cámaras cuentan con un objetivo ajustable manualmente, se procederá a regularlo hasta conseguir una imagen nítida tanto en definición como en luz y color.



Figura 43.- Imagen de la cámara instalada en la diana

A continuación, se realizará una calibración para cada cámara, como se indica en el apartado 5.3, para obtener los parámetros de calibración y poder corregir la distorsión en las imágenes.

Para la medida del tamaño útil de la imagen, se diseñó una sencilla plantilla con el tamaño de la zona a medir, colocándola dentro de la diana, una en la ventana y otra en el fondo, y realizando la captura de una imagen para cada cámara.

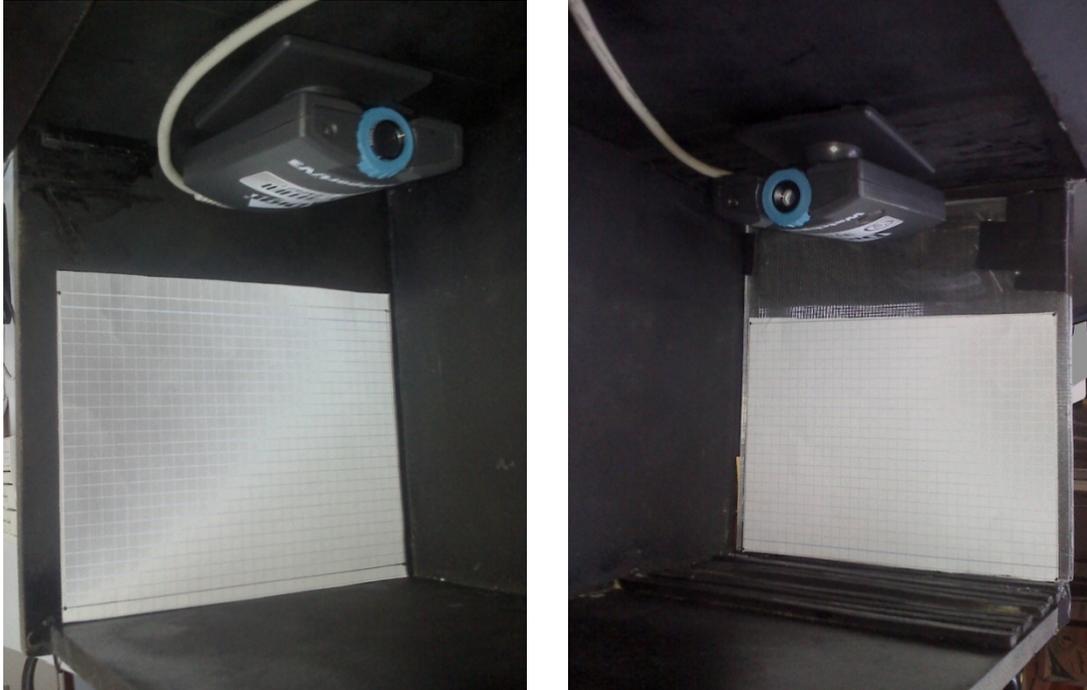


Figura 44.- Vista de la colocación de las plantillas de obtención de los bordes de la imagen

La corrección de la distorsión puede conseguirse por medio de los archivos de Matlab de este proyecto, llamadas “distorsión1.m” para la imagen de la ventana de la diana, y “distorsion2.m” para el fondo de la diana.

Previamente se habrá asignado una variable para cada imagen, con el comando “imread” para, a continuación, redimensionar la imagen hasta el doble de su tamaño, mediante el comando “imresize”, donde “2” es el factor que indica que duplique el tamaño, que es de 320x240 píxeles, hasta 640x480 con el que trabajaremos, y aplicando un filtro bicúbico para la extrapolación en el redimensionamiento. También haremos que los datos de las imágenes sean de tipo “double”, ya que es el formato de variables con el que trabajan las funciones de distorsión.

```
imagen = imread ('imagen.jpg');  
  
imagen_doble = imresize(imagen,2,'bicubic');  
  
imagen= double(imagen);
```

Posteriormente se llama a las funciones de distorsión, con los comandos:

```
imagen_rectificada_ventana = distorsion1(imagen_ventana);
imagen_rectificada_fondo = distorsion2(imagen_fondo);
```

Una vez obtenidas las imágenes sin distorsión de ambas cámaras, se deben convertir hasta conseguir una imagen con las transformaciones de giro que se aplican en la función “vision.m”, antes de realizar la medida de la distancia de los centroides que genera el láser en la diana. Para la imagen de la ventana, se realiza un giro con respecto a un eje horizontal con el comando “flipud”, mientras para la imagen del fondo de la diana se realizan dos, uno con respecto a un eje horizontal, seguido de otro con respecto a uno vertical, con los comandos “flipud” y “fliplr” respectivamente.

```
imagen_medicion_ventana = flipud(imagen_rectificada_ventana);
imagen_medicion_fondo = flipud(imagen_rectificada_fondo);
imagen_medicion_fondo = fliplr(imagen_medicion_fondo);
```

Ahora se está en situación de medir los bordes externos que delimitan nuestro área de interés en la imagen, por lo que mostraremos las imágenes transformadas con el comando “imshow” de Matlab, y usaremos la herramienta “Data cursor” en la ventana “Figure” de Matlab que nos muestra la imagen, y podremos obtener las coordenadas píxel de las cuatro esquinas.

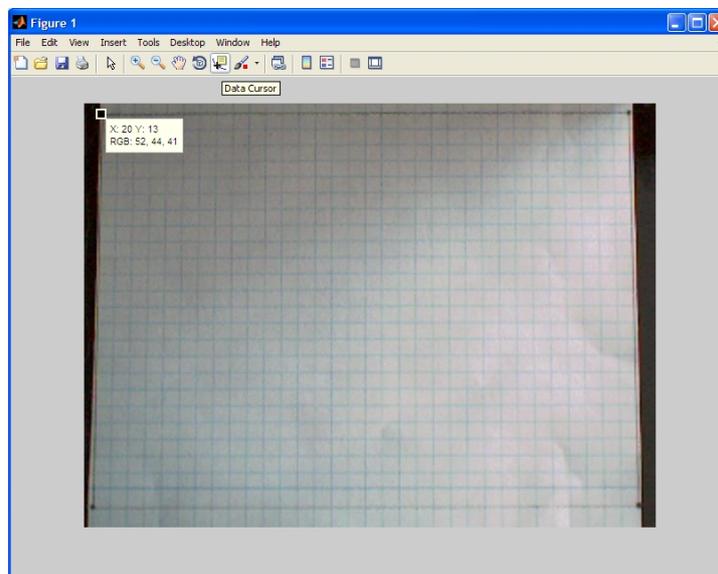


Figura 45.- Imagen de la obtención de los valores pixel de los bordes de las imágenes de las cámaras

Estos valores serán introducidos manualmente en la función “coordenadas1.m” para los datos de la imagen de la ventana, y en la función “coordenadas2.m” para el fondo, asignándoselos a las siguientes variables:

```
Coordenadas1.m:          Coordenadas2.m:  
Pc1 = [13,34];           Pc1 = [20,24];  
Pc2 = [612,32];          Pc2 = [631,21];  
Pc3 = [607,471];         Pc3 = [622,470];  
Pc4 = [21,470];          Pc4 = [27,471];
```

Por último, y como la medida en la realidad del área de interés en cada imagen es conocida, y de 160 mm para el eje X y 120 mm para el Y, se introducirán en las mismas funciones, siendo en este caso igual para ambas cámaras. Con ello se puede calcular la relación entre coordenadas pixel y coordenadas en milímetros, que es el fin de estas funciones:

```
Coordenadas1.m:          Coordenadas2.m:  
Pm1 =[0 0 1];           Pm1 =[0 0 1];  
Pm2 =[160 0 1];          Pm2 =[160 0 1];  
Pm3 =[160 120 1];        Pm3 =[160 120 1];  
Pm4 =[0 120 1];          Pm4 =[0 120 1];
```

5.3. Calibración de las Cámaras

Previo al uso de las cámaras, se debe conocer de ellas los parámetros intrínsecos de las mismas. Para ello, se ha usado una toolbox para Matlab llamada “Camera Calibration Toolbox”, la cual permite conocer estos parámetros, fundamentales a la hora de rectificar las distorsiones que se producen en las imágenes que tomamos de ellas.

Esta toolbox hace uso de las ecuaciones mediante el modelo de Zhang (ver en Bibliografía), para conocer los parámetros de calibración por medio de una plantilla patrón, y son fundamentalmente:

- Distancia focal.
- Punto principal.
- Coeficiente de apuntamiento.
- Matriz de coeficientes de distorsión radial y tangencial.

El modelo patrón para conocer estos parámetros se realiza con forma de tablero de ajedrez, con cuadros blancos y negros alternados, e intentando ocupar la mayor parte de la imagen o, al menos, la parte de la imagen de la que se requiera extraer información. Una vez colocado en una posición en el espacio de la que se quiera conocer las propiedades, se tomará una o varias imágenes para poder analizar por medio de la toolbox.

Las imágenes pueden ser en color o binarizadas a blanco y negro, pero es importante que en los bordes de la plantilla se tenga suficiente color blanco para que sea detectado como fondo de la imagen, y el programa no lo entienda como posibles bordes.

Para la ejecución de programa, se configurará la carpeta de proyecto de Matlab como la que contenga los archivos de calibración. Además, se deberá copiar los archivos de las imágenes capturadas con el patrón de calibración, para que estén accesibles para su lectura. Al mismo tiempo, es importante para la rápida obtención e identificación de las imágenes por el programa, tener todas con el mismo nombre y añadiéndole seguido un número, puesto que es el formato de nombres que reconoce la toolbox para cada calibración.

Aquí hay que hacer especial mención a que, mientras en la mayoría de las toolbox de Matlab se puede implementar las funciones que pertenecen a ella para que estén accesibles añadiendo el “path” o ruta a los archivos, debido a que esta toolbox en concreto modifica los valores de parámetros y variables, así como crea archivos de salida en la propia carpeta de trabajo, es mejor tener una copia de los archivos que comprende la toolbox por separado para cada calibración distinta que se haga.

La toolbox se ejecuta escribiendo el comando “calib” en la línea de comandos de Matlab, obteniendo la siguiente pantalla:

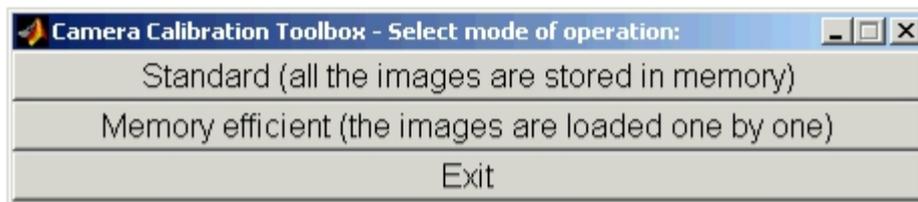


Figura 46.- Ventana de inicio de Toolbox Calibration Camera

Aquí se elegirá la opción “Standard”, normalmente para calibraciones de imágenes no excesivamente pesadas en tamaño, pudiendo usar en ese caso la opción “memory efficient” o también cuando se dispone de un ordenador con baja potencia.

A continuación el programa muestra el menú principal, desde donde se realizarán todos los comandos necesarios para la calibración:



Figura 47.- Menú principal de Toolbox Calibration Camera

El primer botón llamado “Imagen names” pedirá por línea de comandos el nombre base de las imágenes, por lo cual se mencionó que era importante

llamarlas a todas con el mismo nombre, puesto que una vez se escriba, reconocerá todas las imágenes que tenga ese nombre base seguido de un número. A continuación, pedirá una letra que asociará con el nombre de la extensión que tengan las imágenes:

```
Basename camera calibration images (without number nor suffix): imagen
Image format: (['r']='ras', 'b']='bmp', 't']='tif', 'p']='pgm', 'j']='jpg', 'm']='ppm')
```

Al seleccionarla mostrará una imagen de todas las encontradas con esas características. Además, en cualquier paso de la calibración, se puede pulsar el botón del menú principal “Read images” para recargar las imágenes tiene en memoria en ese momento.

Una vez tenemos todas las imágenes disponibles para la calibración, pasaremos a determinar los bordes que componen cada cuadrado de la plantilla, mediante el botón del menú principal “Extract grid corners”. Pedirá que indiquemos las imágenes previamente cargadas de las que se quiere hacer uso para la calibración, pulsando la tecla “intro” en caso de querer usar todas. A continuación, solicitará el tamaño en píxel del eje x e y, que formará para la detección de los puntos que forman los bordes de cada cuadrado de la plantilla de calibración. Hay que tener en cuenta el eje X es el vertical de la imagen, y el Y el horizontal como referencia para la toolbox:

```
Extraction of the grid corners on the images
Number(s) of image(s) to process ([''] = all images) =
Window size for corner finder (wintx and winty):
wintx ([''] = 5) = 7
winty ([''] = 5) = 7
Window size = 15x15
```

Luego, nos indicará si se quiere que realice el mecanismo por defecto para la detección de los cuadros de calibración, o si se quiere realizar manualmente. Por experiencia, el sistema por defecto detecta con facilidad los bordes, aun en imágenes con una deformación acusada, siempre y cuando se haya elegido un tamaño de cuadro para la detección suficientemente grande en relación al tamaño de la imagen, por lo que se pulsará la tecla “intro” para continuar:

```
Do you want to use the automatic square counting mechanism (0=['']=default)
or do you always want to enter the number of squares manually (1,other)?
```

El siguiente paso en la calibración, es marcar las esquinas exteriores de la imagen manualmente. Para ello, mostrará la imagen en la que se seleccionará con el ratón las esquinas y, automáticamente, posicionará los bordes delimitando la imagen con líneas exteriores, como se puede observar en la siguiente imagen:

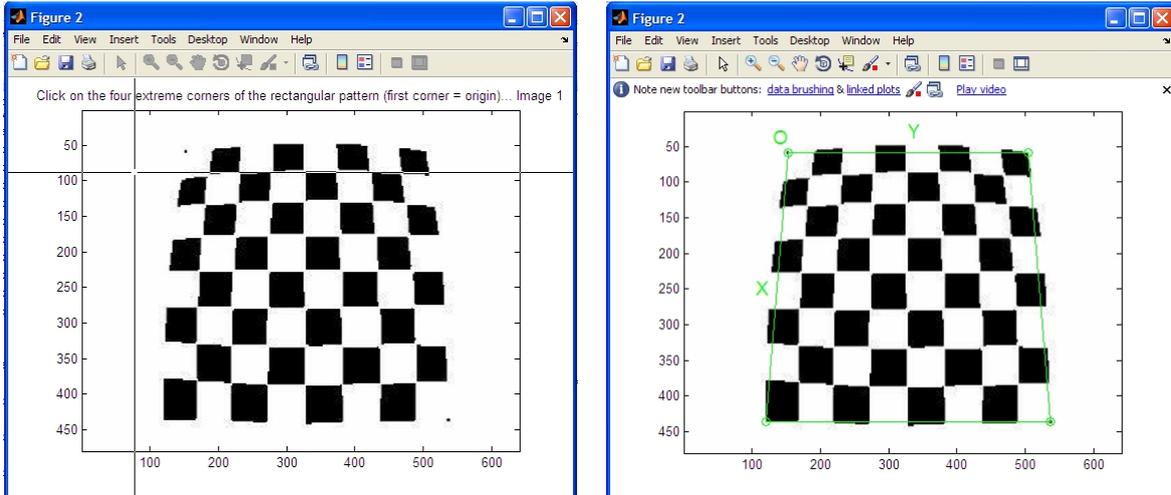


Figura 48.- Ventanas de selección de bordes de Toolbox Calibration Camera

Una vez marcadas las esquinas de la imagen, se escribirá el número de cuadros que tiene la plantilla de calibración, primero en el eje X y luego en el Y, para conocer el número de esquinas a detectar por el programa, contando como cuadro tanto los negros como los blancos. Posteriormente, el programa pedirá el tamaño en los dos ejes de cada cuadro en la realidad, medido en milímetros:

```
Number of squares along the X direction ([]) = 8
Number of squares along the Y direction ([]) = 8
Size dX of each square along the X direction ([]) = 100mm
Size dY of each square along the Y direction ([]) = 23
```

Por último, antes de obtener los parámetros de calibración, nos ofrecerá introducir un valor inicial para el factor de distorsión radial, en caso de que consideremos que por ser muy acuciada, permitirá mejorar el cálculo de la distorsión, pulsando la tecla “intro” si no se quiere añadirlo:

```
If the guessed grid corners (red crosses on the image) are not close to the actual corners,
it is necessary to enter an initial guess for the radial distortion factor kc (useful for subpixel detection)
Need of an initial guess for distortion? ([]) = no, other = yes
Corner extraction...
done
```

Con estos pasos, para todas las imágenes que se tengan, se habrá realizado la parte manual del programa, y mostrará la imagen con todas las esquinas de la plantilla de calibración marcadas. Es fundamental realizar una exploración visual comprobando que se marquen exactamente los puntos que debía, en cuyo caso se plasmará con precisión en la calibración. Si no es así, se debe volver a realizar estos pasos, pero modificando el tamaño de cuadro para la detección de esquinas, aumentándolo de valor, aunque siempre ajustándolo al

mínimo para mayor precisión, o incluso añadiendo un factor de distorsión radial inicial si no se consigue la correcta detección. Antes de volver a ejecutar el programa, es conveniente realizar la eliminación de las variables que estén en memoria de Matlab, mediante el comando en la línea de comandos “clear all”, para que el programa cargue los valores por defecto, y no asuma los que se escribieran en anteriores calibraciones.

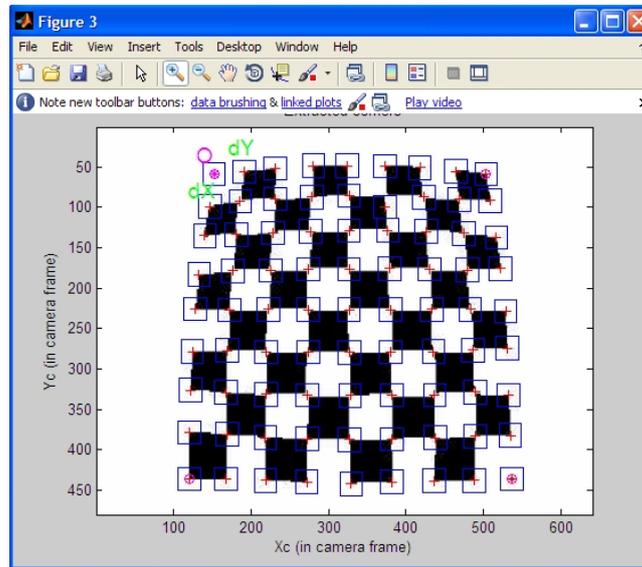


Figura 49.- Ventana de detección de cuadros en Toolbox Calibration Camera

Aunque al principio puede resultar un proceso tedioso, con unas pocas calibraciones se llegará a tener un manejo de los parámetros a definir, en función del tipo de resolución y de la distorsión, bastante rápido.

Una vez se tenga la imagen perfectamente definida, se puede pulsar el botón “Calibration”, y el programa mostrará en la línea de comandos los resultados de la calibración:

```

Calibration parameters after initialization:

Focal Length:      fc = [ 474.28657  474.28657 ]
Principal point:   cc = [ 319.50000  239.50000 ]
Skew:              alpha_c = [ 0.00000 ] => angle of pixel = 90.00000 degrees
Distortion:        kc = [ 0.00000  0.00000  0.00000  0.00000  0.00000 ]

Main calibration optimization procedure - Number of images: 1
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19.
Estimation of uncertainties...done

Calibration results after optimization (with uncertainties):

Focal Length:      fc = [ 342.46687  337.12545 ] ± [ 551.80797  550.72592 ]
Principal point:   cc = [ 319.50000  239.50000 ] ± [ 0.00000  0.00000 ]
Skew:              alpha_c = [ 0.00000 ] ± [ 0.00000 ] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:        kc = [ -0.19610  0.07596  0.01079  0.00041  0.00000 ] ± [ 0.63949  0.49741  0.02000 ]
Pixel error:       err = [ 1.18222  1.91786 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).

Recommendation: Some distortion coefficients are found equal to zero (within their uncertainties).
                To reject them from the optimization set est_dist=[0;0;1;1;0] and run Calibration

```

Además, se ofrecen diversas herramientas posteriores a la calibración como volver a mostrar los parámetros, proyectar sobre imágenes o analizar los errores, por si se quiere llevar un estudio más exhaustivo de la calibración:

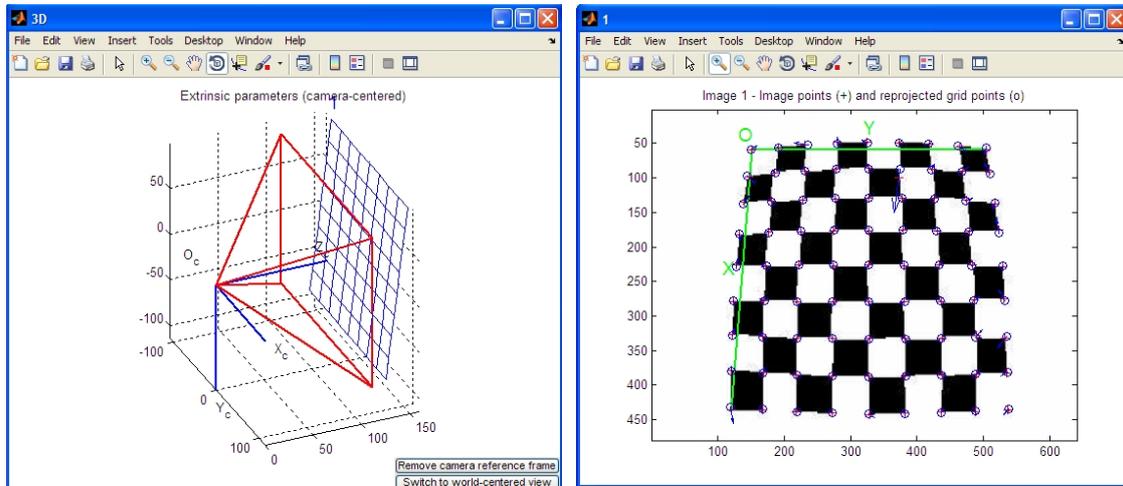


Figura 50.- Proyección de los parámetros de calibración en imágenes en Toolbox Calibration Camera

Un comando muy útil para la comprobar visualmente los efectos de la calibración, es el botón del menú “Undistort image”, el cual muestra la imagen sin distorsión aplicando los parámetros calculados, y mediante las ecuaciones por el método de Heikkilä (ver en Bibliografía). Se puede apreciar que se obtienen unos bordes en la imagen prácticamente rectos, lo que permite realizar un sistema de ecuaciones lineales que relacione las posiciones en

coordenadas píxel frente a las coordenadas en verdadera magnitud, conocidas las dimensiones externas de la imagen.

