



Ingeniería Superior en Informática

Escuela Superior de Ciencias Experimentales y Tecnología

Curso académico 2004-2005

Proyecto Fin de Carrera

SIA:
Sistema Inteligente de Adelantamiento

Tutor: José María Cañas Plaza

Autor1: Carlos Enrique Agüero Durán

Autor2: Víctor Manuel Gómez Gómez

*A mis padres por su cariño eterno y por haberme enseñado cómo se lucha en la vida. A mis abuelos por demostrarme que las inquietudes no tienen edad; **siempre** estaréis conmigo. A mi hermano por demostrarme su eterna amistad. A todos mis amigos por estar ahí cuando se les necesita.*

Carlos E. Agüero

A mi familia y amigos. Gracias por vuestro apoyo incondicional y el cariño que siempre me habéis dispensado.

Víctor M. Gómez

Este proyecto no podría haber llegado a buen puerto sin la ayuda de mucha gente. En primer lugar queríamos dar las gracias a José María Cañas y a Vicente Matellán por confiar y creer en nosotros, y demostrarnos su pasión y sabiduría por lo que hacen día a día, sirviendo de modelo para nosotros. Agradecemos de manera muy especial el esfuerzo de José María, que gracias a su espíritu de trabajo, su dominio del mundo de la robótica y su paciencia, siempre ha tenido la puerta abierta para resolver las dudas que nos iban surgiendo, darnos su punto de vista de las cosas, transmitirnos sus conocimientos, hacer críticas constructivas, felicitarnos cuando nos lo merecíamos e incluso enderezarnos cuando nos desviábamos del buen camino (y le quitábamos horas de sueño).

También queremos agradecer a David Nevado su labor para iluminar un camino que al principio estaba muy oscuro, y que gracias a él fue iluminándose poco a poco. Muchas gracias por recibirnos siempre (a pesar de nuestra insistencia) y por cedernos parte de tu sabiduría sobre el mundo de los microcontroladores.

Finalmente queríamos terminar este capítulo de agradecimientos dando las gracias a todo el grupo de Robótica y a nuestros compañeros de la URJC, por hacernos más ameno el día a día de la universidad y por mantener el buen ambiente que nos ha rodeado siempre. A nuestro compañero Pablo Barrera le debemos un agradecimiento especial por su labor de bombero cuando el fuego nos quemaba y por su inestimable ayuda siempre que la necesitábamos.

Gracias a todos.

Resumen

¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida mas fácil, nos aporta tan poca felicidad? La respuesta es ésta, simplemente: Porque aún no hemos aprendido a usarla con tino.

Albert Einstein

La tecnología avanza imparable y la industria automovilística hace uso de los últimos avances e investigaciones para dotar a sus vehículos de mayor comodidad, seguridad y prestaciones. En el apartado de seguridad, numerosos avances van incorporándose paulatinamente a nuestros automóviles para aumentar tanto la seguridad activa como pasiva (*ABS*, *Airbag*, controles de tracción y estabilidad, etc.). En los últimos tiempos se ha trabajado en mejorar también la seguridad teniendo en cuenta al resto de vehículos (radares que permiten mantener distancias mínimas entre los automóviles [Volvo, 2005], cámaras para vigilar los puntos muertos [Volvo, 2005], etc.).

El objetivo de este proyecto industrial es crear una ayuda a la conducción en esta línea, es decir, teniendo en cuenta el tráfico que rodea a un vehículo, para agilizar la maniobra de adelantamiento. En concreto, esta ayuda se destina a los automóviles pesados. En ocasiones y debido a su alto peso y limitación de velocidad, ocasionan parones en el carril izquierdo cuando realizan maniobras de adelantamiento, poniendo en peligro los vehículos que circulan por detrás y que se ven sorprendidos por un vehículo lento en su carril. Este proyecto pretende diseñar, construir y programar un sistema inteligente de adelantamiento implantado en automóviles pesados, que sea capaz de identificar situaciones de comienzo y final de adelantamiento aumentando la fluidez en el tráfico. Cuando el sistema ha identificado positivamente el inicio de la maniobra, podría subir el límite máximo de velocidad y reducir el tiempo del adelantamiento, disminuyendo el riesgo de encontronazo con otro vehículo que llegara desde atrás.