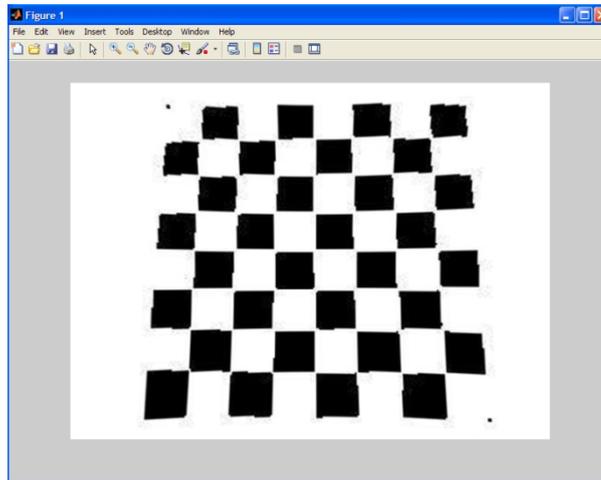


Figura 51.- Obtención de la imagen sin distorsión

Por último, se puede pulsar el botón “Save” del menú principal, para tener guardados los resultados en un archivo de Matlab llamado “calib_result.m”, el cual se ha usado en este proyecto para copiar los parámetros de la calibración de las dos cámaras, en los archivos “Distorsion1.m” y “Distorsion2.m”.

5.4. Manejo del Inclinómetro

El inclinómetro “SCA100T-D01” de la marca VTI Technologies está montado en una placa de desarrollo de Microchip modelo “PICDEM Z”, el cual incluye un microcontrolador de la misma marca y modelo “18F4620”. La principal característica que se requería para usarlo con el inclinómetro es la posibilidad de conectarlo por medio de puerto SPI, para poder hacer uso de la comunicación digital entre el micro y el inclinómetro. Además, la propia placa tiene realizadas las conexiones necesarias con un puerto serie, del cual dispone de un conector ya integrado.

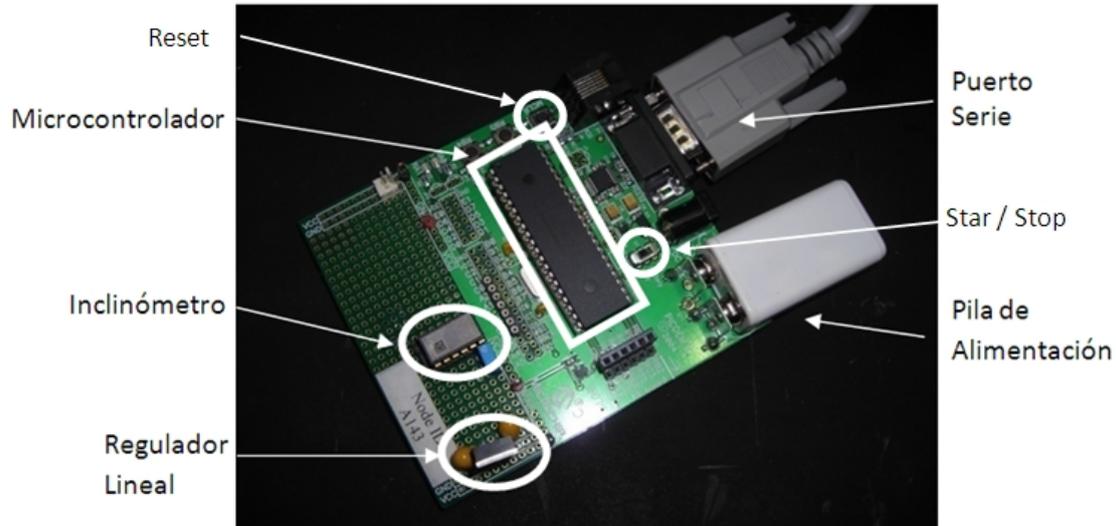


Figura 52.- Imagen de la placa de desarrollo y el inclinómetro

La placa dispone de un pulsador para reset en caso de que se necesite al notar que se obtienen valores incoherentes en la lectura del inclinómetro, por el mismo motivo anteriormente comentado, además de un conmutador de Start / Stop.

En este punto hay que comentar que la posibilidad de alimentar la placa por medio de una pila, es debido a que el modelo de la diana es un prototipo que permite una rápida apertura del mismo, pero no resulta una solución real, ya que no se considera la posibilidad de tener que realizar un mantenimiento dentro de la propia diana, al tener que ser un sistema cerrado y aislado del exterior, puesto que se encontrará en condiciones ambientales extremas. Además, no es técnicamente eficiente tener que realizar un trabajo de mantenimiento que en la mayoría de los casos sería necesario realizarlo antes de finalizar la obra de tunelización que se esté llevando a cabo.

La única condición importante a la hora de poder manejar el inclinómetro por medio del software realizado en este proyecto, es asegurar que el número de puerto serie al que es conectado el inclinómetro, coincide con el que se especifica en el archivo "diana.m", donde se realiza la inicialización y configuración del puerto:

```
PS=serial('COM6');
```

Esta comprobación puede ser llevada a cabo mediante la visualización en el panel de Control de Windows, en la tarea Sistema, pestaña Hardware y entrando en el Administrador de Dispositivos.

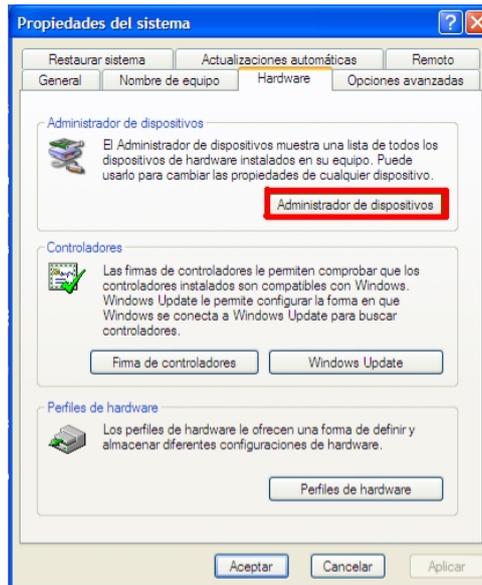


Figura 53.- Ventana de Propiedades del Sistema de Windows

En la lista desplegable de “Puertos” se informa del nombre con el que se debe conectar.

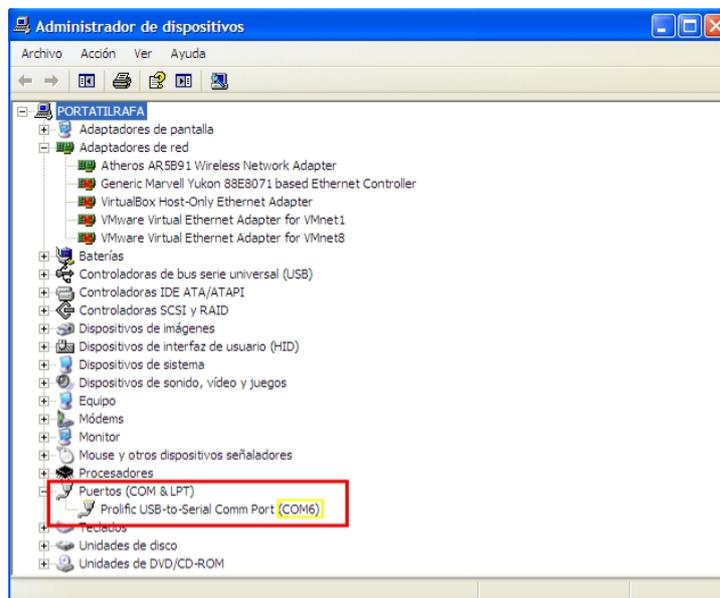


Figura 54.- Vista del Administrador de dispositivos de Windows

En el caso de que no se disponga de puerto serie en el ordenador, se puede utilizar un adaptador Serie-USB, con el cual la conexión del inclinómetro, aunque se realice por un puerto USB del que dispone cualquier ordenador actual, simulará ser un puerto serie, por lo que la configuración del mismo será igual que la explicada en el párrafo anterior.

6. Conclusiones

6.1. Análisis Crítico

Este proyecto tenía como objetivo fundamental desarrollar y crear un prototipo de diana láser para el guiado de una tuneladora.

La principal dificultad ha sido poder ver los puntos generados por el láser en el panel semitranslúcido de la ventana de la diana, y que está expuesto a la luz externa, lo que se resolvió mediante la incorporación del filtro de línea láser en la cámara que obtenía esa imagen, siendo ésta una de las principales características que modificó el diseño de la diana.

El propio uso del filtro provocó la recolocación de las cámaras en el interior de la diana, si bien antes estaban a menor distancia de los paneles de incidencia, la pérdida de visión del láser con la cámara que incorpora un filtro hizo necesario colocar las cámaras ocupando toda la longitud de la diana, e impidiendo la reducción de la misma.

El problema surgido debido a la necesidad de usar cámaras, es la distorsión en efecto de barrilete en las imágenes que toman. Al ser usadas para medir distancias, no se podía mantener esta distorsión, lo que obligó a investigar sobre un campo tan extenso y complejo como este. Gracias a la posibilidad de ampliar las capacidades de Matlab por medio de Toolbox, se consiguió resolver este problema haciendo uso de una herramienta para la calibración previa de las cámaras.

Otro problema que generó dificultades fue la realización de la comunicación entre los distintos módulos (Diana, GUI y Simulador) que forman el sistema de guiado, resuelto por medio del uso de archivos de intercambio de variables, que ha resultado ser una opción muy sencilla para su realización, no sólo por su sencillez, sino también por ser un método fácilmente integrable en un sistema ya desarrollado como era el simulador de hincas de tuberías, para el cual no hubo que realizar a penas modificaciones en la programación que ya se tenía.

Como análisis final, resaltar que la realización de este proyecto me ha supuesto una ardua tarea de investigación y desarrollo durante meses para conseguir resolver los problemas y cumplir los objetivos marcados al inicio. Sin embargo, siento que mi esfuerzo se ha visto recompensado, al considerar que he cumplido con los objetivos previstos, obteniendo un proyecto con una funcionalidad acorde a lo esperado.

6.2. Trabajos Futuros

Aunque en líneas generales se considera que en buena medida se han alcanzado los objetivos para la realización de este proyecto, quedan algunas mejoras que se pueden desarrollar en una versión futura. Son las siguientes:

- Adquisición y colocación de nuevas cámaras, con el objetivo de colocarlas a menor distancia de los paneles, reduciendo la longitud de la diana. Esto se podría conseguir con cámaras de menor distancia focal y mayor tamaño del objetivo, lo que esto último también permitiría usar un filtro de mayor diámetro y aumentaría el rango para la detección del láser. Si además de esto, contaran con mayor resolución, se evitaría tener que llevar un redimensionamiento por software del tamaño de las imágenes tomadas, salvando errores de interpolación al hacerlo.
- También el tipo de conexión por USB que tiene en la actualidad no permite realizar conexiones a gran distancia, por lo que sería conveniente la adquisición de cámaras IP, para realizar dicha conexión mediante cable de red, que sí alcanzaría el objetivo de distancia estimado de 400 metros.
- La anterior medida podría suponer una reducción de la longitud de la diana, consiguiendo un diseño más estandarizado y con mayor rango de detección de las desviaciones angulares.
- Integrar la ejecución y paro de la diana en la GUI, puesto que el software de la diana no cuenta con una interfaz de usuario que lo controle y no es posible realizar la liberación del puerto del inclinómetro, siendo las últimas líneas de la función principal "diana.m".
- Desarrollar y fabricar una placa específica para el inclinómetro, sin detrimento de las actuales características, pero mejorando la alimentación de la misma, que se debería realizar por alimentación externa en lugar de por medio de una pila, para no tener que efectuar labores de mantenimiento. Al encontrarse la diana en un medio agresivo de polvo y suciedad, debe ser un sistema cerrado y aislado, evitando tener que abrirse excepto en casos extremos. Para ello, se podría alimentar por medio del propio cable de datos.
- Una última mejora, sería la inclusión de un ordenador industrial integrado en la propia diana, que realice los cálculos sin tener que llevar los datos recogidos a un ordenador externo. La comunicación a otros módulos, como la GUI de representación, sería llevando directamente las variables

de salida del software actual. Además, realizando esto, sería muy conveniente cambiar el método actual a partir de un archivo de intercambio de variables, a conexión mediante Socket, además de poder seguir con modelos de cámaras de conexión USB, al ser una distancia de conexión muy corta.

A continuación se muestra una imagen como propuesta para las mejoras del proyecto:

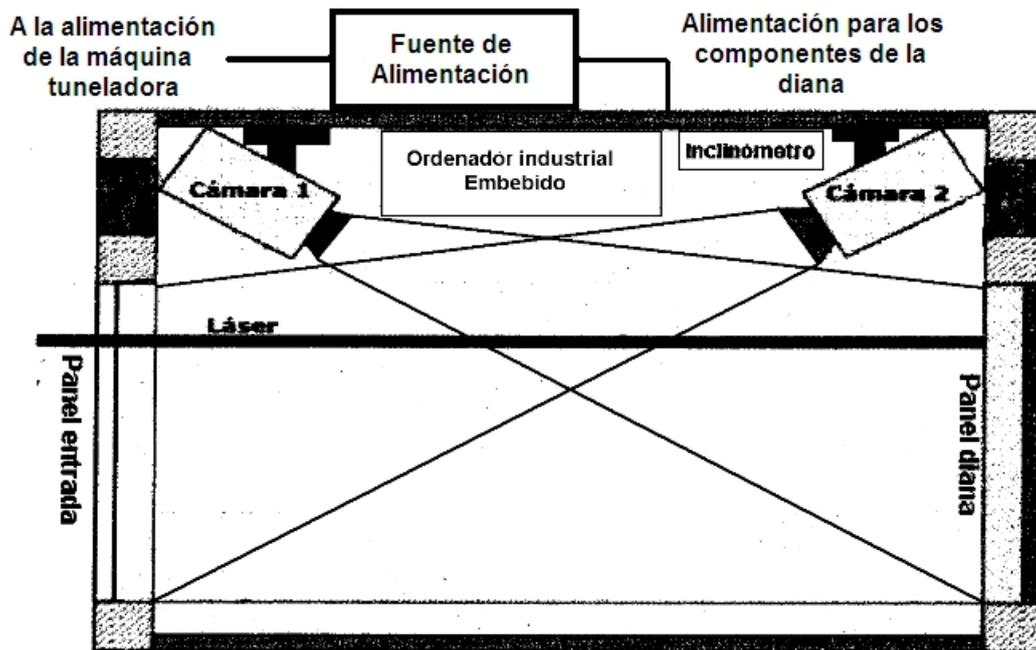


Figura 55.- Propuesta futura de rediseño de la diana

En ella se muestra un nuevo modelo de diana, el cual deberá ser construido en un material resistente y ligero, preferiblemente aluminio de color negro, que sea impermeable al polvo y la suciedad que habrá en su entorno de trabajo.

Se deberá incluir una fuente de conversión AC/DC de 220 AC y 20 W, a doble salida DC de 12 y 5 V, conectada al sistema de alimentación de la propia máquina tuneladora para alimentar a todos los dispositivos que se muestran.

Las cámaras deberán podrán tener conexión USB pero preferiblemente de resolución mayor a las actuales, hasta los 640x480 píxeles, para seguir manteniendo la precisión en los cálculos sin tener que redimensionar por software las imágenes capturadas, al ser una característica determinante a la hora de obtener la precisión en el cálculo de la posición del láser, hasta alcanzar una resolución mayor que en sistemas actuales por medio de tarjetas fotosensibles. Además, la cámara que obtenga las imágenes de la ventana de la diana, deberá contar con un filtro de línea láser de características iguales al actual. En caso de sustituir las cámaras actuales, es condición imprescindible

que tengan un perfil bajo para que ocupen lo menos posible en el área de incidencia del láser.

El inclinómetro podrá mantener los mismos componentes que los actuales ya que se ha comprobado que cumplen adecuadamente su cometido, pero sería necesario desarrollar una placa propia para reducir el espacio que ocupa al incluir un ordenador, ya que al ser actualmente una placa de desarrollo para ámbito general, no está aprovechado al máximo el espacio pudiendo eliminar las zonas para conexión y soldadura de componentes adicionales. Una mejora aconsejable sería la modificación del tipo de conexión serie actual a los más modernos y rápidos USB.

El ordenador industrial embebido deberá tener suficiente potencia para permitir la ejecución del software desarrollado en este proyecto. Además, deberá de contar con al menos 3 puertos USB para la conexión de las cámaras y el inclinómetro, y disponer de un puerto Ethernet para realizar la conexión de la diana con el ordenador base de guiado, instalado en la cabina de control. El tamaño máximo deberá ser en torno a 500x200x60 milímetros, para poder dar cabida de él dentro de la diana, dejando un espacio para el inclinómetro y las conexiones.

Por último, las conexiones con el ordenador de guiado se realizarían por medio de cable Ethernet de categoría 6 o superior y apantallado, puesto que se necesita alcanzar al menos 400 metros, los 200 metros de perforación y otros tantos para permitir llevarlo al ordenador que se encuentra a la salida del pozo de ataque. El método de comunicación que llevaría la salida del programa de la diana hasta el ordenador de guiado que ejecuta la GUI sería necesario modificarlo a comunicación por Socket, al haber sido cambiado a una comunicación IP.

6.3. Conclusiones

Al término del desarrollo de este proyecto fin de carrera se exponen las conclusiones alcanzadas.

De ésta forma, se enumeran los puntos conseguidos en este proyecto:

- Concepción y construcción de un nuevo prototipo de diana, el cual cuenta con dos paneles de incidencia láser, para la detección de un láser verde y que permite aumentar la distancia de detección hasta los 200 metros.

- Diseño de un sistema de visión integrado en la diana, que permite la detección de un rayo láser verde utilizando dos cámaras que obtienen las imágenes de los paneles de la diana.
- Integración y conexión de un inclinómetro que permite conocer el ángulo gamma producido en el eje de avance de la máquina tuneladora, imposible de medir con la diana.
- Desarrollo y programación de los software en Matlab y C que controlan todos los periféricos anteriores (cámaras e inclinómetro), realizan la detección y cálculos de los puntos de incidencia del láser y calculan la posición de la máquina tuneladora en cada instante.
- Concepción y programación de un método de comunicación por medio de archivos de variables compartidas, con el cual se comunican todo el sistema compuesto por Diana, Simulador y GUI.

En primer lugar, el sistema de visión por cámaras para la detección del haz de luz láser ha resultado ser un sistema muy satisfactorio en vista de los resultados. Se ha conseguido obtener la detección de los puntos incluso en condiciones de luz muy altas, que se podrían dar al comienzo de los trabajos de perforación, cuando la ventana de la diana esté expuesta a la luz externa. Con el uso de un filtro de línea láser para la cámara que apunta a ese panel de la ventana, el sistema se ha comportado con total fiabilidad a la hora de reconocer la posición del láser, mejorando a medida que la tuneladora avanza hacia el interior del nuevo túnel.

Como mención al uso de las cámaras, hay que destacar que la posibilidad de conexión por medio de puerto USB ha resultado ser un sistema fácilmente integrable en el software desarrollado. Además, la elección de unas cámaras de perfil bajo a resultado ser una medida acertada, al conseguir el mayor espacio posible para el área útil de incidencia del láser. En la siguiente tabla, se compara la precisión obtenida con el sistema de este proyecto, frente a dianas comerciales actuales:

	Precisión Desplazamiento Horizontal (milímetros)	Precisión Desplazamiento Vertical (milímetros)	Precisión Angular Alfa (grados)	Precisión Angular Beta (grados)	Precisión Angular Gamma (grados)
ELS1	1	1	0,05	0,05	0,05
ELS2	1	1	0,05	0,05	0,05
UC3M	0,267	0,267	0,025	0,022	0,016

Tabla 5.- Tabla de comparación de precisiones entre versiones comerciales y el prototipo del proyecto

Por otro lado, el aumento de tamaño de los paneles de incidencia para el láser en la diana supone aumentar el rango de desplazamiento para el láser, así como el rango de inclinación máximo para los ángulos que se producen. Al no suponer apenas incremento en el precio el aumento de tamaño, como sucede en las dianas que incorporan tarjetas fotosensibles, no representa mayor problema diseñar una diana de mayor tamaño, si bien es cierto que el aumento de la longitud de la diana, impide conseguir calcular ángulos de inclinación mayores, aunque según estiman fabricantes sólo es necesario llegar a ± 5 grados para los ángulos verticales y ± 15 grados para los horizontales, los cuales prácticamente cumple el prototipo realizado. La siguiente tabla recoge tanto las desviaciones verticales y horizontales máximas, como los ángulos de cabeceo e inclinación máximos que permite calcular con el prototipo de diana diseñado, los cuales son muy parecidos a los modelos actuales, pero hay que tener en cuenta el menor coste de los componentes que lo forman:

	Desplazamiento Horizontal Máximo (milímetros)	Desplazamiento Vertical Máximo (milímetros)	Ángulo Horizontal Máximo (grados)	Ángulo Vertical Máximo (grados)
ELS1	115	60	10,95	20,35
ELS2	146	88	15,84	15,84
UC3M	160	120	10	13,24

Tabla 6.- Comparación de Desviaciones y ángulos entre modelos de diana comerciales y el prototipo realizado

El modelo de inclinómetro ha resultado satisfactorio para el cálculo del ángulo gamma del sistema, puesto que la resolución del mismo de 0,016 grados es mucho menor que en los modelos comerciales de diana, que están en 0,05 grados. Además, al ser un dispositivo que mide en dos ejes, se ha configurado para medir también el ángulo alfa del sistema, por lo que se obtiene como medida de comprobación del mismo ángulo obtenido en las medidas del láser en la diana.

En cuanto al uso de archivos de intercambio de variables como método de comunicación, ha resultado ser una opción muy sencilla para su realización, no sólo por su sencillez, sino también por ser un método fácilmente integrable en un sistema ya desarrollado como era el simulador de hinca de tuberías, para el cual no hubo que realizar a penas modificaciones en la programación que ya se tenía. Las pruebas realizadas con los sistemas de Diana, Simulador y GUI dieron a conocer que se podía realizar el refresco de un nuevo dato obtenido tanto por cambios en la posición de la diana como por nuevos cálculos realizados por el simulador, representándose en la GUI en un tiempo medio estimado de 4 segundos, lo que se considera una tasa bastante buena para permitir un guiado de precisión, igualando o incluso mejorando en muchos casos a sistemas actuales, por lo que se considera alcanzado este objetivo.

BIBLIOGRAFÍA

SOFTWARE:

- [1] BOUGUET, Jean-Yves. "Camera Calibration Toolbox for Matlab"
http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
- [2] CORKE , Peter I. "Robotics Toolbox for Matlab". Abril de 2001.
http://www.bridgeport.edu/sed/fcourses/cs460_cpe460/Lectures_Handouts/1/Robotics_Toolbox_for_MATLAB_rel6.pdf

ARTÍCULOS:

- [3] ZHANG, Zhengyou. "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations". Redmond. USA. 1999.
http://www.vision.caltech.edu/bouguetj/calib_doc/papers/zhan99.pdf
- [4] HEIKKILÄ, Janne; SILVÉN, Olli. "A Four-step Camera Calibration Procedure with Implicit Image Correction". Finlandia
http://www.vision.caltech.edu/bouguetj/calib_doc/papers/heikkila97.pdf

PUBLICACIONES:

- [5] GARCÍA DE JALÓN, Javier; RODRÍGUEZ, José Ignacio. "Aprenda Matlab 7.0 como si estuviera en primero". Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid. Septiembre de 2005.
- [6] CUEVAS JIMÉNEZ, Erik V.; ZALDÍVAR NAVARRO, Daniel. "Visión por Computador utilizando MatLAB y el Toolbox de Procesamiento Digital de Imágenes".
- [7] GONZALEZ, Rafael C.; WOODS, Richard E.; EDDINS, Steven L. "Digital Image Processing Using Matlab". Pearson/Prentice Hall. 2004.
- [8] HARTLEY, Richard; ZISSERMAN, Andrew. "Multiple View Geometry in Computer Vision". Cambridge University Press. Reino Unido. 2003.
- [9] ESCALERA HUESO, Arturo de la. "Visión por computador : fundamentos y métodos". Prentice Hall. Madrid. 2001.
- [10] PLATERO, Carlos. "Apuntes de Visión Artificial". Capítulo 6. Procesamiento morfológico. Dpto. Electrónica, Automática e Informática Industrial. Universidad Politécnica de Madrid.

WEBS:

- [11] MARTÍ CARDONA, Marc. "Microtunel". Actualización 01/06/2008
http://www.microtunel.com/41_vmt_es.htm

- [12] HAUGEN, Finn. "Introduction to LabVIEW 8.5". Agosto de 2008
<http://techteach.no/labview/lv85/labview/index.htm>
- [13] The MathWorks, Inc. MATLAB (Matrix Laboratory). Actualización 2009
<http://www.mathworks.com/products/matlab/>
- [14] DIGI International. "Watchport Cameras V3"
<http://www.digi.com/>
- [15] Microchip Technology Inc. "PIC 18F4620" / "PICDEM Z".
<http://www.microchip.com/>
- [16] Edmund Optics Inc. "High Performance Laser-Line Filter 12.5 Dia"
<http://www.edmundoptics.com/>
- [17] VTI Technologies. "Inclinometer SCA100T-D01"
<http://www.vti.fi/en/products/inclinometers/>

PATENTES:

- [18] SCHNELL, Giinter. "Method and apparatus for controlling a tunneling machine". Número de patente: 4.023.861. 21 de Noviembre de 1977
- [19] ONO, Kozo; SUDA, Masao. "Method and system for controlling earth pressure in tunnel boring or shield machine". Número de Patente: 4.167.289. 11 de Septiembre de 1979
- [20] TEACH, Ted L. "Apparatus and method for detecting the position and orientation of a reference beam of light". Número de Patente: 4.786.178. 22 de Noviembre de 1988
- [21] HAYASHI, Mitsutoshi; FUJITA, Akitaka; NAGASHIMA, Shingo; FUJIMORI, Isao. "Method for controlling the direction of a shield tunnelling machine and an apparatus therefor". Número de Patente: 5.501.548. 26 de Marzo de 1996
- [22] FILIPOWSKI, Mark S.; WARDWELL, Steven R.; SHANK, Michael L. "Guidance System and Method for Keeping a tunnel boring machine continuously on a plan line". Número de Patente: 5.529.437. 13 de Septiembre de 1994
- [23] CUNNINGHAM, Peter D. "Automated Guided Apparatus Suitable for Topping Applications". Número de Patente: 6.349.249. 20 de Febrero de 2000

ANEXOS

ANEXO A. Programas Matlab

DIANA.M

```

%-----
% PROGRAMA PRINCIPAL DIANA
%-----

function diana

% Posición inicial del centro de la tuneladora
x=0;
y=0;
alfa=0;
beta=0;
gamma=0;
acceso=0; %Condición para decir a GUI que dibuje nuevo punto

%Llamada a la función de parámetros constantes definibles:
parametros;

cd('C:\Documents and Settings\Delavega\Mis
documentos\Proyecto_javier_romero\Verano(11-08-2009)')
save gui.mat x y alfa beta gamma acceso -append
cd('C:\Documents and Settings\Delavega\Mis
documentos\Proyecto_rafael\Diana_final')

z_anterior=0; %Condición para volver a ejecutar el programa

%-----
% Bucle infinito calculando los parámetros
%-----

while 1,
    pause(0.5)
    cd('C:\Documents and Settings\Delavega\Mis
documentos\Proyecto_javier_romero\Verano(11-08-2009)')
    load diana.mat z_simulada
    cd('C:\Documents and Settings\Delavega\Mis
documentos\Proyecto_rafael\Diana_final')

    if (z_simulada ~= z_anterior)

```

```
[x1,y1,x2,y2] = vision;

[alfa,beta,gamma] = angulos(x1,y1,x2,y2);

[x,y] = posicion(alfa,beta,gamma,x1,y1);

acceso=1;
z_anterior=z_simulada;

%Se guardan las cinco variables de salida en el archivo datos.mat, para
%que sean accesibles para la GUI de representación, así como el valor
%de la variable acceso, puesto a 1 para dar acceso a una nueva
%representación de la GUI, al haber calculado un nuevo punto.
cd('C:\Documents and Settings\Delavega\Mis
documentos\Proyecto_javier_romero\Verano(11-08-2009)')
save gui.mat x y alfa beta gamma acceso -append
cd('C:\Documents and Settings\Delavega\Mis
documentos\Proyecto_rafael\Diana_final')

else
    pause(0.5)

end

end

%Para cerrar puerto del inclinómetro:
fclose(PS);
delete(PS);
clear PS;
```

VISION.M

```
%-----
% Programa Visión Diana
%-----

function [x1,y1,x2,y2]=vision

global vidobj;
global vidobj2;
```

```

global bin_ventana;
global bin_fondo;

%-----
% Captura de imágenes de las cámaras
%-----

%Captura ambas imágenes
I = getsnapshot(vidobj);
I = imresize(I,2,'bicubic');
I = double(I);

R = getsnapshot(vidobj2);
R = imresize(R,2,'bicubic');
R = double(R);

%-----
% Corrección de la distorsión
%-----

Id=distorsion1(I);
Rd=distorsion2(R);

%-----
% Binarización y giros de imágenes
%-----

%Tratamiento de imágenes

Id=uint8(Id);
Rd=uint8(Rd);

I1=im2bw(Id,bin_ventana); %Binariza imagen de ventana
I2=flipud(I1);
R1 = im2bw (Rd,bin_fondo); %Binariza imagen de fondo
R1=fliplr(R1);
R1 = flipud(R1);

%-----
% Llamada a funciones para calcular los centroides
%-----

%Obtengo los puntos de láser en ventana

```

```
[x1,y1] = process1(I2);

%Obtengo los puntos de láser en fondo
[x2,y2] = process2(R1);

%-----
% Llamada a funciones para calcular los centroides
%-----

[x1,y1]=coordenadas1(x1,y1);
[x2,y2]=coordenadas2(x2,y2);
```

DISTORISON1.M

```
%-----
% Programa eliminación de distorsión
%-----

function I2= distorsion1 (I)

%-----
% Obtención de parámetros calibración e imagen
%-----

%-- Focal length:
fc = [ 9580.350546231362400 ; 13088.594222744970000 ];

%-- Principal point:
cc = [ 319.5000000000000000 ; 239.5000000000000000 ];

%-- Skew coefficient:
alpha_c = 0.0000000000000000;

%-- Distortion coefficients:
kc = [ -4.973054082144368 ; 1513.535397693685400 ; 0.069566640703949 ; -
0.006423861163292 ; 0.0000000000000000 ];

[ny,nx] = size(I);

if size(I,3)>1,
    I = 0.299 * I(:, :, 1) + 0.5870 * I(:, :, 2) + 0.114 * I(:, :, 3);
```

```

end;

if (size(I,1)>ny)||(size(I,2)>nx),
    I = I(1:ny,1:nx);
end;

%-----
% Cálculo de parámetros y obtención nueva imagen
%-----

KK = [fc(1) alpha_c*fc(1) cc(1);0 fc(2) cc(2) ; 0 0 1];

[I2] = rect_distorsion(I,eye(3),fc,cc,kc,alpha_c,KK);

```

DISTORSION2.M

```

%-----
% Programa eliminación de distorsión
%-----

function I2= distorsion2 (R)

%-----
% Obtención de parámetros calibración e imagen
%-----

%-- Focal length:
fc = [ 3463.590862306737600 ; 3488.430864885075600 ];

%-- Principal point:
cc = [ 319.500000000000000 ; 239.500000000000000 ];

%-- Skew coefficient:
alpha_c = 0.000000000000000;

%-- Distortion coefficients:
kc = [ -0.112003095961710 ; -23.690178093546802 ; 0.028556205772472 ; -
0.001682495962671 ; 0.000000000000000 ];

I=R;

```

```
[ny,nx] = size(I);

if size(I,3)>1,
    I = 0.299 * I(:, :, 1) + 0.5870 * I(:, :, 2) + 0.114 * I(:, :, 3);
end;

if (size(I,1)>ny)|| (size(I,2)>nx),
    I = I(1:ny,1:nx);
end;

%-----
% Cálculo de parámetros y obtención nueva imagen
%-----

KK = [fc(1) alpha_c*fc(1) cc(1);0 fc(2) cc(2) ; 0 0 1];

[I2] = rect_distorsion(I,eye(3),fc,cc,kc,alpha_c,KK);
```

RECT_DISTORSION.M

```
%-----
% Función rectificación distorsión
%-----

function [Irec] = rect_distorsion(I,R,f,c,k,alpha,KK)

%-----
% Cálculo de parámetros de distorsion
%-----

[nr,nc] = size(I);

Irec = 255*ones(nr,nc);

[mx,my] = meshgrid(1:nc, 1:nr);
px = reshape(mx',nc*nr,1);
py = reshape(my',nc*nr,1);

rays = inv(KK)*[(px - 1)';(py - 1)';ones(1,length(px))];
```

```
% Rotación: (o transformación afín):
rays2 = R*rays;
x = [rays2(1,:)./rays2(3,:);rays2(2,:)./rays2(3,:)];
```

```
% Añadir distorsión:
length_k = length(k);
```

```
if length_k < 5 ,
    k = [k ; zeros(5-length_k,1)];
end;
```

```
[m,n] = size(x);
```

```
% Añadir distorsión:
r2 = x(1,:).^2 + x(2,:).^2;
r4 = r2.^2;
r6 = r2.^3;
```

```
% Distorsión radial:
cdist = 1 + k(1) * r2 + k(2) * r4 + k(5) * r6;
```

```
xd1 = x .* (ones(2,1)*cdist);
```

```
% Distorsión tangencial:
```

```
a1 = 2.*x(1,:).*x(2,:);
a2 = r2 + 2*x(1,:).^2;
a3 = r2 + 2*x(2,:).^2;
```

```
delta_x = [k(3)*a1 + k(4)*a2 ;
k(3) * a3 + k(4)*a1];
```

```
xd = xd1 + delta_x;
```

```
% Reconvertir en píxeles:
px2 = f(1)*(xd(1,:)+alpha*xd(2,:))+c(1);
py2 = f(2)*xd(2,:)+c(2);
```

```
% Interpolar entre los píxeles más cercanos:
px_0 = floor(px2);
```

```
py_0 = floor(py2);

good_points = find((px_0 >= 0) & (px_0 <= (nc-2)) & (py_0 >= 0) & (py_0 <= (nr-2)));

px2 = px2(good_points);
py2 = py2(good_points);
px_0 = px_0(good_points);
py_0 = py_0(good_points);

alpha_x = px2 - px_0;
alpha_y = py2 - py_0;

a1 = (1 - alpha_y).*(1 - alpha_x);
a2 = (1 - alpha_y).*alpha_x;
a3 = alpha_y .* (1 - alpha_x);
a4 = alpha_y .* alpha_x;

ind_lu = px_0 * nr + py_0 + 1;
ind_ru = (px_0 + 1) * nr + py_0 + 1;
ind_ld = px_0 * nr + (py_0 + 1) + 1;
ind_rd = (px_0 + 1) * nr + (py_0 + 1) + 1;

ind_new = (px(good_points)-1)*nr + py(good_points);

lrec(ind_new) = a1 .* l(ind_lu) + a2 .* l(ind_ru) + a3 .* l(ind_ld) + a4 .* l(ind_rd);
```

PROCESS1.M

```
%-----  
% Procesado de la imagen de ventana de diana  
%-----
```

```
function [x1,y1] = process1(l2)
```

```
global obj_ventana;
```

```
%-----
```

```

% Transformaciones morfológicas
%-----

%Definimos la relación de vecinos entre los píxeles
SE = strel('disk',2,6);

%Opening + Closing. Suaviza contorno y rellena huecos
abierta1 = imopen(I2,SE);
cerrada1 = imclose(abierta1,SE);

%Omite objetos menores a un tamaño
imagen1 = bwareaopen(cerrada1,obj_ventana);

%-----
% Búsqueda y cálculo de centroides
%-----

%Etiquetado de objetos sin huecos. L es la matriz de etiquetas
[B,L] = bwboundaries(imagen1,'noholes');

%Busqueda de centroides en la matriz de etiquetas L:

%Calcula los centroides de objetos etiquetados en L
s = regionprops(L, 'centroid');
%Crea un array con las posiciones de los centroides
centroides1 = cat(1, s.Centroid);

x1 = centroides1(1,1);
y1 = centroides1(1,2);

```

PROCESS2.M

```

%-----
% Procesado de la imagen de fondo de diana
%-----

function [x2,y2] = process2(R1)

global obj_fondo;

```

```
%-----  
% Transformaciones morfológicas  
%-----  
  
%Definimos la relación de vecinos entre los píxeles  
SE = strel('disk',2,6);  
  
%Opening + Closing. Suaviza contorno y rellena huecos  
abierta2 = imopen(R1,SE);  
cerrada2 = imclose(abierta2,SE);  
  
%Omite objetos menores a un tamaño  
imagen2 = bwareaopen(cerrada2,obj_fondo);  
  
%-----  
% Búsqueda y cálculo de centroides  
%-----  
  
%Etiquetado de objetos sin huecos. L es la matriz de etiquetas  
[B,L] = bwboundaries(imagen2,'noholes');  
  
%Busqueda de centroides en la matriz de etiquetas L:  
  
%Calcula los centroides de objetos etiquetados en L  
s = regionprops(L, 'centroid');  
%Crea un array con las posiciones de los centroides  
centroides2 = cat(1, s.Centroid);  
  
x2 = centroides2(1,1);  
y2 = centroides2(1,2);
```

COORDENADAS1.M

```
%-----  
% Programa cálculo de coordenadas en imagen distorsionada  
%-----  
  
function [x1m,y1m]=coordenadas1(puntopx,puntopy)
```

```

%-----
% Obtención de la matriz de transformación mundo-cámara H
%-----

Pm1 =[0 0 1];
Pm2 =[160 0 1];
Pm3 =[160 120 1];
Pm4 =[0 120 1];

Pc1 = [13,34];
Pc2 = [612,32];
Pc3 = [607,471];
Pc4 = [21,470];

A=zeros(9,9);

%Punto superior izquierda
A(1,[1:3]) = Pm1;
A(2,[4:6]) = Pm1;
A(1,7) = -Pc1(1)*Pm1(1);
A(1,8) = -Pc1(1)*Pm1(2);
A(2,7) = -Pc1(2)*Pm1(1);
A(2,8) = -Pc1(2)*Pm1(2);
A([1:2],9) = -Pc1';

%Punto superior derecha
A(3,[1:3]) = Pm2;
A(4,[4:6]) = Pm2;
A(3,7) = -Pc2(1)*Pm2(1);
A(3,8) = -Pc2(1)*Pm2(2);
A(4,7) = -Pc2(2)*Pm2(1);
A(4,8) = -Pc2(2)*Pm2(2);
A([3:4],9) = -Pc2';

%Punto inferior derecha
A(5,[1:3]) = Pm3;
A(6,[4:6]) = Pm3;
A(5,7) = -Pc3(1)*Pm3(1);
A(5,8) = -Pc3(1)*Pm3(2);
A(6,7) = -Pc3(2)*Pm3(1);
A(6,8) = -Pc3(2)*Pm3(2);
A([5:6],9) = -Pc3';

```

```
%Punto inferior izquierda
```

```
A(7,[1:3]) = Pm4;  
A(8,[4:6]) = Pm4;  
A(7,7) = -Pc4(1)*Pm4(1);  
A(7,8) = -Pc4(1)*Pm4(2);  
A(8,7) = -Pc4(2)*Pm4(1);  
A(8,8) = -Pc4(2)*Pm4(2);  
A([7:8],9) = -Pc4';
```

```
%Resolvemos el sistema con autovalores
```

```
B = A'*A;  
[V,D] = eig(B);
```

```
%Creamos la matriz H
```

```
H = zeros(3);
```

```
%Llenamos la matriz con el autovector que esta relacionado con el  
%autovalor más pequeño:
```

```
H(1,:) = V([1:3],1);  
H(2,:) = V([4:6],1);  
H(3,:) = V([7:9],1);
```

```
%-----
```

```
% Cálculo de puntos mundo (en milímetros) a partir de pixel
```

```
%-----
```

```
puntom1=[puntopy,puntopx,1];  
puntom1 = inv(H) * puntom1';  
puntom1(1,1) = puntom1(1,1)/puntom1(3,1);  
puntom1(2,1) = puntom1(2,1)/puntom1(3,1);  
puntom1(3,1) = puntom1(3,1)/puntom1(3,1);
```

```
x1m=puntom1(2,1);  
y1m=puntom1(1,1);
```

```
%Se convierten los centroides referenciados a una esquina de la  
%imagen, al centro de la imagen:
```

```
if x1m <=80  
    x1m=80-x1m;  
else  
    x1m=-(x1m-80);
```

```

end

if y1m <=60
    y1m=60-y1m;
else
    y1m=-(y1m-60);
end

```

COORDENADAS2.M

```

%-----
% Programa cálculo de coordenadas en imagen distorsionada
%-----

```

```

function [x2m,y2m]=coordenadas2(puntopx,puntopy)

```

```

%-----
% Obtención de la matriz de transformación mundo-cámara H
%-----

```

```

Pm1 =[0 0 1];
Pm2 =[160 0 1];
Pm3 =[160 120 1];
Pm4 =[0 120 1];

```

```

Pc1 = [20,24];
Pc2 = [631,21];
Pc3 = [622,470];
Pc4 = [27,471];

```

```

A=zeros(9,9);

```

```

%Punto superior izquierda

```

```

A(1, 1:3 ) = Pm1;
A(2, 4:6 ) = Pm1;
A(1,7) = -Pc1(1)*Pm1(1);
A(1,8) = -Pc1(1)*Pm1(2);
A(2,7) = -Pc1(2)*Pm1(1);
A(2,8) = -Pc1(2)*Pm1(2);
A( 1:2 ,9) = -Pc1';

```

```

%Punto superior derecha

```

```
A(3, 1:3 ) = Pm2;  
A(4, 4:6 ) = Pm2;  
A(3,7) = -Pc2(1)*Pm2(1);  
A(3,8) = -Pc2(1)*Pm2(2);  
A(4,7) = -Pc2(2)*Pm2(1);  
A(4,8) = -Pc2(2)*Pm2(2);  
A( 3:4 ,9) = -Pc2';
```

%Punto inferior derecha

```
A(5, 1:3 ) = Pm3;  
A(6, 4:6 ) = Pm3;  
A(5,7) = -Pc3(1)*Pm3(1);  
A(5,8) = -Pc3(1)*Pm3(2);  
A(6,7) = -Pc3(2)*Pm3(1);  
A(6,8) = -Pc3(2)*Pm3(2);  
A( 5:6 ,9) = -Pc3';
```