El sistema utiliza una serie de sensores de ultrasonido y un sensor láser de barrido distribuidos por el vehículo. La función de estos sensores es medir la distancia a los automóviles cercanos y entregar estos datos a un módulo hardware. Este unidad se encarga de recoger los números ofrecidos por los sensores y los envía de una manera simplificada, informativa y ordenada a un ordenador central. Dicho ordenador central está programado para computar todas las medidas y estimar si el vehículo se encuentra en una situación de comienzo o final de adelantamiento, elevando o descendiendo respectivamente el umbral máximo de velocidad.

Índice general

Índice de figuras	XI
Índice de tablas	XI
1. Introducción	1
1.1. Tecnología en la automoción	1
1.2. Robótica	5
1.3. SIA (Sistema Inteligente de Adelantamiento)	8
2. Objetivos	11
2.1. Objetivos	11
2.2. Requisitos	13
2.3. Metodología	14
3. Estudio Sensorial	17
3.1. Sensores de ultrasonido	18
3.2. Sensores láser	20
3.3. Sensores radar	22
3.4. Visión	23
4. Diseño	25
4.1. Diseño global	25
4.1.1. Fase perceptiva	26
4.1.2. Fase de análisis y control	28
4.2. Subsistema sónar	29
4.2.1. Microcontroladores	30
4.3. Subsistema láser	31
4.4. Unidad de proceso	32
4.5. Arquitectura final	33
4.6. Diseño eléctrico	35
5. Descripción informática	39

5.1. Subsistema sónar	40
5.1.1. Configuración del PIC	41
5.1.2. Ciclo de medidas del ultrasonido	45
5.1.3. Rutina de tratamiento de interrupción	47
5.1.4. Interfaz de comandos	48
5.1.5. Especificación del protocolo	49
5.1.6. Implementación usando un <i>buffer</i> circular	50
5.2. Programa Monitor	53
5.2.1. <i>Toolkit</i> gráfico: GTK+	53
5.2.2. GUI de Monitor	55
5.2.3. Estructura de <i>Monitor</i>	56
5.3. Aplicación de control	59
5.3.1. Detección frontal del vehículo	61
5.3.2. Detección lateral del vehículo	64
6. Pruebas Experimentales	67
6.1. Pruebas del sistema sensorial en laboratorio	67
6.2. Pruebas del sistema sensorial en exteriores	75
6.3. Pruebas de la aplicación	78
7. Conclusiones y mejoras	81
7.1. Conclusiones	81
7.2. Líneas Futuras	84
Bibliografía	85

Índice de figuras

1.1. Sistema de ayuda al aparcamiento.	3
1.2. Robot conductor Klaus.	4
1.3. Sandstorm.	5
1.4. Robot de soldadura.	6
1.5. Mascota AIBO de Sony (a) y aspiradora Roomba (b)	7
2.1. Sistema detectando el comienzo de un adelantamiento	11
2.2. Sensor radar Eaton Vorad que además de medir distancias es capaz de estimar la velocidad de los obstáculos hasta los 200 Km/h	12
2.3. Metodología en espiral	14
3.1. Esquema que muestra la situación de los sensores en el vehículo	17
3.2. Interpretación de la lectura sónar (izquierda) y alcance experimental (derecha)	18
3.3. Polaroid Smart Sensor.	19
3.4. Imagen del sensor láser (a) y medidas obtenidas por un modelo de barrido (b)	21
3.5. Sensor radar	22
4.1. Mapa con las dos zonas cubiertas por los sensores	26
4.2. Tarjeta de adquisición PCI	29
4.3. Ejemplo de microcontrolador Hitachi H8	30
4.4. Detalle del programador empleado en el proyecto SIA	31
4.5. Modelo láser empleado en el proyecto SIA	32
4.6. Unidades funcionales que componen la aplicación	34
4.7. Detalle de los reguladores y bornas de alimentación	36
4.8. Diagrama esquemático de la placa de control sónar	37
4.9. Fotografía de la placa de control sónar	37
5.1. Programas que componen el proyecto SIA	39
5.2. Diagrama de flujo del programa del PIC 16F876.	40
5.3. Ejemplo de uso del <i>buffer</i> de comandos recibidos	44