%Punto inferior izquierda

```
A(7, 1:3 ) = Pm4;  
A(8, 4:6 ) = Pm4;  
A(7,7) = -Pc4(1)*Pm4(1);  
A(7,8) = -Pc4(1)*Pm4(2);  
A(8,7) = -Pc4(2)*Pm4(1);  
A(8,8) = -Pc4(2)*Pm4(2);  
A( 7:8 ,9) = -Pc4';
```

%Resolvemos el sistema con autovalores

```
B = A'*A;  
[V,D] = eig(B);
```

%Creamos la matriz H

```
H = zeros(3);
```

%Llenamos la matriz con el autovector que esta relacionado con el

%autovalor mas pequeno:

```
H(1,:) = V( 1:3 ,1);  
H(2,:) = V( 4:6 ,1);  
H(3,:) = V( 7:9 ,1);
```

%-----

% Cálculo de puntos mundo (en milímetros) a partir de pixel

```

%-----

puntop1=[puntopy,puntopx,1];
puntom1 = inv(H) * puntop1';
puntom1(1,1) = puntom1(1,1)/puntom1(3,1);
puntom1(2,1) = puntom1(2,1)/puntom1(3,1);
puntom1(3,1) = puntom1(3,1)/puntom1(3,1);

x2m=puntom1(2,1);
y2m=puntom1(1,1);

%Se convierten los centroides referenciados a una esquina de la
%imagen, al centro de la imagen:

if x2m <=80
    x2m=80-x2m;
else
    x2m=-(x2m-80);
end

if y2m <=60
    y2m=60-y2m;
else
    y2m=-(y2m-60);
end

```

ANGULOS.M

```

%-----
% Programa Ángulos Diana
%-----

function [alfa,beta,gamma] = angulos(x1,y1,x2,y2)

global gamma_cero;
global alfa_cero;

%-----
% Cálculo de alabeo, cabeceo y guiñada
%-----

close all; %cierra ventanas abiertas

```

```
%Llamada a la funcion del inclinometro
[gamma,alfa_inc]=inclinometro;

gamma=gamma-gamma_cero;
alfa_inc=-alfa_inc+alfa_cero;

distancia=680;

if gamma~=0,
    %Giro del alabeo de coordenadas de ventana
    delta1=atan2(y1,x1);
    d= sqrt((y1^2)+(x1^2));
    x1g= d*cos(gamma+delta1);
    y1g= d*sin(gamma+delta1);

    %Giro del alabeo de coordenadas de fondo
    delta2=atan2(y2,x2);
    d= sqrt((y2^2)+(x2^2));
    x2g= d*cos(gamma+delta2);
    y2g= d*sin(gamma+delta2);

else
    x1g=x1;
    y1g=y1;
    x2g=x2;
    y2g=y2;
end;

%Giro en X (se genera un ángulo alfa):
dy=(y2g-y1g);
alfa=atan2(dy,distancia)

%Giro en Y (se genera un ángulo beta):
dx=(x2g-x1g);
beta=atan2(dx,distancia)

%Se comprueba que los dos ángulos alfa obtenidos no varían en más de 0.5
%grados:
if (alfa>=alfa_inc+0.0087 || alfa<=alfa_inc-0.0087)
    %AVISO DE ERROR DE IMPRECISIÓN EN ÁNGULO ALFA:
    mensaje=sprintf('WARNING: Incoherencia en la redundancia de ángulo alfa');
```

```

disp(mensaje);
fprintf(1,'\n');
end

```

INCLINOMETRO.M

```

function [gamma, alfa_inc]=inclinometro

global PS;

fprintf(PS,'%c','A'); %Petición de datos de gamma

%Lectura de datos del inclinómetro
dato=fread(PS);
dato1=char(dato);
ang1=str2double(dato1);

%Cálculo del ángulo:
gamma=asin((ang1-1024)/1638);

pause(0.05)

fprintf(PS,'%c','B'); %Petición de datos de alfa

%Lectura de datos del inclinómetro
dato=fread(PS);
dato1=char(dato);
ang1=str2double(dato1);

%Cálculo del ángulo:
alfa_inc=asin((ang1-1024)/1638);

```

POSICION.M

```

%-----
% Programa Posición Tuneladora
%-----

function [x,y] = posicion(alfa,beta,gamma,x1,y1)

```

```
%-----  
% Cálculo nueva posición del centro de la tuneladora  
%-----  
  
global z_diana;  
  
cd('C:\Documents and Settings\Delavega\Mis  
documentos\Proyecto_javier_romero\Verano(11-08-2009)')  
load diana.mat z_simulada  
cd('C:\Documents and Settings\Delavega\Mis  
documentos\Proyecto_rafael\Diana_final')  
  
T = transl(0,0,z_simulada); %Matriz homogénea de traslación  
G = rpy2tr(gamma,beta,alfa); %Matriz homogénea de rotación  
MT = G*T; %Matriz Taslación + Rotación homogénea  
  
z=z_diana;  
x=x1;  
y=y1;  
  
nuevapos=MT*[x;y;z;1]; %Matriz homogénea por coord. de referencia  
  
x=nuevapos(1,1);  
y=nuevapos(2,1);
```

PARAMETROS.M

```
%-----  
% PROGRAMA DE PARÁMETROS DEFINIBLES  
%-----  
  
function parametros  
  
%Parámetros de binarización de ventana y fondo respectivamente:  
global bin_ventana;  
global bin_fondo;  
  
bin_ventana=0.65;  
bin_fondo=0.50;  
  
%Parámetros de binarización de ventana y fondo respectivamente:
```

```
global obj_ventana;  
global obj_fondo;
```

```
obj_ventana=100;  
obj_fondo=50;
```

```
%Parámetro de colocación de la diana en el interior de la tuneladora:
```

```
global z_diana;
```

```
z_diana=-1728;
```

```
%Configuración y apertura del puerto para el inclinómetro
```

```
global PS;
```

```
PS=serial('COM3');  
set(PS,'Baudrate',9600); %se configura la velocidad  
set(PS,'StopBits',1); %se configura bit de parada a uno  
set(PS,'DataBits',8); %se configura que el dato es de 8 bits  
set(PS,'Parity','none'); %se configura sin paridad  
set(PS,'Terminator','CR/LF'); %caracter con que finaliza el envío  
set(PS,'OutputBufferSize',1); %número de bytes a enviar  
set(PS,'InputBufferSize',4); %número de bytes a recibir  
set(PS,'Timeout',0.15); %tiempo de espera  
fopen(PS);
```

```
%Llamada para obtener los valores en el origen de los ángulos:
```

```
global gamma_cero;
```

```
global alfa_cero;
```

```
[gamma_cero, alfa_cero]=inclinometro;
```

```
%Se asigna una variable para cada cámara:
```

```
global vidobj;
```

```
global vidobj2;
```

```
vidobj = videoinput('winvideo', 2);
```

```
vidobj2 = videoinput('winvideo', 1);
```

ANEXO B. Programa Inclinómetro

MAIN.C

```
#include <p18f4620.h>
#include <stdio.h>
#include <usart.h>
#include <spi.h>

#pragma config OSC = XT
#pragma config WDT = OFF

unsigned char DoutH, DoutL, control = 0;
unsigned int i, Dout = 0;

void main()
{

    // Configurar el puerto SPI
    OpenSPI(SPI_FOSC_64, MODE_00, SMPMID);
    TRISCbits.TRISC0 = 0;
    PORTCbits.RC0 = 1; //Chip select

    // inicializar puerto USART para 9600 baudios
    OpenUSART (USART_TX_INT_ON &
        USART_RX_INT_ON &
        USART_ASYNC_MODE &
        USART_EIGHT_BIT &
        USART_CONT_RX &
        USART_BRGH_HIGH, 25); //[4000000/(9600*16)] - 1 = 25

    // Configurar interrupcion del USART como alta prioridad.
    IPR1bits.RCIP = 1;
    IPR1bits.TXIP = 1;

    while (1)
    {
        //Lectura del puerto serie para obtener función a realizar
        getsUSART((unsigned char *)&control, 1 );

        for (i = 0; i <= 5000; i++);

        if (control == 'A') // Lectura de alfa
        {
            Dout = 0;
            PORTCbits.RC0 = 0; // Chip Select
        }
    }
}
```

```

WriteSPI( 16 );
getsSPI((unsigned char *)&DoutH, 1 );
getsSPI((unsigned char *)&DoutL, 1 );
PORTCbits.RC0 = 1;
Dout = DoutH;
Dout = Dout << 8;
Dout |= DoutL;
Dout = Dout >> 5;

printf ("%04d", Dout);
for (i = 0; i <= 5000; i++);

//Se puede añadir petición de datos de temperatura:
/*
Dout = 0;
PORTCbits.RC0 = 1; //Activo a nivel alto
WriteSPI( 8 );
getsSPI((unsigned char *)&DoutH, 1 );
getsSPI((unsigned char *)&DoutL, 1 );
Dout = DoutH;
Dout = Dout << 8;
Dout |= DoutL;
Dout = Dout >> 5;

printf ("%d", Dout);
for (i = 0; i <= 5000; i++);
*/

}

else if (control == 'B' ) //Lectura de gamma
{
Dout = 0;
PORTCbits.RC0 = 0;
WriteSPI( 17 );
getsSPI((unsigned char *)&DoutH, 1 );
getsSPI((unsigned char *)&DoutL, 1 );
PORTCbits.RC0 = 1;
Dout = DoutH;
Dout = Dout << 8;
Dout |= DoutL;
Dout = Dout >> 5;

printf ("%04d", Dout);
for (i = 0; i <= 5000; i++);

//Se puede añadir petición de datos de temperatura:
/*
Dout = 0;

```

```
        PORTCbits.RC0 = 1; //Activo a nivel alto
        WriteSPI( 8 );
        getsSPI((unsigned char *)&DoutH, 1 );
        getsSPI((unsigned char *)&DoutL, 1 );
        Dout = DoutH;
        Dout = Dout << 8;
        Dout |= DoutL;
        Dout = Dout >> 5;

        printf ("%d", Dout);
        for (i = 0; i <= 5000; i++);
        */
    }

    /*else if (control == 'C' )
    {
//Código para self test, pero no es necesario al tener Self Test y Failure
//Detection Mode automático:
//The self test function must not be activated for both channels at the
//same time.

        PORTCbits.RC0 = 0;
        WriteSPI( 14 ); //ON command selftest eje X
        PORTCbits.RC0 = 1;

        for (i = 0; i <= 5000; i++);

        PORTCbits.RC0 = 0;
        WriteSPI( 0 ); //OFF command selftest
        PORTCbits.RC0 = 1;

        for (i = 0; i <= 5000; i++);

        PORTCbits.RC0 = 0;
        WriteSPI( 15 ); //ON command selftest eje Y
        PORTCbits.RC0 = 1;

        for (i = 0; i <= 5000; i++);

        PORTCbits.RC0 = 0;
        WriteSPI( 0 ); //OFF command selftest
        PORTCbits.RC0 = 1;

        for (i = 0; i <= 5000; i++);
    }
    */
    control = 'D';
}
}
```

ANEXO C. Cámaras

Watchport®

**USB ENVIRONMENTAL
MONITORING
APPLIANCES**



**Installation
Guide**

Sensor Models:

Watchport/H
Watchport/P
Watchport/T
Watchport/W
Watchport/D
Watchport/A

Camera Models:

Watchport/VE
Watchport/V2
Watchport/V3



www.digi.com

Installing Watchport Camera Drivers

NOTE: If any Watchport Camera drivers are already installed on your system, you MUST uninstall them before continuing with these installation instructions. Also, the two sets of drivers (one for DirectShow and one for TWAIN and AVI) are mutually exclusive; you may install one **OR** the other, but not both.

If you are using DirectShow (DirectX), follow the instructions below. If you are using TWAIN or AVI, go to "TWAIN and AVI Drivers" on page 3.

DirectShow (DirectX) Drivers

You may want to check that you have the most recent version of DirectX before continuing with these installation instructions.

For Windows XP, 2000, and 2003 Server Users

Drivers for DirectShow (DirectX) are on the "DirectShow Driver and Manual" CD. Note that you must be logged into an account with administrator privileges.

- 1) Insert the "DirectShow Driver and Manual" CD version 2.90 or above into your CD-ROM drive.
- 2) Plug in the Watchport Camera. (See "Cabling Watchport" on page 6).
- 3) When the Found New Hardware screen for the Watchport USB Loader software appears, choose option a) or option b):
 - a) Specify the location of the drivers and click **Next**.
 - b) Install from a list or specific location and click **Next**. The subsequent screen allows you to search removable media, type the location path, or browse to locate the drivers on your CD-ROM drive. Use one of these methods to locate the drivers and click **Next**.
- 4) Windows may warn that the software "has not passed Windows Logo testing." Click **Continue Anyway** to continue with driver installation.
- 5) When the Found New Hardware Wizard finishes installing the Watchport USB Loader software, click **Finish**.
- 6) Wait for the second Found New Hardware screen (for the Watchport Camera software) to appear. When it does, again choose option a) or option b) as described in step 3.
- 7) Windows may warn that the software "has not passed Windows Logo testing." Click **Continue Anyway** to continue with driver installation.
- 8) When the Found New Hardware Wizard finishes installing the Watchport USB Camera software, click **Finish**.

Your Watchport Camera is now ready to use. To verify installation you may start the Watchport Viewer (found in **Start/Programs/Watchport imaging/Watchport Viewer**).

The Watchport Viewer defaults to activating the camera (in **Options/Preview**). You can open a Properties screen through **Options/Video Capture Filter** and adjust brightness, sharpness, contrast, color, compression, etc.

Cabling Watchport



USB Type A Connector



USB Type B Connector

Watchport Camera Products

To connect a Watchport Camera, plug the Type A (flat) end of the USB cable into the USB port located in the back of your PC or into an available USB port on a standard hub or into a Digi Hubport. Plug the Type B (rounded) end of the USB cable into the Watchport.

The Watchport/VE Camera is additionally connected using an RCA video cable or an S-Video cable. When using an RCA video cable, match the composite video (yellow), right audio (red), and left audio (white) connectors with the corresponding receptacles on your video device. When using a standard S-Video cable, match the S-Video (black), right audio (red), and left audio (white) with the corresponding S-Video and audio connectors on your video device.

The software default is "Composite Video." To select S-Video, open the Watchport Viewer (**Start / Programs / Watchport Imaging / Watchport Viewer**). While in the Watchport Viewer, go to **View / Options / Video Capture Filter** and select the **Advanced** tab. In the **Video Source** box, select **S-Video** and click **OK**.

Watchport Sensor Products

The sensing device of the Watchport Sensor Products comes with an RJ45 terminated cable. Plug the RJ45 connector into the corresponding jack on the Watchport base unit.

Plug the Type A (flat) end of the USB cable into the USB port located in the back of your PC or into an available USB port on a standard hub or into a Digi Hubport. Plug the Type B (rounded) end of the USB cable into the Watchport base unit.

Extending the Cable

Watchport Sensor products, which are shipped with a 6 foot cable, can be extended using a CAT-5 twisted pair cable up to 500 feet for the Watchport/A, Watchport/H, Watchport/P, Watchport/D, and Watchport/T and up to 1000 feet for the Watchport/W. Contact Digi International if you need to extend it beyond these limits.

To extend your cable, we recommend that you use CAT-5 twisted pair cable and follow the wiring in Table 1. Your color assignment may vary and you will need to adjust the tables accordingly. Connect only the actively used signals as indicated in the picture to the right.

Proximity Sensor		All Other Sensors	
RJ45	Wire Color	RJ45	Wire Color
3	Black	3	Yellow
6	Blue	4	Green
7	Red	5	Black
		8	Grey

Wiring Table 1: Extending the Cable



Base unit, RJ45 connector, CAT-5 twisted pair cable

Shortening the Cable

If you need to shorten the cable, cut off the RJ45 connector and crimp a new connector at the end following the wiring in Table 2 (for all sensors).

RJ45	Wire Color	RJ45	Wire Color
1	White	5	Black
2	Brown	6	Blue
3	Yellow	7	Red
4	Green	8	Grey

Wiring Table 2: Shortening the Cable

Installing Watchport Manager

The Watchport Manager allows you to monitor your Watchport Camera and Sensor readings. To install the Watchport Manager:

1. Use the browser to locate the Watchport Manager folder located on the Edgeport CD and then double-click on **400002560.exe**.
2. Click **OK**.
3. Click on the large button to install Watchport Manager to the specified directory.
4. Click **Continue** to add items to the Watchport Manager program group.
5. Wait while files are copied and the system is updated. If prompted to keep newer existing files while installing, click **Yes**.
6. Click **OK** to complete installation.

You can open the Watchport Manager from the **Start/Programs** menu.

Updating Base Unit Firmware

This section applies to Watchport Sensor base units only (not Watchport Camera products).

1. Connect the Watchport base unit to your PC. Refer to "Cabling Watchport" on page 6 for information on connecting your Watchport base unit.
2. Open the Edgeport configuration utility program, found in the **Start/Programs** menu.
3. In the General tab, click on **Update**.
4. The **Open** button allows you to browse for the Watchport firmware (*.rom). The Edgeport Utility program can detect if the firmware is the same version, an older version, or for a different sensor.
5. After clicking **Update**, you will always be asked to confirm that you want to update your firmware. **NOTE: Do NOT disconnect your Watchport until the update is complete!**
6. After clicking **OK**, the update is complete.

See "The Edgeport Utility Program" on page 16 for more information about using this feature.

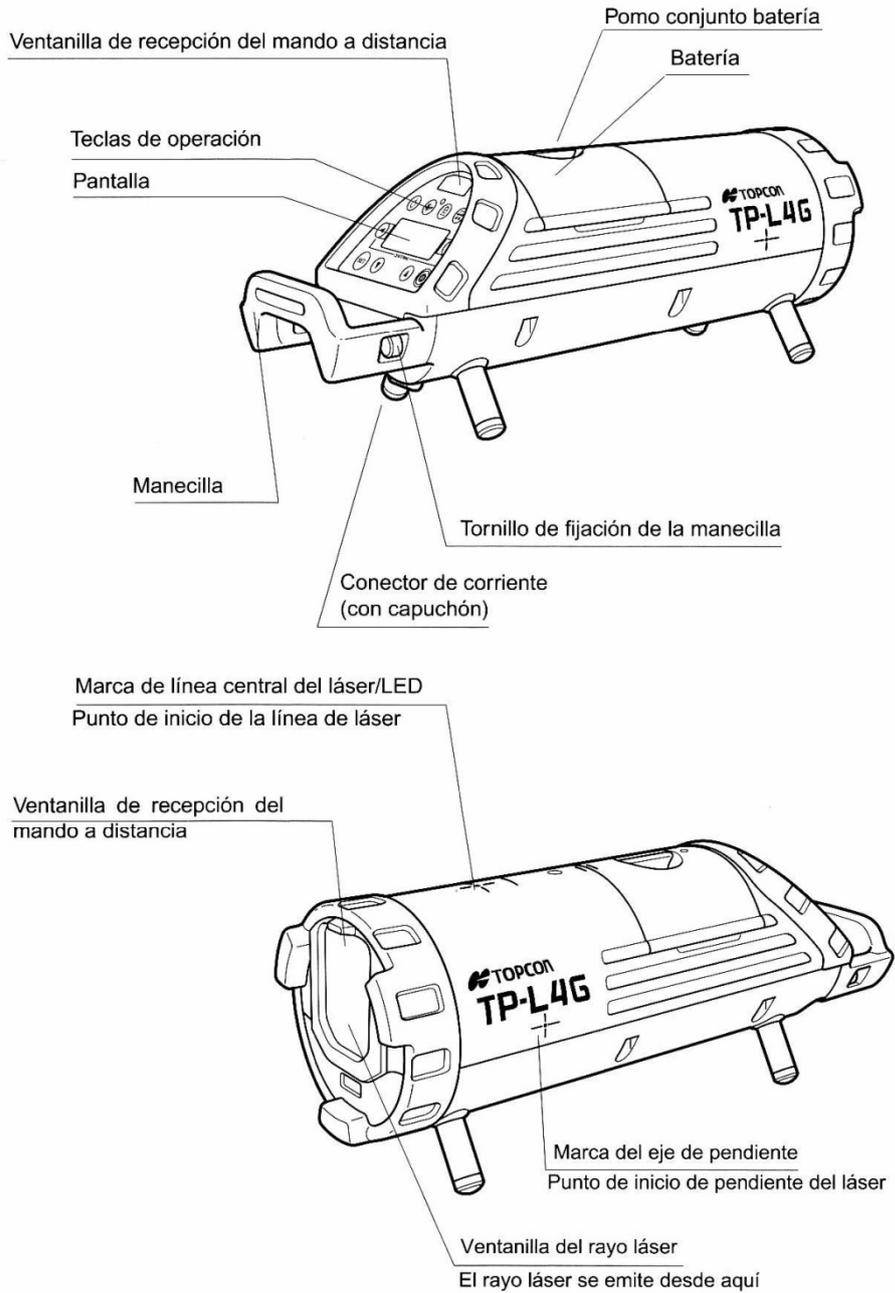
Features/Specifications

FEATURES	FEATURES (CONTINUED)	DIMENSIONS
<p>Watchport/V2 Watchport/V3</p> <ul style="list-style-type: none"> Ideal for photo ID badging, kiosks, ATMs, access control and general surveillance Only USB camera of its kind with Windows NT® drivers Advanced facial recognition software for password-free access security Motion detection software available to create cost-effective remote monitoring system Low light sensitivity: <1 lux Adjustable resolutions <ul style="list-style-type: none"> 1280 x 960 (software enhanced) 640 x 480 352 x 288 320 x 240 176 x 144 160 x 120 128 x 96 Maximum frame rates by resolution <ul style="list-style-type: none"> 30 fps at all resolutions Removable 4.9 mm lens; F:2.0 adjustable Non-interlaced model for even higher quality pictures Optional lens pack available <ul style="list-style-type: none"> 3 mm wide angle (green) 8 mm close-up (red) 12 mm telephoto (blue) Optional mounts <ul style="list-style-type: none"> Wall mount Clamp mount Exposure controls <ul style="list-style-type: none"> Manual and automatic gain and shutter control Automatic White Balance (AWB) Manual color balance Color saturation Color formats: 16.8 million TrueColor (24 bit RGB), YUY2, UYVY Motion JPEG compression Sensor: color CCD Scan frequencies: 60, 50, 30, 25, 15, 12.5, 7.5, 6.25 and 4 Hz Shutter rate controls: 1/4 second to over 1/30,000 second 	<p>Watchport Sentry Watchport Console</p> <ul style="list-style-type: none"> Automatic detection of motion event in camera field of vision Capture of specific object in motion Pre-event and post-event recording prevents missing event boundaries Sensitivity controls and user-defined masks ensure shadows, clouds and other unwanted motion are not recorded as events Scheduling of operation reduces image storage requirements Sophisticated alerting functions can send captured images by email, or upload them to a website for remote viewing SMS message alerts can also be configured via an external application <p>Watchport/FastAccess</p> <ul style="list-style-type: none"> Instant facial recognition Automatic login without typing Automatic logoff when user steps away Seamless account switch for new users Eliminates password theft Centrally managed Easily deployed 	<p>Watchport/V2 Watchport/V3</p> <ul style="list-style-type: none"> Length: 3.5 in (9.0 cm) Width: 2.5 in (6.4 cm) Height: 1.5 in (3.8 cm) Weight: 4.5 oz (127.6 g) <p>Watchport/VE</p> <ul style="list-style-type: none"> Length: 3.2 in (8.0 cm) - not including cables Width: 1.8 in (4.4 cm) Height: 0.8 in (2.0 cm) Weight: 1.9 oz (54.0 g) Cable lengths: 4.4 in (11.2 cm)
<p>Watchport/VE</p> <ul style="list-style-type: none"> Video encoder to connect conventional 3rd party cameras via USB Converts NTSC video streams Supports composite and S-type video connections Supports external mono or stereo audio devices with dual RCA connectors Microsoft DirectShow WDM driver Compatible with Watchport Vision Series software products 	<p>POWER REQUIREMENTS</p> <ul style="list-style-type: none"> USB-powered (no external power supply required) 	<p>REGULATORY APPROVALS</p> <ul style="list-style-type: none"> FCC Part 15, Class B CE EN55022 EN55024 EN60950 UL 1950 CSA 2.2 No. 950 IEC-950 <p>STANDARDS AND PROTOCOLS</p> <ul style="list-style-type: none"> Microsoft DirectShow WDM Microsoft Video for Windows (except Watchport/VE) TWAIN (except Watchport/VE)
	<p>ENVIRONMENTAL</p> <ul style="list-style-type: none"> Operating temperature: 32° F to 149° F (0° C to 65° C) Relative humidity: 0% to 95% (non-condensing) 	
	<p>OPERATING SYSTEMS</p> <ul style="list-style-type: none"> Windows® XP Windows XP Embedded Windows NT 4.0, Windows NT Embedded Windows 2000/Me Windows Server™ 2003 	

ANEXO D. Láser

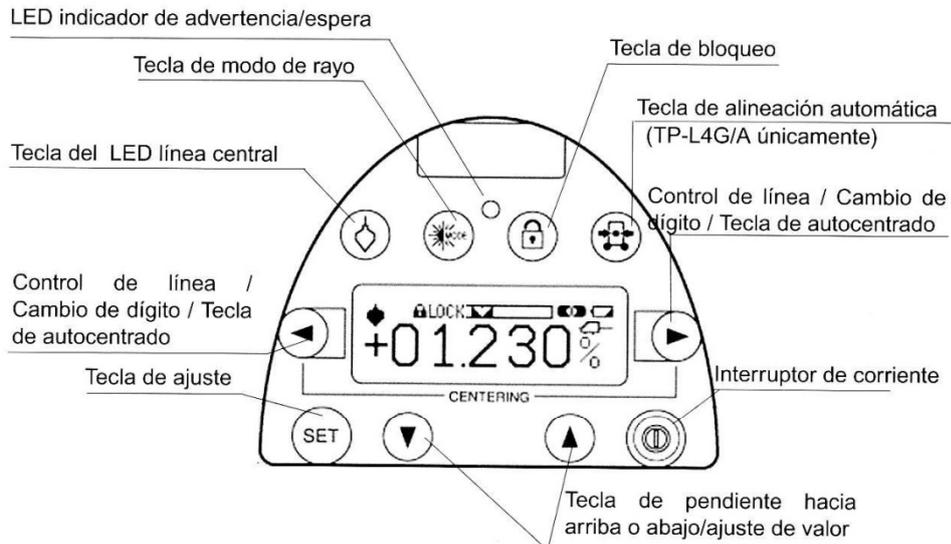
1 NOMENCLATURA Y FUNCIONES

1.1 Nomenclatura



1 NOMENCLATURA Y FUNCIONES

Panel de operación



*La iluminación de la pantalla permanecerá durante 30 segundos siempre que se pulse alguna tecla

Tecla	Nombre	Funciones
	Tecla del LED línea central	Encienda (ON) o apague (OFF) el LED de línea central presionando. (Auto apagado tras 30 minutos.)
	Tecla de modo de rayo	El modo del rayo láser se conmuta de forma alternativa. (corriente estándar, alta corriente)
	Tecla de bloqueo	Esta tecla impide la entrada de datos de las teclas de control de pendiente. Cuando se presiona de nuevo, se libera el bloqueo.
	Tecla de alineación automática	El rayo láser se alinea automáticamente con el centro del objetivo.
	Control de línea/Cambio de dígito/ Tecla de autocentrado	Presione la tecla para mover el rayo láser hacia la derecha o hacia la izquierda. El dígito indicador activo cambia a derecha o a izquierda. Presionando ambas teclas al mismo tiempo, el rayo láser vuelve al centro automáticamente.
	Tecla de ajuste de pendiente arriba o abajo/ Tecla de ajuste de valor	Presiónela para mover el rayo láser arriba o abajo. Los valores positivos o negativos del ajuste de pendiente se indican en la pantalla. Presionando ambas teclas al mismo tiempo, se retorna la pendiente a 00.000%.
	Tecla de ajuste	Presione la tecla para introducir la pendiente mostrada en pantalla.
	Interruptor de corriente	Encender o apagar el instrumento.
	LED indicador de advertencia/espera	Parpadea indicando el modo de advertencia o espera.

1 NOMENCLATURA Y FUNCIONES

Indicadores

Frasco de nivel digital, gire el instrumento hasta que se centre

Indicador de línea del láser
Indica la dirección de la línea del láser

LED indicador de línea central

Indicador de advertencia de la batería
La capacidad que queda de carga de la batería se muestra en 3 pasos

Indicación de pendiente
Símbolo de porcentaje o de por mil.

Indicador de bloqueo
Indica que el instrumento está bloqueado

Indicación modo del láser

Indicación de autonivelación (parpadeando)
Parpadea cuando el instrumento está autonivelado. Después de finalizada la autonivelación, la pantalla cambia el modo láser

Intermitencia

Cargando mientras se utiliza Batería externa de 12 V CC

El tiempo de operación está limitado
Se precisará recargar u otra fuente de corriente pronto

Listo para funcionar

Modo ON

Modo Intermitencia

Modo Alta Potencia (TP-L4G/BG únicamente)

La pantalla del frasco de nivel muestra el estado de rotación del instrumento
 Cuando el instrumento está inclinado hacia los laterales, el indicador del frasco de nivel pequeño de la pantalla se agranda cubriendo toda la pantalla para ayudar al usuario a ajustar con precisión el instrumento. (consulte la sección 6, Ajuste de los modos de operación, para más información).

Indicaciones de advertencia

Indicador de advertencia de la batería
 La operación es imposible, no emitiéndose el rayo láser. Cambie o recargue la batería.

Indicación de advertencia de nivel
 El instrumento inclinado más allá de los tramos de nivelación hacia atrás o hacia delante. Coloque de nuevo el instrumento inclinandolo en la dirección indicada por la flecha..

Indicación de advertencia sobre giro
 El instrumento está inclinado demasiado hacia la derecha o hacia la izquierda. Vuelva a colocar el instrumento en la dirección indicada por la flecha.

11 ESPECIFICACIONES

11 ESPECIFICACIONES

TP-L4G/A/B/BG

Fuente de luz	:	Diodo de láser visible
Longitud de onda	:	532nm (TP-L4G/BG, verde) 633nm (TP-L4A/B, rojo)
Potencia del láser	:	2.8mW(Max.)
Diámetro del láser	:	ø12mm
Ancho del control de línea	:	±15% (±15 pies y 100 pies de distancia)
Lectura de pendiente	:	-15% ~ +40% (-150 ‰ ~ +400 ‰)
Ajuste mínimo de pendiente	:	0.001%
Método de ajuste de pendiente	:	Absoluto codificador
Gama de autonivelación	:	
Dirección de la gradiente	:	± 10%
Dirección del eje	:	± 4°
Precisión horizontal	:	± 10 arco segundos
Distancia de alineación automática	:	5m ~150 m (TP-L4G/A)
Tiempo de funcionamiento [+20°C]	:	
TP-L4G/BG	:	DB-53: Aproximadamente 45 horas (Uso de baterías secas alcalinas de manganeso) BT-53Q: Aprox.32 horas
TP-L4A/B	:	DB-53: Aprox.70 horas (Uso de baterías secas alcalinas de manganeso) BT-53Q: Aprox.48 horas
Temperatura de funcionamiento	:	-20°C ~ +50°C
Dimensiones	:	ø122mm ~ 330mm (sin mango) ø125mm ~ 374mm (con mango trasero)
Peso	:	Aprox. 3,8 kg

Mando la distancia, modelo RC-200

Distancia de operación	:	A través del tubo desde la parte delantera, aprox.200m
Funciones	:	Modo de control de línea, ENCENDIDO/ APAGADO del rayo láser, Rayo Láser, Modo de alineación automático, centrado de línea
Alimentación de corriente	:	Cuatro baterías secas tamaño AAA
Tiempo de funcionamiento	:	Aprox. 8 meses (Uso de baterías secas alcalinas de manganeso)

EURHINCA
Europea de Hincas Teledirigidas, S.A.
Avda. Finanzauto, 29
28500 ARGANDA DEL REY
(Madrid)



EMC NOTICE

In industrial locations or in proximity to industrial power installations, this instrument might be affected by electromagnetic noise. Under such conditions, please test the instrument performance before use.

ANEXO E. Inclinómetro



SCA100T Series

Data Sheet



THE SCA100T DUAL AXIS INCLINOMETER SERIES

The SCA100T Series is a 3D-MEMS-based dual axis inclinometer family that provides instrumentation grade performance for leveling applications. The measuring axes of the sensing elements are parallel to the mounting plane and orthogonal to each other. Low temperature dependency, high resolution and low noise, together with a robust sensing element design, make the SCA100T the ideal choice for leveling instruments. The VTI inclinometers are insensitive to vibration, due to their over damped sensing elements, and can withstand mechanical shocks of up to 20000 g.

Features

- Dual axis inclination measurement (X and Y)
- Measuring ranges $\pm 30^\circ$ SCA100T-D01 and $\pm 90^\circ$ SCA100T-D02
- 0.0025° resolution (10 Hz BW, analog output)
- Sensing element controlled over damped frequency response (-3dB 18Hz)
- Robust design, high shock durability (20000g)
- High stability over temperature and time
- Single +5 V supply
- Ratiometric analog voltage outputs
- Digital SPI inclination and temperature output
- Comprehensive failure detection features
 - True self test by deflecting the sensing elements' proof mass by electrostatic force.
 - Continuous sensing element interconnection failure check.
 - Continuous memory parity check.
- RoHS compliant
- Compatible with Pb-free reflow solder process

Applications

- Platform leveling and stabilization
- 360° vertical orientation measurement
- Leveling instruments
- Construction levels

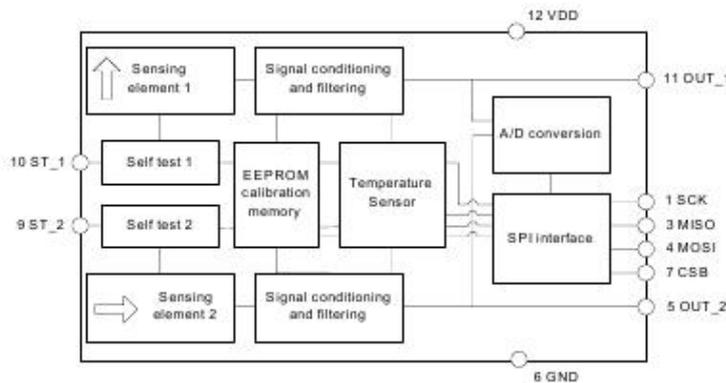


Figure 1. Functional block diagram



SCA100T Series

TABLE OF CONTENTS

The SCA100T Dual Axis Inclinometer Series 1

Features 1

Applications 1

Table of Contents 2

1 Electrical Specifications 3

1.1 Absolute Maximum Ratings 3

1.2 Performance Characteristics 3

1.3 Electrical Characteristics 4

1.4 SPI Interface DC Characteristics 4

1.5 SPI Interface AC Characteristics 4

1.6 SPI Interface Timing Specifications 5

1.7 Electrical Connection 6

1.8 Typical Performance Characteristics 6

 1.8.1 Additional External Compensation 7

2 Functional Description 9

2.1 Measuring Directions 9

2.2 Voltage to Angle Conversion 9

2.3 Ratiometric Output 10

2.4 SPI Serial Interface 10

2.5 Digital Output to Angle Conversion 12

2.6 Self Test and Failure Detection Modes 13

2.7 Temperature Measurement 14

3 Application Information 15

3.1 Recommended Circuit Diagrams and Printed Circuit Board Layouts 15

3.2 Recommended Printed Circuit Board Footprint 16

4 Mechanical Specifications and Reflow Soldering 16

4.1 Mechanical Specifications (Reference only) 16

4.2 Reflow Soldering 17

5 Document Change Control 18

6 Contact Information 18



1 Electrical Specifications

The SCA100T product family comprises two versions, the SCA100T-D01 and the SCA100T-D02 that differ in measurement range. The product version specific performance specifications are listed in the table SCA100T performance characteristics below. All other specifications are common with both versions. Vdd=5.00V and ambient temperature unless otherwise specified.

1.1 Absolute Maximum Ratings

Supply voltage (VDD)	-0.3 V to +5.5V
Voltage at input / output pins	-0.3V to (VDD + 0.3V)
Storage temperature	-55°C to +125°C
Operating temperature	-40°C to +125°C
Mechanical shock	Drop from 1 meter onto a concrete surface (20000g). Powered or non-powered

1.2 Performance Characteristics

Parameter	Condition	SCA100T	SCA100T	Units
		-D01	-D02	
Measuring range	Nominal	±30	±90	°
		±0.5	±1.0	g
Frequency response	-3dB LP ¹⁾	8-28	8-28	Hz
Offset (Output at 0g)	Ratiometric output	Vdd/2	Vdd/2	V
Offset calibration error		±0.11	±0.23	°
Offset Digital Output		1024	1024	LSB
Sensitivity		4	2	V/g
	between 0...1° ²⁾	70	35	mV/°
Sensitivity calibration error		±0.5	±0.5	%
Sensitivity Digital Output		1638	819	LSB / g
Offset temperature dependency	-25...85°C (typical)	±0.008	±0.008	°/°C
	-40...125°C (max)	±0.86	±0.86	°
Sensitivity temperature dependency	-25...85°C (typical)	±0.014	±0.014	%/°C
	-40...125°C (max)	-2.5...+1	-2.5...+1	%
Typical non-linearity	Measuring range	±0.11	±0.57	°
Digital output resolution		11	11	Bits
	between 0...1° ²⁾	0.035	0.07	° / LSB
Output noise density	From DC...100Hz	0.0008	0.0008	° / √Hz
Analog output resolution	Bandwidth 10 Hz ³⁾	0.0025	0.0025	°
Ratiometric error	Vdd = 4.75...5.25V	±1	±1	%
Cross-axis sensitivity	Max.	4	4	%
Long term Stability ⁴⁾		<0.014	<0.014	°

Note 1. The frequency response is determined by the sensing element's internal gas damping.
 Note 2. The angle output has SIN curve relationship to voltage output refer to paragraph **Error! Reference source not found.**
 Note 3. Resolution = Noise density * √(bandwidth)
 Note 4. Power continuously connected (@ 23°C).



SCA100T Series

1.3 Electrical Characteristics

Parameter	Condition	Min.	Typ	Max.	Units
Supply voltage Vdd		4.75	5.0	5.25	V
Current consumption	Vdd = 5 V; No load		4	5	mA
Operating temperature		-40		+125	°C
Analog resistive output load	Vout to Vdd or GND	10			kOhm
Analog capacitive output load	Vout to Vdd or GND			20	nF
Start-up delay	Reset and parity check			10	ms

1.4 SPI Interface DC Characteristics

Parameter	Conditions	Symbol	Min	Typ	Max	Unit
Input terminal CSB						
Pull up current	$V_{IN} = 0\text{ V}$	I_{PU}	13	22	35	μA
Input high voltage		V_{IH}	4		Vdd+0.3	V
Input low voltage		V_{IL}	-0.3		1	V
Hysteresis		V_{HYST}		0.23*Vdd		V
Input capacitance		C_{IN}		2		pF
Input terminal MOSI, SCK						
Pull down current	$V_{IN} = 5\text{ V}$	I_{PD}	9	17	29	μA
Input high voltage		V_{IH}	4		Vdd+0.3	V
Input low voltage		V_{IL}	-0.3		1	V
Hysteresis		V_{HYST}		0.23*Vdd		V
Input capacitance		C_{IN}		2		pF
Output terminal MISO						
Output high voltage	$I > -1\text{mA}$	V_{OH}	Vdd-0.5			V
Output low voltage	$I < 1\text{ mA}$	V_{OL}			0.5	V
Tristate leakage	$0 < V_{MISO} < V_{dd}$	I_{LEAK}		5	100	pA

1.5 SPI Interface AC Characteristics

Parameter	Condition	Min.	Typ.	Max.	Units
Output load	@500kHz			1	nF
SPI clock frequency				500	kHz
Internal A/D conversion time			150		μs
Data transfer time	@500kHz		38		μs



SCA100T Series

1.6 SPI Interface Timing Specifications

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
Terminal CSB, SCK						
Time from CSB (10%) to SCK (90%)		T_{LS1}	120			ns
Time from SCK (10%) to CSB (90%)		T_{LS2}	120			ns
Terminal SCK						
SCK low time	Load capacitance at MISO < 2 nF	T_{CL}	1			μ s
SCK high time	Load capacitance at MISO < 2 nF	T_{CH}	1			μ s
Terminal MOSI, SCK						
Time from changing MOSI (10%, 90%) to SCK (90%).		T_{SET}	30			ns
Data setup time		T_{HDL}	30			ns
Time from SCK (90%) to changing MOSI (10%,90%).						
Data hold time						
Terminal MISO, CSB						
Time from CSB (10%) to stable MISO (10%, 90%).	Load capacitance at MISO < 15 pF	T_{VAL1}	10		100	ns
Time from CSB (90%) to high impedance state of MISO.	Load capacitance at MISO < 15 pF	T_{LZ}	10		100	ns
Terminal MISO, SCK						
Time from SCK (10%) to stable MISO (10%, 90%).	Load capacitance at MISO < 15 pF	T_{VAL2}			100	ns
Terminal CSB						
Time between SPI cycles, CSB at high level (90%)		T_{LH}	15			μ s
When using SPI commands RDAX, RDAY, RWTR: Time between SPI cycles, CSB at high level (90%)		TLH	150			μ s

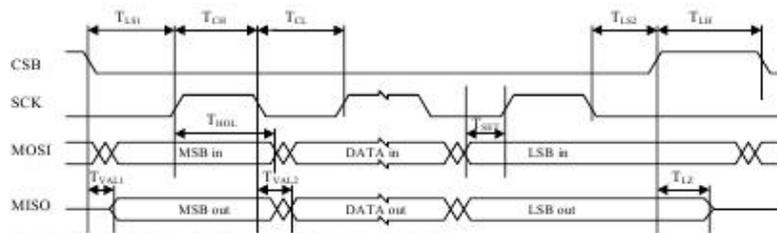


Figure 2. Timing diagram for SPI communication



SCA100T Series

1.7 Electrical Connection

If the SPI interface is not used SCK (pin1), MISO (pin3), MOSI (pin4) and CSB (pin7) must be left floating. Self-test can be activated applying logic "1" (positive supply voltage level) to ST_1 or ST_2 pins (pins 10 or 9). Self-test must not be activated for both channels at the same time. If ST feature is not used pins 9 and 10 must be left floating or connected to GND. Inclination signals are provided from pins OUT_1 and OUT_2.

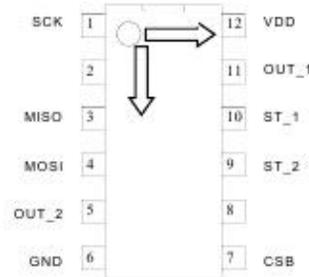


Figure 3. SCA100T electrical connection

No.	Node	I/O	Description
1	SCK	Input	Serial clock
2	NC	Input	No connect, left floating
3	MISO	Output	Master in slave out; data output
4	MOSI	Input	Master out slave in; data input
5	Out_2	Output	Y axis Output (Ch 2)
6	GND	Supply	Ground
7	CSB	Input	Chip select (active low)
8	NC	Input	No connect, left floating
9	ST_2	Input	Self test input for Ch 2
10	ST_1	Input	Self test input for Ch 1
11	Out_1	Output	X axis Output (Ch 1)
12	VDD	Supply	Positive supply voltage (+5V DC)

1.8 Typical Performance Characteristics

Typical offset and sensitivity temperature dependencies of the SCA100T are presented in following diagrams. These results represent the typical performance of SCA100T components. The mean value and 3 sigma limits (mean $\pm 3 \times$ standard deviation) and specification limits are presented in following diagrams. The 3 sigma limits represents 99.73% of the SCA100T population.