5.4.	Diagrama de flujo del ciclo de medidas sónar	45
5.5.	Formato del mensaje enviado al PC en caso de recepción de ECHO . .	46
5.6.	Formato del mensaje enviado al PC en caso de desbordamiento del contador	46
5.7.	Ejemplo de lectura usando direccionamiento indirecto	51
5.8.	Aspecto general de la aplicación Monitor	53
5.9.	Pantallazos de los entornos Anjuta a) y Glade b)	54
5.10.	Detalle de la zona de visualización de ultrasonidos	55
5.11.	Detalle de la zona de visualización láser	56
5.12.	Árbol de directorios creados para Monitor	57
5.13.	Diagrama de flujo de Monitor	58
5.14.	Diagrama de flujo del autómatas empleado en la aplicación de control .	60
5.15.	Aspecto de la aplicación de control desarrollada para el proyecto SIA .	60
5.16.	Medidas obtenidas por el sensor láser	61
5.17.	Proceso teórico de segmentación	63
5.18.	Proceso de segmentación de los puntos láser	64
6.1.	Elementos utilizados para el montaje en interiores	68
6.2.	Gráfica que muestra que el tiempo de vuelo de la onda ultrasónica es lineal a medida que aumenta la distancia al obstáculo	69
6.3.	Gráfica que muestra la apertura del ángulo de la onda ultrasónica a medida que el tiempo de vuelo es mayor	69
6.4.	Imagen del objeto medido (a) y medidas obtenidas por el sonar (b) . .	70
6.5.	Secuencia de imágenes de una persona acercándose al sónar a distancia larga (a), media (b), corta (c) y las medidas obtenidas por el sensor (d)	71
6.6.	Imagen de la persona pasando (a) y medidas obtenidas por el sonar (b)	72
6.7.	Secuencia de imágenes de tres personas pasando delante del sonar, primera persona (a), segunda persona (b), tercera (c) y las medidas obtenidas por el sensor (d)	73
6.8.	Imagen del láser midiendo objetos (a) y medidas obtenidas por el láser (b). Imagen del láser girado hacia su derecha y midiendo los objetos (c) y figura con las medidas obtenidas (d)	74
6.9.	Montaje del sistema sensorial para exteriores.	75
6.10.	Secuencia de fotografías del <i>parking</i> real	76
6.11.	Datos obtenidos por el subsistema sónar	76
6.12.	Secuencia de fotografías de la maniobra de adelantamiento	77
6.13.	Datos obtenidos durante el inicio, transcurso y fin del adelantamiento .	77
6.14.	Maniobra de adelantamiento real a un camión a 120 Km/h.	77
6.15.	Medidas obtenidas durante la maniobra de adelantamiento	78
6.16.	Secuencia de simulación de adelantamiento usando un robot	78
6.17.	Capturas de pantalla obtenidas desde la aplicación de control	79

Índice de tablas

3.1. Características del modelo láser LMS-221 de Sick	22
4.1. Comandos de control	35
4.2. Comandos de datos	35
5.1. Comandos de configuración	49
5.2. Comandos de datos	50
6.1. Medida Frontal	68
6.2. Apertura del haz	69

Introducción

En este capítulo introductorio se describe el contexto en el que se desarrolla este proyecto, el cuál está ligado al campo de la automoción y al de la robótica. Además se comentan los avances tecnológicos que se han producido a lo largo de los años en el campo de la automoción, haciendo uso de elementos comunes con la robótica como las decisiones autónomas, los sensores, etc. Para terminar se presenta el proyecto que se comenta con más detalle en esta memoria.

1.1. Tecnología en la automoción

En los últimos años en la industria automovilística ha surgido la necesidad de hacer vehículos con más y mejores prestaciones, principalmente en seguridad y facilidad de conducción. Estos avances vienen impulsados por la competencia entre fabricantes. La tecnología es un atractivo más del automóvil, y supone una ventaja competitiva frente a otros fabricantes. Los fabricantes han ido creando nuevos sistemas que se han centrado mayormente en la seguridad, comodidad y mayores prestaciones. Estos nuevos sistemas hacen que los vehículos cada vez sean más seguros, ya que los accidentes de tráfico son una de las primeras causas de mortalidad en los países desarrollados. En los últimos años aumentan las prestaciones técnicas de los vehículos, mayor velocidad, potencia, etc., y se hace necesario aumentar los sistemas de seguridad de éstos.