SCA100T Series

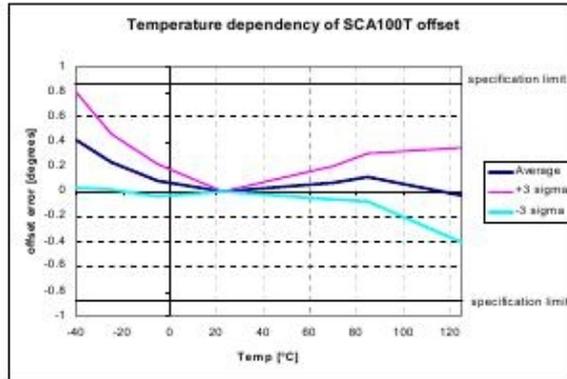


Figure 4. Typical temperature dependency of SCA100T offset

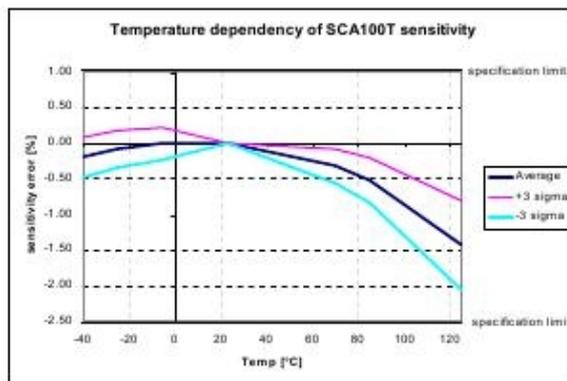


Figure 5. Typical temperature dependency of SCA100T sensitivity

1.8.1 Additional External Compensation

To achieve the best possible accuracy, the temperature measurement information and typical temperature dependency curves can be used for SCA100T offset and sensitivity temperature dependency compensation. The equation of fitted 3rd order polynome curve for offset compensation is:

$$Offcorr = -0.0000006 * T^3 + 0.0001 * T^2 - 0.0039 * T - 0.0522$$

Where:

- Offcorr: 3rd order polynome fitted to average offset temperature dependency curve
- T: temperature in °C (Refer to paragraph 2.7 Temperature Measurement)

The calculated compensation curve can be used to compensate the temperature dependency of the SCA100T offset by using following equation:

$$OFFSETcomp = Offset - Offcorr$$

Where:

- OFFSETcomp: temperature compensated offset in degrees
- Offset: Nominal offset in degrees



SCA100T Series

The equation of fitted 2nd order polynome curve for sensitivity compensation is:

$$Scorr = -0.00011 * T^2 + 0.0022 * T + 0.0408$$

Where:

- Scorr: 2nd order polynome fitted to average sensitivity temperature dependency curve
- T: temperature in °C

The calculated compensation curve can be used to compensate the temperature dependency of the SCA100T sensitivity by using following equation:

$$SENScomp = SENS * (1 + Scorr / 100)$$

Where:

- SENScomp: temperature compensated sensitivity
- SENS: Nominal sensitivity (4V/g SCA100T-D01, 2V/g SCA100T-D02)

The typical offset and sensitivity temperature dependency after external compensation is shown in the pictures below.

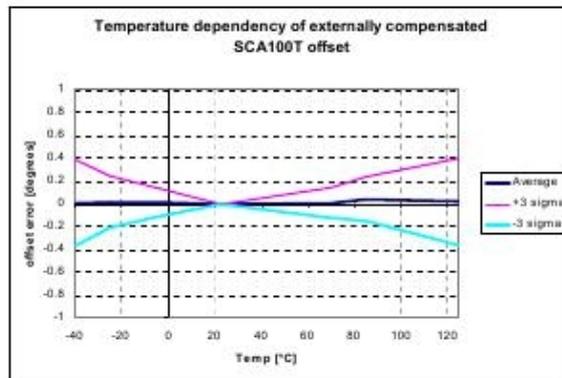


Figure 6. The temperature dependency of an externally compensated SCA100T offset

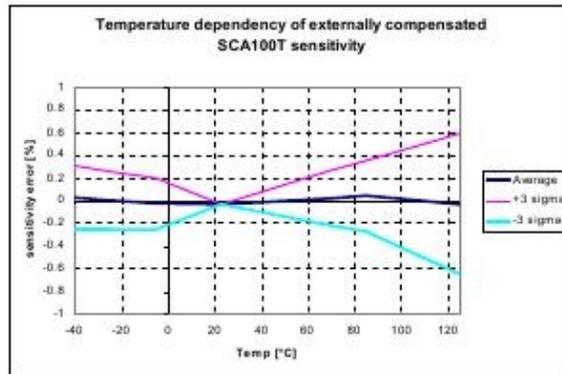


Figure 7. The temperature dependency of an externally compensated SCA100T sensitivity



SCA100T Series

2 Functional Description

2.1 Measuring Directions

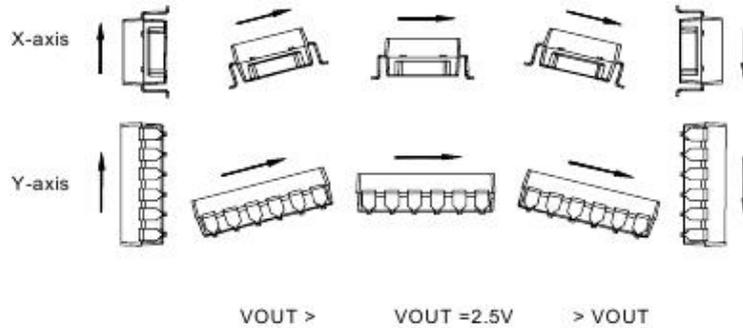


Figure 8. The measuring directions of the SCA100T

2.2 Voltage to Angle Conversion

Analog output can be transferred to angle using the following equation for conversion:

$$\alpha = \arcsin\left(\frac{V_{out} - Offset}{Sensitivity}\right)$$

where: Offset = output of the device at 0° inclination position, Sensitivity is the sensitivity of the device and V_{Dout} is the output of the SCA100T. The nominal offset is 2.5 V and the sensitivity is 4 V/g for the SCA100T-D01 and 2 V/g for the SCA100T-D02.

Angles close to 0° inclination can be estimated quite accurately with straight line conversion but for the best possible accuracy, arcsine conversion is recommended to be used. The following table shows the angle measurement error if straight line conversion is used.

Straight line conversion equation:

$$\alpha = \frac{V_{out} - Offset}{Sensitivity}$$

Where: Sensitivity = 70mV/° with SCA100T-D01 or Sensitivity= 35mV/° with SCA100T-D02

Tilt angle [°]	Straight line conversion error [°]
0	0
1	0.0027
2	0.0058
3	0.0094
4	0.0140
5	0.0198
10	0.0787
15	0.2185
30	1.668



SCA100T Series

2.3 Ratiometric Output

Ratiometric output means that the zero offset point and sensitivity of the sensor are proportional to the supply voltage. If the SCA100T supply voltage is fluctuating the SCA100T output will also vary. When the same reference voltage for both the SCA100T sensor and the measuring part (A/D-converter) is used, the error caused by reference voltage variation is automatically compensated for.

2.4 SPI Serial Interface

A Serial Peripheral Interface (SPI) system consists of one master device and one or more slave devices. The master is defined as a micro controller providing the SPI clock and the slave as any integrated circuit receiving the SPI clock from the master. The ASIC in VTI Technologies' products always operates as a slave device in master-slave operation mode.

The SPI has a 4-wire synchronous serial interface. Data communication is enabled by a low active Slave Select or Chip Select wire (CSB). Data is transmitted by a 3-wire interface consisting of wires for serial data input (MOSI), serial data output (MISO) and serial clock (SCK).

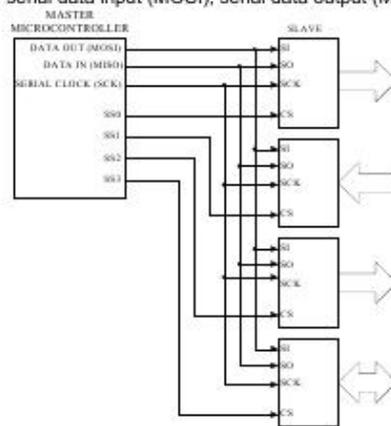


Figure 9. Typical SPI connection

The SPI interface in VTI products is designed to support any micro controller that uses SPI bus. Communication can be carried out by either a software or hardware based SPI. Please note that in the case of hardware based SPI, the received acceleration data is 11 bits. The data transfer uses the following 4-wire interface:

MOSI	master out slave in	$\mu P \rightarrow$ SCA100T
MISO	master in slave out	SCA100T \rightarrow μP
SCK	serial clock	$\mu P \rightarrow$ SCA100T
CSB	chip select (low active)	$\mu P \rightarrow$ SCA100T

Each transmission starts with a falling edge of CSB and ends with the rising edge. During transmission, commands and data are controlled by SCK and CSB according to the following rules:

- commands and data are shifted; MSB first, LSB last
- each output data/status bits are shifted out on the falling edge of SCK (MISO line)



- each bit is sampled on the rising edge of SCK (MOSI line)
- after the device is selected with the falling edge of CSB, an 8-bit command is received. The command defines the operations to be performed
- the rising edge of CSB ends all data transfer and resets internal counter and command register
- if an invalid command is received, no data is shifted into the chip and the MISO remains in high impedance state until the falling edge of CSB. This reinitializes the serial communication.
- data transfer to MOSI continues immediately after receiving the command in all cases where data is to be written to SCA100T's internal registers
- data transfer out from MISO starts with the falling edge of SCK immediately after the last bit of the SPI command is sampled in on the rising edge of SCK
- maximum SPI clock frequency is 500kHz
- maximum data transfer speed for RDAX and RDAY is 5300 samples per sec / channel

SPI command can be either an individual command or a combination of command and data. In the case of combined command and data, the input data follows uninterruptedly the SPI command and the output data is shifted out parallel with the input data.

The SPI interface uses an 8-bit instruction (or command) register. The list of commands is given in Table below.

Command name	Command format	Description:
MEAS	00000000	Measure mode (normal operation mode after power on)
RWTR	00001000	Read and write temperature data register
RDSR	00001010	Read status register
RLOAD	00001011	Reload NV data to memory output register
STX	00001110	Activate Self test for X-channel
STY	00001111	Activate Self test for Y-channel
RDAX	00010000	Read X-channel acceleration through SPI
RDAY	00010001	Read Y-channel acceleration through SPI

Measure mode (MEAS) is standard operation mode after power-up. During normal operation, the MEAS command is the exit command from Self test.

Read temperature data register (RWTR) reads temperature data register during normal operation without affecting the operation. The temperature data register is updated every 150 μs. The load operation is disabled whenever the CSB signal is low, hence CSB must stay high at least 150 μs prior to the RWTR command in order to guarantee correct data. The data transfer is presented in Figure 10 below. The data is transferred MSB first. In normal operation, it does not matter what data is written into temperature data register during the RWTR command and hence writing all zeros is recommended.

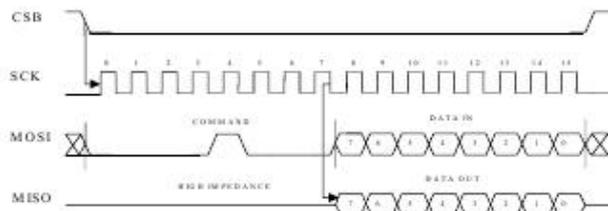


Figure 10. Command and 8 bit temperature data transmission over the SPI

Self test for X-channel (STX) activates the self test function for the X-channel (Channel 1). The internal charge pump is activated and a high voltage is applied to the X-channel acceleration



SCA100T Series

sensor element electrode. This causes the electrostatic force that deflects the beam of the sensing element and simulates the acceleration to the positive direction. The self-test is de-activated by giving the MEAS command. **The self test function must not be activated for both channels at the same time.**

Self test for Y-channel (STY) activates the self test function for the Y-channel (Channel 2). The internal charge pump is activated and a high voltage is applied to the Y-channel acceleration sensor element electrode.

Read X-channel acceleration (RDAX) accesses the AD converted X-channel (Channel 1) acceleration signal stored in acceleration data register X.

Read Y-channel acceleration (RDAY) accesses the AD converted Y-channel (Channel 2) acceleration signal stored in acceleration data register Y.

During normal operation, acceleration data registers are reloaded every 150 μs. The load operation is disabled whenever the CSB signal is low, hence CSB must stay high at least 150 μs prior the RDAX command in order to guarantee correct data. Data output is an 11-bit digital word that is fed out MSB first and LSB last.

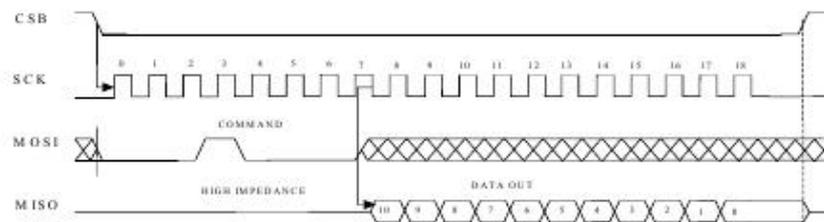


Figure 11. Command and 11 bit acceleration data transmission over the SPI

2.5 Digital Output to Angle Conversion

The acceleration measurement results in RDAX and RDAY data registers are in 11 bit digital word format. The data range is from 0 to 2048. The nominal content of RDAX and RDAY data registers in zero angle position are:

Binary: 100 0000 0000
 Decimal: 1024

The transfer function from differential digital output to angle can be presented as

$$\alpha = \arcsin\left(\frac{D_{out} [LSB] - D_{offset} [LSB]}{Sens [LSB/g]}\right)$$

where;

- D_{out} digital output (RDAX or RDAY)
- D_{offset} digital offset value, nominal value = 1024
- α angle
- $Sens$ sensitivity of the device. (SCA100T-D01: 1638, SCA100T-D02: 819)



SCA100T Series

As an example following table contains data register values and calculated differential digital output values with -5, -1 0, 1 and 5 degree tilt angles.

Angle [°]	Acceleration [mg]	RDAX (SCA100T-D01)	RDAX (SCA100T-D02)
-5	-87.16	dec: 881 bin: 011 0111 0001	dec: 953 bin: 011 1011 1001
-1	-17.45	dec: 995 bin: 011 1110 0011	dec: 1010 bin: 011 1111 0010
0	0	dec: 1024 bin: 100 0000 0000	dec: 1024 bin: 100 0000 0000
1	17.45	dec: 1053 bin: 100 0001 1101	dec: 1038 bin: 100 0000 1110
5	87.16	dec: 1167 bin: 100 1000 1111	dec: 1095 bin: 100 0100 0111

2.6 Self Test and Failure Detection Modes

To ensure reliable measurement results the SCA100T has continuous interconnection failure and calibration memory validity detection. A detected failure forces the output signal close to power supply ground or VDD level, outside the normal output range. The normal output ranges are: analog 0.25-4.75 V (@Vdd=5V) and SPI 102...1945 counts.

The calibration memory validity is verified by continuously running parity check for the control register memory content. In the case where a parity error is detected, the control register is automatically re-loaded from the EEPROM. If a new parity error is detected after re-loading data both analog output voltages are forced to go close to ground level (<0.25 V) and SPI outputs go below 102 counts.

The SCA100T also includes a separate self test mode. The true self test simulates acceleration, or deceleration, using an electrostatic force. The electrostatic force simulates acceleration that is high enough to deflect the proof mass to the extreme positive position, and this causes the output signal to go to the maximum value. The self test function is activated either by a separate on-off command on the self test input, or through the SPI.

The self-test generates an electrostatic force, deflecting the sensing element's proof mass, thus checking the complete signal path. The true self test performs following checks:

- Sensing element movement check
- ASIC signal path check
- PCB signal path check
- Micro controller A/D and signal path check

The created deflection can be seen in both the SPI and analogue outputs. The self test function is activated digitally by a STX or STY command, and de-activated by a MEAS command. Self test can be also activated applying logic "1" (positive supply voltage level) to ST pins (pins 9 & 10) of SCA100T. The self test Input high voltage level is 4 – Vdd+0.3 V and input low voltage level is 0.3 – 1 V. **The self test function must not be activated for both channels at the same time.**



SCA100T Series

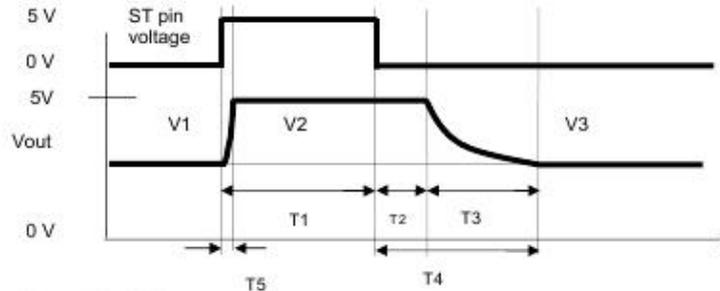


Figure 12. Self test wave forms

- V1 = initial output voltage before the self test function is activated.
- V2 = output voltage during the self test function.
- V3 = output voltage after the self test function has been de-activated and after stabilization time
- Please note that the error band specified for V3 is to guarantee that the output is within 5% of the initial value after the specified stabilization time. After a longer time (max. 1 second) V1=V3.
- T1 = Pulse length for Self test activation
- T2 = Saturation delay
- T3 = Recovery time
- T4 = Stabilization time =T2+T3
- T5 = Rise time during self test.

Self test characteristics:

T1 [ms]	T2 [ms]	T3 [ms]	T4 [ms]	T5 [ms]	V2:	V3:
20-100	Typ. 25	Typ. 30	Typ. 55	Typ. 15	Min 0.95*VDD (4.75V @Vdd=5V)	0.95*V1-1.05*V1

2.7 Temperature Measurement

The SCA100T has an internal temperature sensor, which is used for internal offset compensation. The temperature information is also available for additional external compensation. The temperature sensor can be accessed via the SPI interface and the temperature reading is an 8-bit word (0...255). The transfer function is expressed with the following formula:

$$T = \frac{Counts - 197}{-1.083}$$

Where:

Counts Temperature reading
T Temperature in °C

The temperature measurement output is not calibrated. The internal temperature compensation routine uses relative results where absolute accuracy is not needed. If the temperature measurement results are used for additional external compensation then one point calibration in the system level is needed to remove the offset. With external one point calibration the accuracy of the temperature measurement is about ±1 °C.



3 Application Information

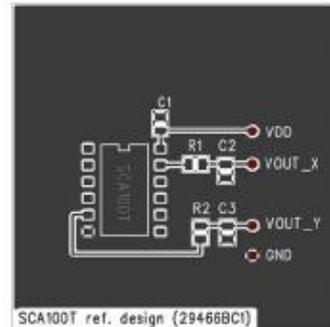
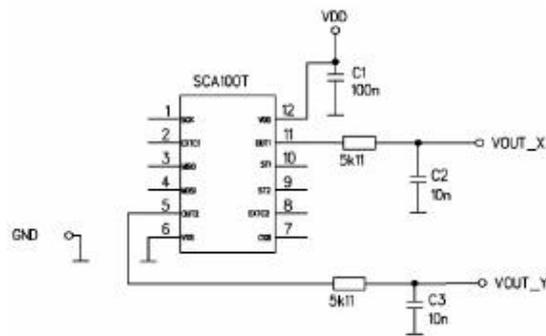
3.1 Recommended Circuit Diagrams and Printed Circuit Board Layouts

The SCA100T should be powered from a well regulated 5 V DC power supply. Coupling of digital noise to the power supply line should be minimized. 100nF filtering capacitor between VDD pin 12 and GND plane must be used.

The SCA100T has a ratiometric output. To get the best performance use the same reference voltage for both the SCA100T and Analog/Digital converter.

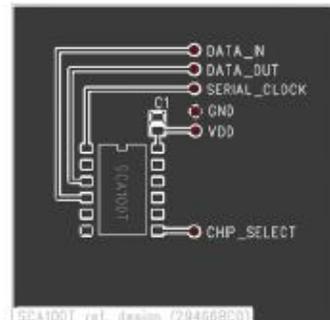
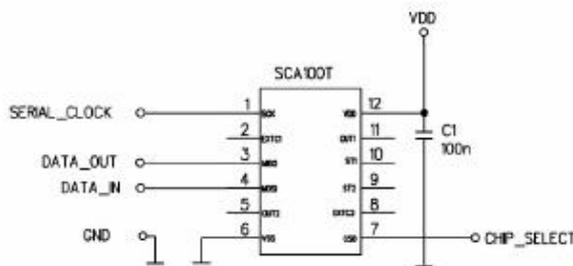
Use low pass RC filters with 5.11 kΩ and 10nF on the SCA100T outputs to minimize clock noise.

Locate the 100nF power supply filtering capacitor close to VDD pin 12. Use as short a trace length as possible. Connect the other end of capacitor directly to the ground plane. Connect the GND pin 6 to underlying ground plane. Use as wide ground and power supply planes as possible. Avoid narrow power supply or GND connection strips on PCB.



SCA100T ref. design (29466BC1)

Figure 13. Analog connection and layout example



SCA100T ref. design (29466BC0)

Figure 14. SPI connection example



SCA100T Series

3.2 Recommended Printed Circuit Board Footprint

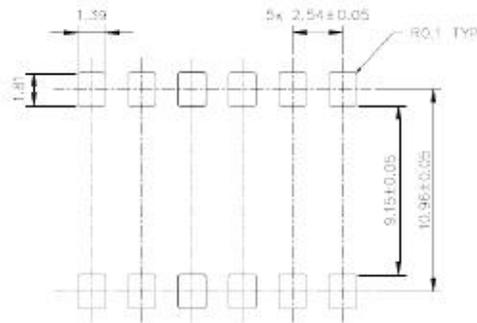


Figure 15. Recommended PCB footprint

4 Mechanical Specifications and Reflow Soldering

4.1 Mechanical Specifications (Reference only)

Lead frame material:	Copper
Plating:	Nickel followed by Gold
Solderability:	JEDEC standard: JESD22-B102-C
RoHS compliance:	RoHS compliant lead free component.
Co-planarity error	0.1mm max.
The part weights	<1.2 g

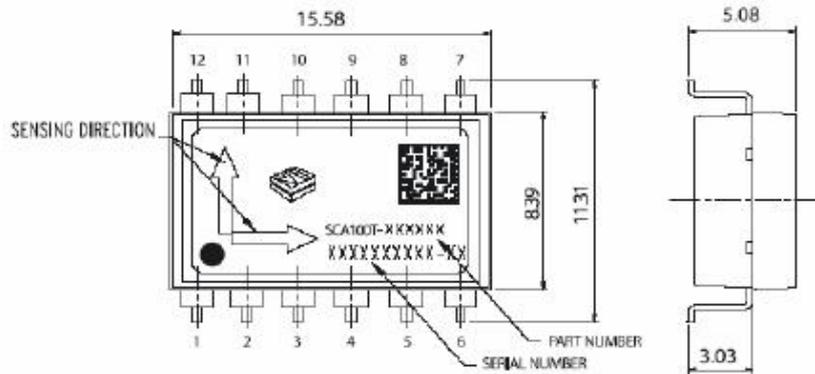


Figure 16. Mechanical dimensions of the SCA100T (Dimensions in mm)



SCA100T Series

4.2 Reflow Soldering

The SCA100T is suitable for Sn-Pb eutectic and Pb- free soldering process and mounting with normal SMD pick-and-place equipment.