Desde el punto de vista técnico se distinguen tres tipos de seguridad en los vehículos:

- *Seguridad activa.* Al hablar de seguridad activa en un automóvil se hace referencia al conjunto de mecanismos o dispositivos destinados a disminuir el riesgo de que se produzca un accidente. Así, unos frenos eficaces, una dirección precisa, unos neumáticos y amortiguadores en buen estado o un motor con buena capacidad de respuesta son factores que intervienen en la seguridad activa. Los sistemas específicamente desarrollados en las últimas décadas para mejorar la seguridad activa son el *ABS*, la tracción total o los controles de estabilidad y tracción.

Los *controles de tracción* evitan que patinen las ruedas sirviéndose de los sensores del *ABS* para funcionar. Éstos controles evitan que se produzcan pérdidas de motricidad por exceso de aceleración, y no son capaces de recuperar la trayectoria del vehículo en caso de excesivo subviraje (el coche gira menos de lo que quiere

el conductor, el coche entra atravesado) o sobreviraje (el coche gira más de lo que quiere el conductor, el coche culea). Hay controles de tracción que sólo actúan sobre el motor, reduciendo la potencia, aunque el conductor mantenga el acelerador pisado a fondo (ya sea mediante el control del encendido, la inyección o, en algunos casos, incluso desconectando momentáneamente algún cilindro). Otros actúan sobre los frenos, a modo de diferencial autoblocante, pues frenan la rueda que patina para que llegue la potencia a la que tiene más adherencia. También hay sistemas de control de tracción que combinan la actuación sobre motor y frenos.

Al igual que el control de tracción, el *control de estabilidad* se trata de un sistema que utilizando los sensores y la instalación del *ABS*, es capaz de evitar que se produzca una pérdida de control del vehículo, para lo cual actúa sobre el motor y selectivamente sobre los frenos. Básicamente, se trata de generar una fuerza contraria a la que tiende a sacar el coche de su trayectoria ideal. Para realizar esto, el control de tracción se sirve de una serie de sensores (de velocidad de giro de las ruedas, de aceleración transversal y vertical, etc.) y una centralita electrónica que es capaz de saber si el vehículo se sale de la trayectoria marcada por el volante. Su principal ventaja es que hace mejor incluso al conductor más experto, gracias a su capacidad para frenar una única rueda generando pares de fuerza imposibles de conseguir por un conductor que aplica el freno sobre los dos ejes.

- *Seguridad pasiva.* Se encarga de minimizar los posibles daños de los ocupantes del vehículo en el caso de que llegue a producirse un accidente. En la seguridad pasiva se engloban desde el diseño de las estructuras de deformación del vehículo para que absorban la energía en caso de impacto, hasta los cinturones de seguridad o los *airbag*.

En Volvo [Volvo, 2005] han desarrollado sistemas de seguridad pasiva como un *airbag* de techo lateral o el sistema *WHIPS*. El exclusivo sistema de *airbag* de techo lateral consta de dos *cortinas de protección* hinchables que se inflan en paralelo a las ventanillas delanteras y traseras en 25 milésimas de segundo, absorbiendo un 75 % de la energía que recibe la cabeza cuando sale despedida hacia el lateral. El *airbag* de techo lateral también ayuda a reducir el riesgo de que los ocupantes salgan despedidos por las ventanillas.

WHIPS es un sistema de protección contra latigazos cervicales. Los asientos delanteros, incluidos los respaldos y los reposacabezas, se desplazan y se inclinan hacia atrás para ayudar a absorber el impacto. De esta forma, se consigue una protección notablemente mayor contra lesiones del cuello y la espalda en caso de colisión por detrás.

- *Seguridad preventiva.* Afecta a todos aquellos aspectos que influyen en el conductor para que no llegue a producirse un accidente. Dentro de los elementos de seguridad preventiva se pueden incluir desde un diseño que asegure la ergonomía y la visibilidad o una correcta climatización hasta unos asientos cómodos que sujeten correctamente el cuerpo.

Actualmente empresas como Citroën han querido ir más allá y sus modelos C5 y C4 poseen un sistema que alerta al conductor de la pérdida de la trayectoria por

sueño o despiste haciendo vibrar el asiento.

Pero no sólo se investiga en temas de seguridad, también existen avances para aumentar la comodidad y facilitar la conducción. Los faros autodireccionables permiten tener una mayor iluminación en curvas. Los asistentes de aparcamiento dan al conductor una medida de distancia al objeto anterior y/o posterior mediante sensores, como el mostrado en la figura 1.1, etc.

Asímismo, en la actualidad Bosch está trabajando en una ampliación de estos asistentes de aparcamiento que serán capaces de medir los huecos de aparcamiento. Unos sensores de ultrasonido que se montan en el lateral de los parachoques miden, al pasar el coche, la longitud del hueco de aparcamiento. El conductor sabrá, de esta forma, si el hueco es suficiente para aparcar su coche. En otra fase de desarrollo, el sistema podrá calcular las maniobras necesarias y dirigir al conductor con señales acústicas y ópticas. Mientras que en esta variante el conductor todavía maneja el volante, en el siguiente nivel de desarrollo el sistema asumirá incluso el manejo del volante, convirtiéndose en un asistente semi-autónomo.

Bosch está trabajando ya en un asistente de aparcamiento totalmente autónomo, que no sólo conducirá, sino que controlará también el motor y los frenos. La quimera del aparcamiento automático al pulsar un botón podría convertirse en realidad a lo largo de la próxima década.



Figura 1.1: Sistema de ayuda al aparcamiento.

Estos sistemas que dotan de inteligencia a los vehículos no sólo están dirigidos a los turismos. En el segmento de los vehículos pesados también se han creado sistemas de alta tecnología. Por ejemplo el *Adaptative Cruise Control* que se encarga de mantener la distancia respecto al vehículo que le precede, observando el tráfico mediante un sensor radar. El sensor radar registra la zona delantera del vehículo y, a partir de estas informaciones, analiza la situación del tráfico, frenando y acelerando automáticamente. Además es capaz de preveer el rumbo del vehículo propio. Ésto permite reconocer a los vehículos que le preceden incluso en las curvas. En el Instituto Nacional de Técnica Aeroespacial han desarrollado el *Copiloto Tecnológico*, un sistema para vehículos pesados. Se sirve de 4 sensores situados en el frontal del vehículo para

detectar variaciones en la trayectoria del vehículo, así como posibles zigzagueos. Cuando el sistema advierte falta de respuesta del conductor o cansancio avisa a éste mediante señales acústicas y luminosas a través de una pequeña pantalla situada en el cuadro de mandos del vehículo. El detector de presión de neumáticos es otra muestra de tecnología en vehículos pesados.

Además de estos sistemas completamente comerciales, se está investigando también en otros que actualmente son prototipos, como los vehículos completamente autónomos. En la universidad Militar de Munich se desarrolló un sistema de seguimiento de trayectoria a alta velocidad por Dickmanns y Zapp [Dickmanns, 1997]¹, haciendo uso de cámaras de vídeo. En Volkswagen existe un robot conductor llamado *Klaus*², fabricado de aluminio, que tiene cuatro brazos (uno para accionar el contacto, otro para cambiar la marcha y los dos restantes para manejar el volante) y tres piernas (una para cada pedal) como se puede ver en la figura 1.2. Este robot es capaz de conducir, ayudado por otras innovaciones tecnológicas, mediante un sistema de sensores con cuatro escáneres, una cámara de vídeo y un radar que controlan todas las incidencias en el camino. Los datos llegan a un ordenador, que ha sido programado con una determinada ruta a través de un sistema de navegación vía satélite, para tomar la ruta más adecuada. El robot reacciona de acuerdo a las órdenes que recibe de este ordenador.



Figura 1.2: Robot conductor Klaus.

También el departamento de defensa estadounidense busca que los coches circulen por sí solos. Los necesitan para fabricar con ellos vehículos espía, o convoyes de suministro no tripulados. La idea es minimizar la pérdida de vidas humanas en los conflictos bélicos. En esta línea realizó este año la primera edición del *Darpa Grand Challenge*, un reto que premia al vehículo autónomo sin piloto que sea capaz de recorrer la distancia entre Los Ángeles y Las Vegas llevando mercancías por cualquier terreno. Las mejores universidades del país y algunos de los técnicos más cualificados respondieron a la llamada. Un millón de dólares de premio sirvió para estimular todavía más el talento creativo de los estadounidenses.