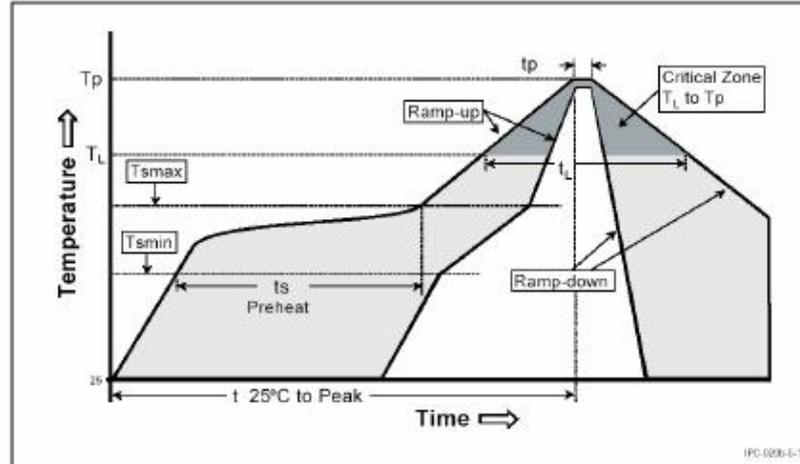


Figure 17. Recommended SCA100T body temperature profile during reflow soldering. Ref. IPC/JEDEC J-STD-020B.

Profile feature	Sn-Pb Eutectic Assembly	Pb-free Assembly
Average ramp-up rate (T _l to T _p)	3°C/second max.	3°C/second max.
Preheat		
- Temperature min (T _{smmin})	100°C	150°C
- Temperature max (T _{smmax})	150°C	200°C
- Time (min to max) (ts)	60-120 seconds	60-180 seconds
T _{smmax} to T _p Ramp up rate		3°C/second max
Time maintained above:		
- Temperature (T _l)	183°C	217°C
- Time (t _l)	60-150 seconds	60-150 seconds
Peak temperature (T _p)	240 +0/-5°C	250 +0/-5°C
Time within 5°C of actual Peak Temperature (T _p)	10-30 seconds	20-40 seconds
Ramp-down rate	6°C/second max	6°C/second max
Time 25° to Peak temperature	6 minutes max	8 minutes max

The Moisture Sensitivity Level of the part is 3 according to the IPC/JEDEC J-STD-020B. The part should be delivered in a dry pack. The manufacturing floor time (out of bag) in the customer's end is 168 hours.

Notes:

- Preheating time and temperatures according to guidance from solder paste manufacturer.
- It is important that the part is parallel to the PCB plane and that there is no angular alignment error from intended measuring direction during assembly process.
- Wave soldering is not recommended.
- **Ultrasonic cleaning is not allowed.** The sensing element may be damaged by an ultrasonic cleaning process.



SCA100T Series

5 Document Change Control

Version	Date	Change Description
A	1.9.-06	Initial release

6 Contact Information

Finland

(head office)

VTI Technologies Oy
P.O. Box 27
Myllynkivenkuja 6
FI-01621 Vantaa
Finland
Tel. +358 9 879 181
Fax +358 9 8791 8791
E-mail: sales@vti.fi

Germany

VTI Technologies Oy
Branch Office Frankfurt
Rennbahnstrasse 72-74
D-60528 Frankfurt am Main,
Germany
Tel. +49 69 6786 880
Fax +49 69 6786 8829
E-mail: sales.de@vti.fi

USA

VTI Technologies, Inc.
One Park Lane Blvd.
Suite 804 - East Tower
Dearborn, MI 48126
USA
Tel. +1 313 425 0850
Fax +1 313 425 0860
E-mail sales@vtitechnologies.com

Japan

VTI Technologies Oy Tokyo Office
Tokyo-to, Minato-ku 2-7-16
Bureau Toranomon
401105-0001
Japan
Tel. +81 3 6277 6618
Fax +81 3 6277 6619

China

VTI Technologies Shanghai Office
6th floor, Room 618
780 Callun Lu
Pudong New Area
201203 Shanghai
P.R. China
Tel. +86 21 5132 0417
Fax +86 21 513 20 416

To find out your local sales
representative visit www.vti.fi

VTI Technologies reserves all rights to modify this document without prior notice.

ANEXO F. Microcontrolador



PIC18F2525/2620/4525/4620 **Data Sheet**

28/40/44-Pin
Enhanced Flash Microcontrollers
with 10-Bit A/D and nanoWatt Technology



MICROCHIP **PIC18F2525/2620/4525/4620**

**28/40/44-Pin Enhanced Flash Microcontrollers with
10-Bit A/D and nanoWatt Technology**

Power Management Features:

- Run: CPU on, Peripherals on
- Idle: CPU off, Peripherals on
- Sleep: CPU off, Peripherals off
- Ultra Low 50nA Input Leakage
- Run mode Currents Down to 11 μ A Typical
- Idle mode Currents Down to 2.5 μ A Typical
- Sleep mode Current Down to 100 nA Typical
- Timer1 Oscillator: 900 nA, 32 kHz, 2V
- Watchdog Timer: 1.4 μ A, 2V Typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, up to 40 MHz
- 4x Phase Lock Loop (PLL) – Available for Crystal and Internal Oscillators
- Two External RC modes, up to 4 MHz
- Two External Clock modes, up to 40 MHz
- Internal Oscillator Block:
 - Fast wake from Sleep and Idle, 1 μ s typical
 - 8 use-selectable frequencies, from 31 kHz to 8 MHz
 - Provides a complete range of clock speeds from 31 kHz to 32 MHz when used with PLL
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock stops

Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA
- Three Programmable External Interrupts
- Four Input Change Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart

Peripheral Highlights (Continued):

- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I²C™ Master and Slave modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN/J2602
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-wake-up on Start bit
 - Auto-Baud Detect
- 10-Bit, up to 13-Channel Analog-to-Digital (A/D) Converter module:
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Programmable 16-Level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection

Special Microcontroller Features:

- C Compiler Optimized Architecture:
 - Optional extended instruction set designed to optimize re-entrant code
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory Typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory Typical
- Flash/Data EEPROM Retention: 100 Years Typical
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 4 ms to 131s
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) via Two Pins
- Wide Operating Voltage Range: 2.0V to 5.5V
- Programmable Brown-out Reset (BOR) with Software Enable Option

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		USART	Comp.	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I ² C™			
PIC18F2525	48K	24576	3968	1024	25	10	2/0	Y	Y	1	2	1/3
PIC18F2620	64K	32768	3968	1024	25	10	2/0	Y	Y	1	2	1/3
PIC18F4525	48K	24576	3968	1024	36	13	1/1	Y	Y	1	2	1/3
PIC18F4620	64K	32768	3968	1024	36	13	1/1	Y	Y	1	2	1/3

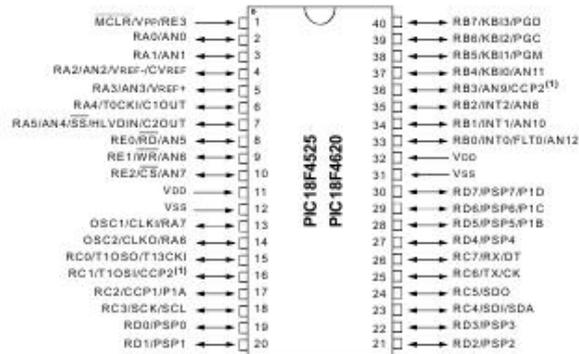
PIC18F2525/2620/4525/4620

Pin Diagrams

28-Pin SPDIP, SOIC



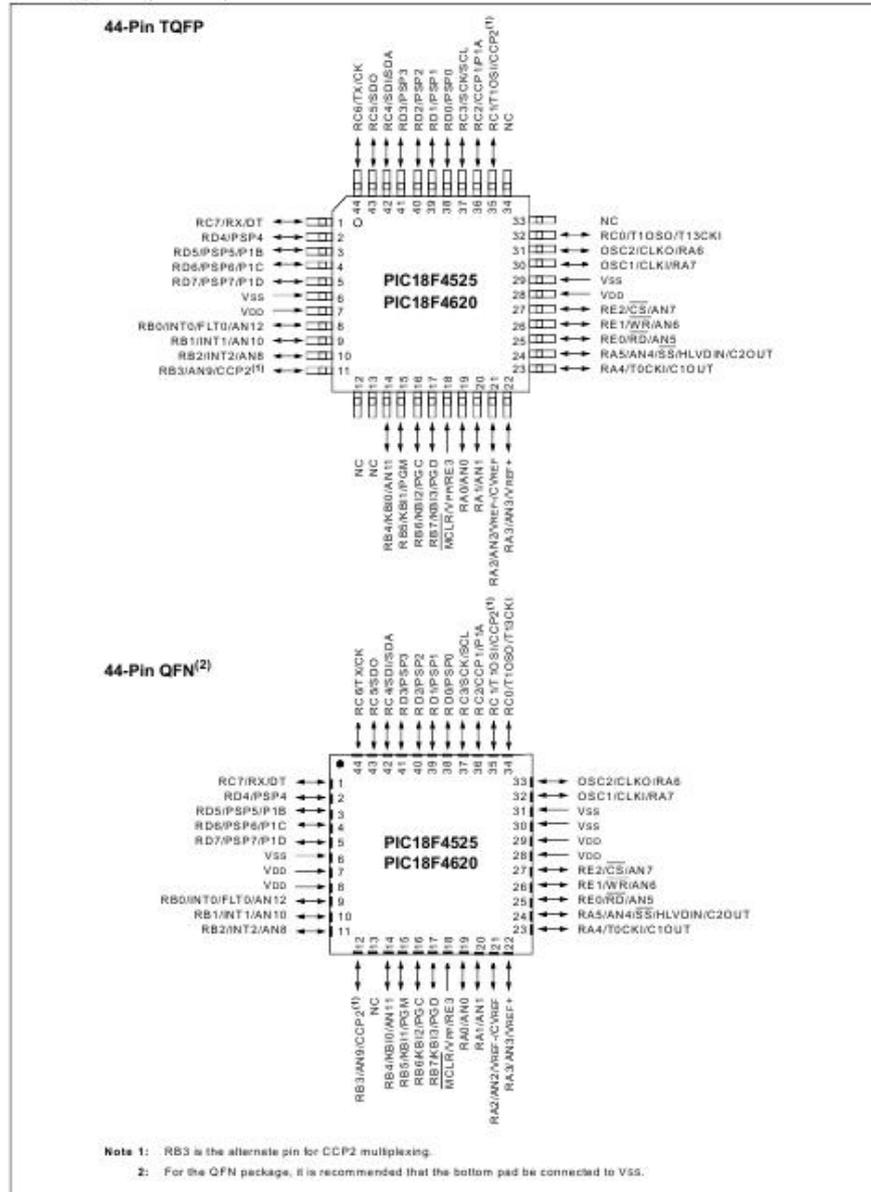
40-Pin PDIP



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

PIC18F2525/2620/4525/4620

Pin Diagrams (Cont.'d)



PIC18F2525/2620/4525/4620

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC18F2525
- PIC18F2620
- PIC18F4525
- PIC18F4620
- PIC18LF2525
- PIC18LF2620
- PIC18LF4525
- PIC18LF4620

This family offers the advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price – with the addition of high-endurance, Enhanced Flash program memory. On top of these features, the PIC18F2525/2620/4525/4620 family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

1.1 New Core Features

1.1.1 nanoWatt TECHNOLOGY

All of the devices in the PIC18F2525/2620/4525/4620 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the internal oscillator block, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4%, of normal operation requirements.
- **On-the-Fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.
- **Low Consumption in Key Modules:** The power requirements for both Timer1 and the Watchdog Timer are minimized. See **Section 26.0 "Electrical Characteristics"** for values.

1.1.2 MULTIPLE OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F2525/2620/4525/4620 family offer ten different oscillator options, allowing users a wide range of choices in developing application hardware. These include:

- Four Crystal modes, using crystals or ceramic resonators
- Two External Clock modes, offering the option of using two pins (oscillator input and a divide-by-4 clock output) or one pin (oscillator input, with the second pin reassigned as general I/O)
- Two External RC Oscillator modes with the same pin options as the External Clock modes
- An internal oscillator block which provides an 8 MHz clock and an INTRC source (approximately 31 kHz), as well as a range of 6 user-selectable clock frequencies, between 125 kHz to 4 MHz, for a total of 8 clock frequencies. This option frees the two oscillator pins for use as additional general purpose I/O.
- A Phase Lock Loop (PLL) frequency multiplier, available to both the High-Speed Crystal and Internal Oscillator modes, which allows clock speeds of up to 40 MHz. Used with the internal oscillator, the PLL gives users a complete selection of clock speeds, from 31 kHz to 32 MHz – all without using an external crystal or clock circuit.

Besides its availability as a clock source, the internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator block, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or wake-up from Sleep mode, until the primary clock source is available.

PIC18F2525/2620/4525/4620

1.2 Other Special Features

- **Memory Endurance:** The Enhanced Flash cells for both program memory and data EEPROM are rated to last for many thousands of erase/write cycles – up to 100,000 for program memory and 1,000,000 for EEPROM. Data retention without refresh is conservatively estimated to be greater than 40 years.
- **Self-Programmability:** These devices can write to their own program memory spaces under internal software control. By using a bootloader routine located in the protected Boot Block at the top of program memory, it becomes possible to create an application that can update itself in the field.
- **Extended Instruction Set:** The PIC18F2525/2620/4525/4620 family introduces an optional extension to the PIC18 instruction set, which adds 8 new instructions and an Indexed Addressing mode. This extension, enabled as a device configuration option, has been specifically designed to optimize re-entrant application code originally developed in high-level languages, such as C.
- **Enhanced CCP Module:** In PWM mode, this module provides 1, 2 or 4 modulated outputs for controlling half-bridge and full-bridge drivers. Other features include auto-shutdown, for disabling PWM outputs on interrupt or other select conditions and auto-restart, to reactivate outputs once the condition has cleared.
- **Enhanced Addressable USART:** This serial communication module is capable of standard RS-232 operation and provides support for the LIN bus protocol. Other enhancements include automatic baud rate detection and a 16-bit Baud Rate Generator for improved resolution. When the microcontroller is using the internal oscillator block, the EUSART provides stable operation for applications that talk to the outside world without using an external crystal (or its accompanying power requirement).
- **10-Bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period and thus, reducing code overhead.
- **Extended Watchdog Timer (WDT):** This enhanced version incorporates a 16-bit prescaler, allowing an extended time-out range that is stable across operating voltage and temperature. See **Section 26.0 “Electrical Characteristics”** for time-out periods.

1.3 Details on Individual Family Members

Devices in the PIC18F2525/2620/4525/4620 family are available in 28-pin and 40/44-pin packages. Block diagrams for the two groups are shown in Figure 1-1 and Figure 1-2.

The devices are differentiated from each other in five ways:

1. Flash program memory (48 Kbytes for PIC18FX525 devices, 64 Kbytes for PIC18FX620 devices).
2. A/D channels (10 for 28-pin devices, 13 for 40/44-pin devices).
3. I/O ports (3 bidirectional ports on 28-pin devices, 5 bidirectional ports on 40/44-pin devices).
4. CCP and Enhanced CCP implementation (28-pin devices have 2 standard CCP modules, 40/44-pin devices have one standard CCP module and one ECCP module).
5. Parallel Slave Port (present only on 40/44-pin devices).

All other features for devices in this family are identical. These are summarized in Table 1-1.

The pinouts for all devices are listed in Table 1-2 and Table 1-3.

Like all Microchip PIC18 devices, members of the PIC18F2525/2620/4525/4620 family are available as both standard and low-voltage devices. Standard devices with Enhanced Flash memory, designated with an “F” in the part number (such as PIC18F2620), accommodate an operating V_{DD} range of 4.2V to 5.5V. Low-voltage parts, designated by “LF” (such as PIC18LF2620), function over an extended V_{DD} range of 2.0V to 5.5V.

PIC18F2525/2620/4525/4620**TABLE 1-1: DEVICE FEATURES**

Features	PIC18F2525	PIC18F2620	PIC18F4525	PIC18F4620
Operating Frequency	DC – 40 MHz			
Program Memory (Bytes)	49152	65536	49152	65536
Program Memory (Instructions)	24576	32768	24576	32768
Data Memory (Bytes)	3968	3968	3968	3968
Data EEPROM Memory (Bytes)	1024	1024	1024	1024
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/Compare/ PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Parallel Communications (PSP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT			
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set Enabled			
Packages	28-Pin SPDIP 28-Pin SOIC	28-Pin SPDIP 28-Pin SOIC	40-Pin PDIP 44-Pin QFN 44-Pin TQFP	40-Pin PDIP 44-Pin QFN 44-Pin TQFP

PIC18F2525/2620/4525/4620

2.0 OSCILLATOR CONFIGURATIONS

2.1 Oscillator Types

PIC18F2525/2620/4525/4620 devices can be operated in ten different oscillator modes. The user can program the Configuration bits, FOSC3:FOSC0, in Configuration Register 1H to select one of these ten modes:

1. LP Low-Power Crystal
2. XT Crystal/Resonator
3. HS High-Speed Crystal/Resonator
4. HSPLL High-Speed Crystal/Resonator with PLL Enabled
5. RC External Resistor/Capacitor with Fosc/4 Output on RA6
6. RCIO External Resistor/Capacitor with I/O on RA6
7. INTIO1 Internal Oscillator with Fosc/4 Output on RA6 and I/O on RA7
8. INTIO2 Internal Oscillator with I/O on RA6 and RA7
9. EC External Clock with Fosc/4 Output
10. ECIO External Clock with I/O on RA6

2.2 Crystal Oscillator/Ceramic Resonators

In XT, LP, HS or HSPLL Oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. Figure 2-1 shows the pin connections.

The oscillator design requires the use of a parallel cut crystal.

Note: Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

FIGURE 2-1: CRYSTAL/CERAMIC RESONATOR OPERATION (XT, LP, HS OR HSPLL CONFIGURATION)

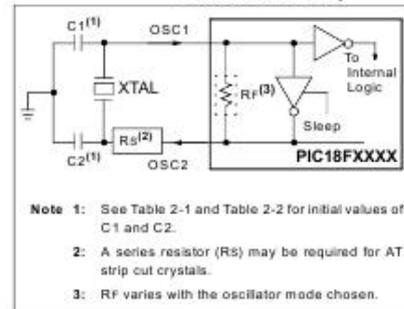


TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Typical Capacitor Values Used:			
Mode	Freq	OSC1	OSC2
XT	3.58 MHz	15 pF	15 pF
	4.19 MHz	15 pF	15 pF
	4 MHz	30 pF	30 pF
	4 MHz	50 pF	50 pF

Capacitor values are for design guidance only.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected V_{DD} and temperature range for the application.

See the notes following Table 2-2 for additional information.

PIC18F2525/2620/4525/4620

17.3.1 REGISTERS

The MSSP module has four registers for SPI mode operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible

SSPCON1 and SSPSTAT are the control and status registers in SPI mode operation. The SSPCON1 register is readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

REGISTER 17-1: SSPSTAT: MSSP STATUS REGISTER (SPI MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE ⁽¹⁾	D/Ā	P	S	R/W	UA	BF
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

- bit 7 **SMP:** Sample bit
SPI Master mode:
 1 = Input data sampled at end of data output time
 0 = Input data sampled at middle of data output time
SPI Slave mode:
 SMP must be cleared when SPI is used in Slave mode.
- bit 6 **CKE:** SPI Clock Select bit⁽¹⁾
 1 = Transmit occurs on transition from active to idle clock state
 0 = Transmit occurs on transition from idle to active clock state
- bit 5 **D/Ā:** Data/Address bit
 Used in I²C mode only.
- bit 4 **P:** Stop bit
 Used in I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.
- bit 3 **S:** Start bit
 Used in I²C mode only.
- bit 2 **R/W:** Read/Write Information bit
 Used in I²C mode only.
- bit 1 **UA:** Update Address bit
 Used in I²C mode only.
- bit 0 **BF:** Buffer Full Status bit (Receive mode only)
 1 = Receive complete, SSPBUF is full
 0 = Receive not complete, SSPBUF is empty

Note 1: Polarity of clock state is set by the CKP bit (SSPCON1<4>).

PIC18F2525/2620/4525/4620

REGISTER 17-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV ⁽¹⁾	SSPEN ⁽²⁾	CKP	SSPM3 ⁽³⁾	SSPM2 ⁽³⁾	SSPM1 ⁽³⁾	SSPM0 ⁽³⁾
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

- bit 7 **WCOL:** Write Collision Detect bit
 1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
 0 = No collision
- bit 6 **SSPOV:** Receive Overflow Indicator bit⁽¹⁾
SPI Slave mode:
 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).
 0 = No overflow
- bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit⁽²⁾
 1 = Enables serial port and configures SCK, SDO, SDI and \overline{SS} as serial port pins
 0 = Disables serial port and configures these pins as I/O port pins
- bit 4 **CKP:** Clock Polarity Select bit
 1 = Idle state for clock is a high level
 0 = Idle state for clock is a low level
- bit 3-0 **SSPM3:SSPM0:** Master Synchronous Serial Port Mode Select bits⁽³⁾
 0101 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control disabled, \overline{SS} can be used as I/O pin
 0100 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control enabled
 0011 = SPI Master mode, clock = TMR2 output/2
 0010 = SPI Master mode, clock = Fosc/64
 0001 = SPI Master mode, clock = Fosc/16
 0000 = SPI Master mode, clock = Fosc/4

- Note 1:** In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
- 2:** When enabled, these pins must be properly configured as input or output.
- 3:** Bit combinations not specifically listed here are either reserved or implemented in I²C™ mode only.

PIC18F2525/2620/4525/4620

17.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL

(SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 17-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP status register (SSPSTAT) indicates the various status conditions.

Note: The SSPBUF register cannot be used with read-modify-write instructions such as BCF, BTFSC and COMF, etc.

Note: To avoid lost data in Master mode, a read of the SSPBUF must be performed to clear the Buffer Full (BF) detect bit (SSPSTAT<0>) between each transmission.

EXAMPLE 17-1: LOADING THE SSPBUF (SSPSR) REGISTER

LOOP	BTFSS	SSPSTAT, BF	; Has data been received (transmit complete)?
	BRA	LOOP	; No
	MOVF	SSPBUF, W	; WREG reg = contents of SSPBUF
	MOVWF	RXDATA	; Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	; W reg = contents of TXDATA
	MOVWF	SSPBUF	; New data to xmit

PIC18F2525/2620/4525/4620

17.3.3 ENABLING SPI I/O

To enable the serial port, MSSP Enable bit, SSPEN (SSPCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and SS pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- SS must have TRISA<5> bit set

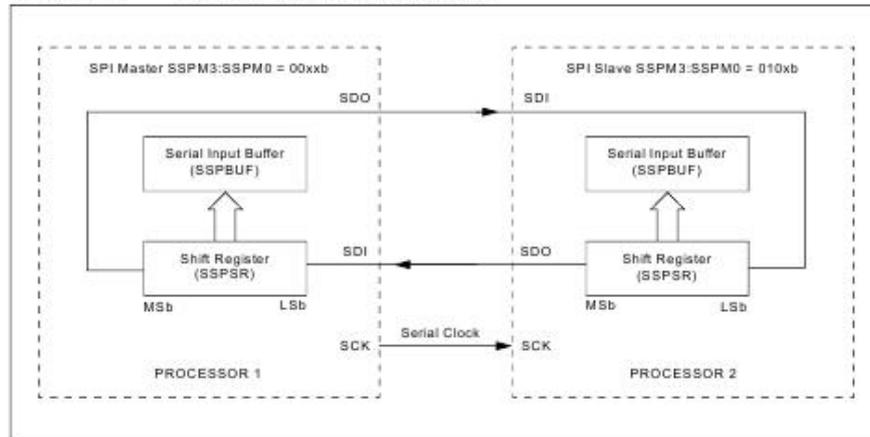
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

17.3.4 TYPICAL CONNECTION

Figure 17-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

FIGURE 17-2: SPI MASTER/SLAVE CONNECTION



PIC18F2525/2620/4525/4620

17.3.5 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 17-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI operation is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "Line Activity Monitor" mode.

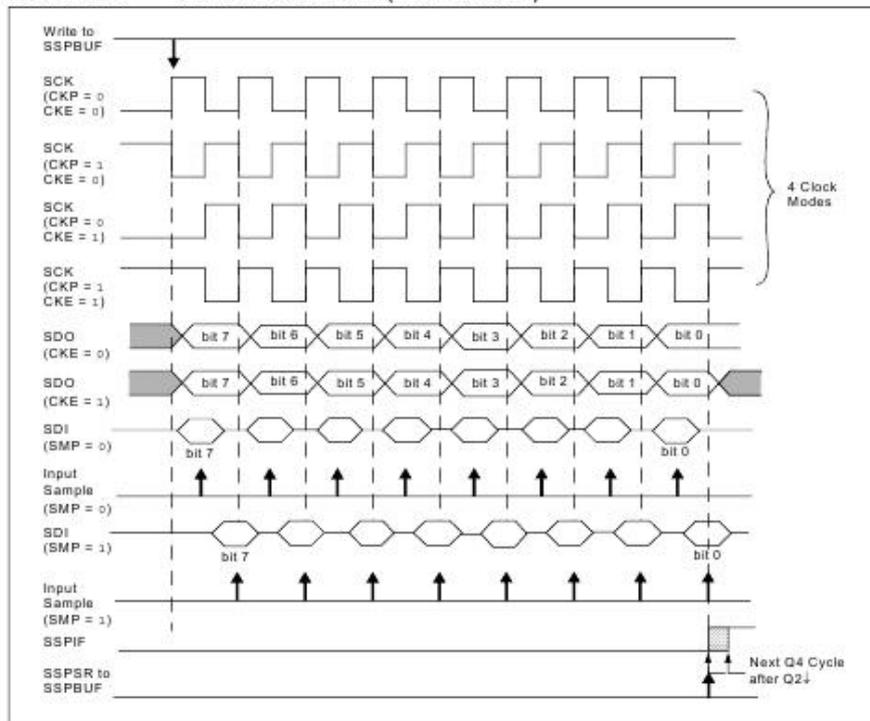
The clock polarity is selected by appropriately programming the CKP bit (SSPCON1<4>). This then, would give waveforms for SPI communication, as shown in Figure 17-3, Figure 17-5 and Figure 17-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{osc}/4$ (or T_{cy})
- $F_{osc}/16$ (or $4 \cdot T_{cy}$)
- $F_{osc}/64$ (or $16 \cdot T_{cy}$)
- $\text{Timer2 output}/2$

This allows a maximum data rate (at 40 MHz) of 10.00 Mbps.

Figure 17-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

FIGURE 17-3: SPI MODE WAVEFORM (MASTER MODE)



PIC18F2525/2620/4525/4620

18.0 ENHANCED UNIVERSAL SYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of the two serial I/O modules. (Generically, the USART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN bus) systems.

The EUSART can be configured in the following modes:

- Asynchronous (full duplex) with:
 - Auto-wake-up on character reception
 - Auto-baud calibration
 - 12-bit Break character transmission
- Synchronous – Master (half duplex) with selectable clock polarity
- Synchronous – Slave (half duplex) with selectable clock polarity

The pins of the Enhanced USART are multiplexed with PORTC. In order to configure RC6/TX/CK and RC7/RX/DT as a USART:

- SPEN bit (RCSTA<7>) must be set (= 1)
- TRISC<7> bit must be set (= 1)
- TRISC<6> bit must be set (= 1)

Note: The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of the Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These are detailed on the following pages in Register 18-1, Register 18-2 and Register 18-3, respectively.

PIC18F2525/2620/4525/4620

REGISTER 18-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7						bit 0	

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

bit 7	<p>CSRC: Clock Source Select bit</p> <p><u>Asynchronous mode:</u> Don't care.</p> <p><u>Synchronous mode:</u> 1 = Master mode (clock generated internally from BRG) 0 = Slave mode (clock from external source)</p>
bit 6	<p>TX9: 9-Bit Transmit Enable bit</p> <p>1 = Selects 9-bit transmission 0 = Selects 8-bit transmission</p>
bit 5	<p>TXEN: Transmit Enable bit⁽¹⁾</p> <p>1 = Transmit enabled 0 = Transmit disabled</p>
bit 4	<p>SYNC: EUSART Mode Select bit</p> <p>1 = Synchronous mode 0 = Asynchronous mode</p>
bit 3	<p>SENDB: Send Break Character bit</p> <p><u>Asynchronous mode:</u> 1 = Send Sync Break on next transmission (cleared by hardware upon completion) 0 = Sync Break transmission completed</p> <p><u>Synchronous mode:</u> Don't care.</p>
bit 2	<p>BRGH: High Baud Rate Select bit</p> <p><u>Asynchronous mode:</u> 1 = High speed 0 = Low speed</p> <p><u>Synchronous mode:</u> Unused in this mode.</p>
bit 1	<p>TRMT: Transmit Shift Register Status bit</p> <p>1 = TSR empty 0 = TSR full</p>
bit 0	<p>TX9D: 9th Bit of Transmit Data</p> <p>Can be address/data bit or a parity bit.</p>

Note 1: SREN/CREN overrides TXEN in Sync mode.

PIC18F2525/2620/4525/4620

REGISTER 18-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

- bit 7 **SPEN:** Serial Port Enable bit
 1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
 0 = Serial port disabled (held in Reset)
- bit 6 **RX9:** 9-Bit Receive Enable bit
 1 = Selects 9-bit reception
 0 = Selects 8-bit reception
- bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
 Don't care.
Synchronous mode – Master:
 1 = Enables single receive
 0 = Disables single receive
 This bit is cleared after reception is complete.
Synchronous mode – Slave:
 Don't care.
- bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
 1 = Enables receiver
 0 = Disables receiver
Synchronous mode:
 1 = Enables continuous receive until enable bit, CREN, is cleared (CREN overrides SREN)
 0 = Disables continuous receive
- bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-Bit (RX9 = 1):
 1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set
 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 9-Bit (RX9 = 0):
 Don't care.
- bit 2 **FERR:** Framing Error bit
 1 = Framing error (can be cleared by reading RCREG register and receiving next valid byte)
 0 = No framing error
- bit 1 **OERR:** Overrun Error bit
 1 = Overrun error (can be cleared by clearing bit, CREN)
 0 = No overrun error
- bit 0 **RX9D:** 9th Bit of Received Data
 This can be address/data bit or a parity bit and must be calculated by user firmware.

PIC18F2525/2620/4525/4620

REGISTER 18-3: BAUDCON: BAUD RATE CONTROL REGISTER

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

bit 7	ABDOVF: Auto-Baud Acquisition Rollover Status bit 1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode (must be cleared in software) 0 = No BRG rollover has occurred
bit 6	RCIDL: Receive Operation Idle Status bit 1 = Receive operation is Idle 0 = Receive operation is active
bit 5	RXDTP: Received Data Polarity Select bit (Asynchronous mode only) <u>Asynchronous mode:</u> 1 = RX data is inverted 0 = RX data received is not inverted
bit 4	TXCKP: Clock and Data Polarity Select bit <u>Asynchronous mode:</u> 1 = Idle state for transmit (TX) is a low level 0 = Idle state for transmit (TX) is a high level <u>Synchronous mode:</u> 1 = Idle state for clock (CK) is a high level 0 = Idle state for clock (CK) is a low level
bit 3	BRG16: 16-Bit Baud Rate Register Enable bit 1 = 16-bit Baud Rate Generator – SPBRGH and SPBRG 0 = 8-bit Baud Rate Generator – SPBRG only (Compatible mode), SPBRGH value ignored
bit 2	Unimplemented: Read as '0'
bit 1	WUE: Wake-up Enable bit <u>Asynchronous mode:</u> 1 = EUSART will continue to sample the RX pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge 0 = RX pin not monitored or rising edge detected <u>Synchronous mode:</u> Unused in this mode.
bit 0	ABDEN: Auto-Baud Detect Enable bit <u>Asynchronous mode:</u> 1 = Enable baud rate measurement on the next character. Requires reception of a Sync field (55h); cleared in hardware upon completion. 0 = Baud rate measurement disabled or completed <u>Synchronous mode:</u> Unused in this mode.

PIC18F2525/2620/4525/4620

18.1 Baud Rate Generator (BRG)

The BRG is a dedicated 8-bit or 16-bit generator that supports both the Asynchronous and Synchronous modes of the EUSART. By default, the BRG operates in 8-bit mode; setting the BRG16 bit (BAUDCON<3>) selects 16-bit mode.

The SPBRGH:SPBRG register pair controls the period of a free-running timer. In Asynchronous mode, bits, BRGH (TXSTA<2>) and BRG16 (BAUDCON<3>), also control the baud rate. In Synchronous mode, BRGH is ignored. Table 18-1 shows the formula for computation of the baud rate for different EUSART modes which only apply in Master mode (internally generated clock).

Given the desired baud rate and F_{osc} , the nearest integer value for the SPBRGH:SPBRG registers can be calculated using the formulas in Table 18-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 18-1. Typical baud rates and error values for the various Asynchronous modes are shown in Table 18-2. It may be advantageous to use the high baud rate (BRGH = 1) or the 16-bit BRG to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing any value (even the same value) to the SPBRGH:SPBRG registers immediately reloads the BRG timer. This may corrupt a transmission or reception already in progress. This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

18.1.1 OPERATION IN POWER-MANAGED MODES

The device clock is used to generate the desired baud rate. When one of the power-managed modes is entered, the new clock source may be operating at a different frequency. This may require an adjustment to the value in the SPBRG register pair.

18.1.2 SAMPLING

The data on the RX pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin when SYNC is clear or when BRG16 and BRGH are both not set. The data on the RX pin is sampled once when SYNC is set or when BRG16 and BRGH are both set.

TABLE 18-1: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	$F_{osc}/[64 (n + 1)]$
0	0	1	8-bit/Asynchronous	$F_{osc}/[16 (n + 1)]$
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	$F_{osc}/[4 (n + 1)]$
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

Legend: x = Don't care, n = value of SPBRGH:SPBRG register pair

PIC18F2525/2620/4525/4620

EXAMPLE 18-1: CALCULATING BAUD RATE ERROR

For a device with F_{OSC} of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

Desired Baud Rate = $F_{OSC}/64 ([SPBRGH:SPBRG] + 1)$

Solving for SPBRGH:SPBRG:

$$X = ((F_{OSC}/\text{Desired Baud Rate})/64) - 1$$

$$= ((16000000/9600)/64) - 1$$

$$= [25.042] = 25$$

Calculated Baud Rate = $16000000/(64 (25 + 1))$

$$= 9615$$

Error = $(\text{Calculated Baud Rate} - \text{Desired Baud Rate})/\text{Desired Baud Rate}$

$$= (9615 - 9600)/9600 = 0.16\%$$

TABLE 18-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART Baud Rate Generator Register High Byte								51
SPBRG	EUSART Baud Rate Generator Register Low Byte								51

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the BRG.

PIC18F2525/2620/4525/4620

26.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings^(†)

Ambient temperature under bias	-40°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to V _{SS} (except V _{DD} and $\overline{\text{MCLR}}$)	-0.3V to (V _{DD} + 0.3V)
Voltage on V _{DD} with respect to V _{SS}	-0.3V to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to V _{SS} (Note 2)	0V to +13.25V
Total power dissipation (Note 1)	1.0W
Maximum current out of V _{SS} pin	300 mA
Maximum current into V _{DD} pin	250 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > V _{DD})	±20 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > V _{DD})	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by all ports	200 mA
Maximum current sourced by all ports	200 mA

Note 1: Power dissipation is calculated as follows:

$$P_{dis} = V_{DD} \times (I_{DD} - \sum I_{OH}) + \sum ((V_{DD} - V_{OH}) \times I_{OH}) + \sum (V_{OL} \times I_{OL})$$

- 2:** Voltage spikes below V_{SS} at the $\overline{\text{MCLR}}$ /V_{PP}/RE3 pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{\text{MCLR}}$ /V_{PP}/RE3 pin, rather than pulling this pin directly to V_{SS}.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

ANEXO G. Placa de Desarrollo



PICDEM™ Z Demonstration Kit User's Guide

Overview

1.4 PICDEM™ Z MOTHERBOARD

The PICDEM Z Motherboard contains many features to enable the evaluation and development of wireless solutions. Figure 1-2 shows the motherboard with a MRF24J40MA Module Daughter Board plugged in. Below is a description of the callouts in the figure. The motherboard schematic, PCB layout and bill of materials are available in Appendix A.

1. Microcontroller 28 and 40-pin DIP Sockets: The sockets allow the use of a variety of 28 and 40-pin DIP package PIC® microcontrollers. The 40-pin socket (U4) contains a PIC18LF4620 microcontroller clocked by a 4 MHz crystal (Y1). The 28-pin socket is unpopulated and is located under the 40-pin DIP package. To use the 28-pin connector, carefully remove the 40-pin DIP microcontroller.
2. Temperature Sensor: The motherboard contains the Microchip TC77 thermal sensor with SPI interface. The TC77 shares the SPI bus with the wireless daughter card plugged into connector J2.
3. LEDs: D1 and D2 are driven by microcontroller ports RA0 and RA1, respectively.
4. Push Button Switches: S2 and S3 are connected to microcontroller ports RB5 and RB4, respectively. There are no external pull-up resistors on the motherboard. Therefore, the internal pull-up on PORTB feature must be enabled.
5. MCLR Push Button Switch: Connected to the MCLR pin of the microcontroller.
6. ICSP™ Jack: The 6-pin RJ-11 jack is used to connect a microcontroller programmer such as the MPLAB ICD 2 in-circuit debugger or MPLAB REAL ICE™ in-circuit emulator.
7. RS-232 Connector: Allows the motherboard to connect to a PC serial port for interactive control or debugging. The microcontroller USART interfaces to the RS-232 connector via a RS-232 level shifter (U5).
8. Wireless Daughter Card Connector: Connector J2 is a 12-pin connector that supplies 3.3V power, 4-wire SPI, reset, wake and interrupt connections to the microcontroller. The pinout is shown in Figure 1-3. The 12-pin connector is a Samtec P/N LST-106-07-F-D.
9. Prototyping Area: Provided to breadboard additional circuitry for development. Connections to +3.3V, ground and microcontroller I/O signals are provided.
10. Power: The motherboard is powered by an external power supply via the 2.5 mm coaxial power connector (J1) or portably via a 9V battery (B1). The on-board voltage regulator (U2) is a LP2981, a micropower 100 mA ultra low-dropout regulator in a SOT-23 package. The maximum input voltage for the LP2981 is 16V. The motherboard is protected from accidental reverse power connection by diode D3. When using a 9V battery, S7 switches the power on and off. When powering via an external power supply, J1 will disconnect the battery from the circuit and power the circuit continuously (switch S7 only switches the battery on or off, not the external power source).

CAUTION

Do not exceed the combined current rating of 100 mA for all circuitry or the voltage regulator will overheat and possibly fail.

Note: A 9V wall adapter may be ordered from microchipDIRECT under part number AC162039.

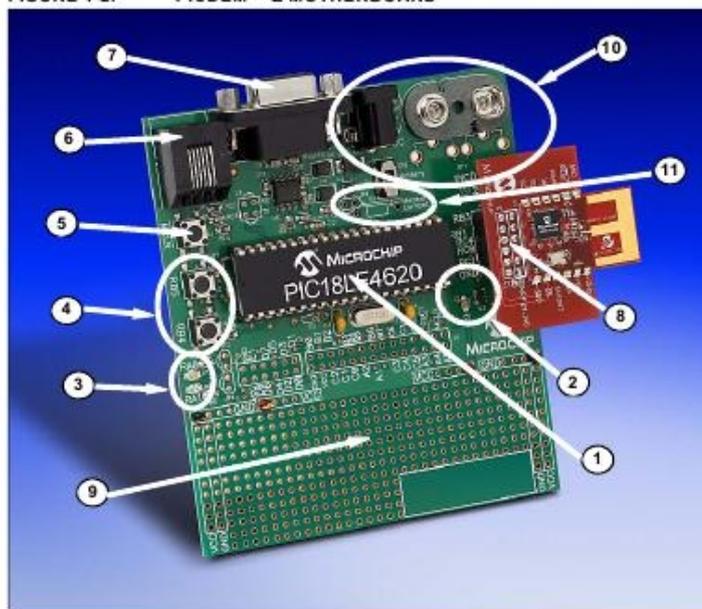
11. Measure Current: Jumper JP4 can be used to measure the current draw by all the circuitry on the motherboard past the voltage regulator (U2). To measure the current, cut the PCB trace on the bottom side and insert an ammeter. You can also install a low value resistor into position R9 and measure the voltage across to determine the current.

PICDEM Z Demonstration Kit User's Guide

TABLE 1-2: JUMPERS

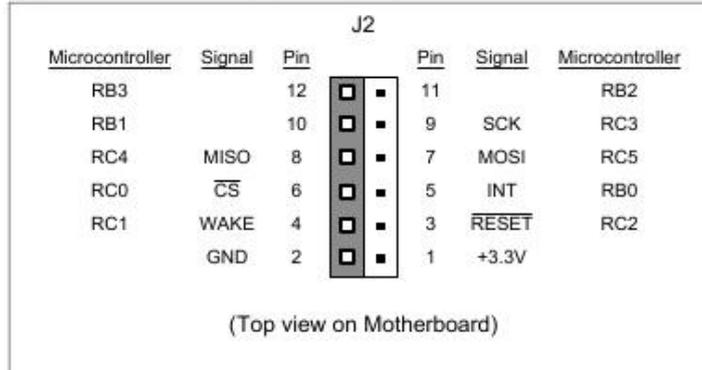
Jumper	Purpose
J3	To connect or disconnect the RS-232 level shifter U5 from the microcontroller USART. The pins are connected by PCB traces.
JP2	To enable or disable LED D1. The pins are connected by PCB traces.
JP3	To enable or disable LED D2. The pins are connected by PCB traces.
JP4	To measure current draw by all circuitry past the voltage regulator U2.

FIGURE 1-2: PICDEM™ Z MOTHERBOARD



Overview

FIGURE 1-3: CONNECTOR J2 PINOUT



1.5 WIRELESS DAUGHTER BOARDS

The PICDEM Z Motherboard is designed to interface with a variety of wireless daughter boards via connector J2. Connector J2 is a 12-pin connector that supplies 3.3V power, 4-wire SPI, reset, wake and interrupt connections to the microcontroller. The pinout is shown in Figure 1-3. The 12-pin connector is a Samtec P/N LST-106-07-F-D.

1.6 PICDEM™ Z CD-ROM

The PICDEM Z CD-ROM contains documentation on the motherboard, wireless daughter boards, data sheets, application notes and wireless protocol software. Check the Microchip web site for the latest revisions <http://www.microchip.com/wireless/>.