¹http://imagelab.ing.unimo.it/IVCCVS/01_dickmanns.pdf

²<http://www.terra.com.mx/Automovil/articulo/065400>

Red Whittaker, especialista en robótica que lidera el equipo participante de la Universidad Carnegie Mellon, fue el que más lejos llegó aunque no alcanzó la meta. Su vehículo no tripulado de la figura 1.3, el Sandstorm, posee un sistema de sensores consistente en una cúpula que mantiene en posición un conjunto de cámaras estéreo y un sistema láser.



Figura 1.3: Sandstorm.

En otra línea de investigación, la casa Toyota ha desarrollado junto con Sony, un prototipo de *coche kit*, el Pod³. El coche saluda con los faros cuando el conductor le va a recoger y es capaz de tomarle las constantes vitales y emocionales mientras conduce, expresando los sentimientos del dueño y dejándose conducir según estas emociones. Es decir, el coche es capaz de percibir a través de sensores el pulso del conductor y otras constantes vitales, de modo que no deja pasar de cierto límite de velocidad, etc. Si el coche se queda sin gasolina, se vuelve azul con un tipo de manchas que se asemejan a las lágrimas. Incluso puede enseñar al conductor a mejorar la conducción porque tiene almacenados los parámetros de conducción de un experto.

1.2. Robótica

Esa incorporación de tecnología a los vehículos que hemos comentado en la sección anterior tiene muchos puntos comunes con la robótica y las tecnologías desarrolladas en ella. En este apartado comentaremos de manera general en qué consiste la robótica, detallaremos algunos ejemplos de aplicaciones que ayudan a entender mejor de qué manera influye en el campo de la automoción y en este proyecto en concreto. También se detallará qué partes de la robótica han aportado más al desarrollo de sistemas inteligentes para vehículos.

Primero vamos a ver qué se entiende por robot. Un robot es una máquina programable que puede medir características de la realidad, medir alguna característica suya, y además es capaz de interactuar con su entorno, de actuar en él provocando cambios. Estas dos capacidades lo caracterizan. Un robot no es sólo una computadora, es algo más. Utiliza los procesadores como elementos de cómputo pero un robot tiene más elementos, como los sensores y los actuadores, que lo sumergen y conectan inevitablemente con la realidad física y todo ello en una única plataforma. En cuanto a su fisonomía, los robots pueden presentarse en múltiples configuraciones, pero lo

³<http://www.contactomagazine.com/autos0321.htm>

realmente distintivo es su funcionalidad, no su apariencia. Hay robots estáticos, cuya base no se mueve de un punto fijo, y robots móviles que pueden desplazarse por el entorno. Dentro de los móviles hay robots con ruedas y robots con patas. Hay robots humanoides que tienen forma similar al cuerpo humano, algunos que se asemejan a insectos, otros a coches, etc.

En cuanto a su función, el hombre siempre se ha servido de herramientas para su propio beneficio. Un paso más en esta línea es la construcción de máquinas que realizan alguna tarea de modo automático. Esta idea ha estado presente en la cultura occidental desde hace al menos dos siglos. De hecho forma parte del corazón mismo de la revolución industrial, que aplica esas máquinas al proceso productivo, a la fabricación de bienes.

Desde sus orígenes, la robótica ha tenido una motivación industrial que ha impulsado su desarrollo. De hecho, la mayoría de los robots que existen en la actualidad trabajan en fábricas, muchos de ellos en cadenas de fabricación de automóviles. Un hito paradigmático en este escenario son los brazos articulados tipo PUMA (*Programmable Universal Manipulator for Assembly*) de la figura 1.4. Se utilizan con frecuencia para soldar, pintar, transporte de materiales, etc. Las principales ventajas que ofrece un brazo robotizado frente a un operario humano para estas labores repetitivas son la precisión y la rapidez. Desde el punto de vista del empresario también hay que considerar que los robots no hacen huelga, no se aburren y pueden trabajar 24 horas al día, todos los días del año. Además, salvando cierta inversión inicial, su coste de mantenimiento puede ser menor que el equivalente para que la misma labor la realicen operadores humanos.

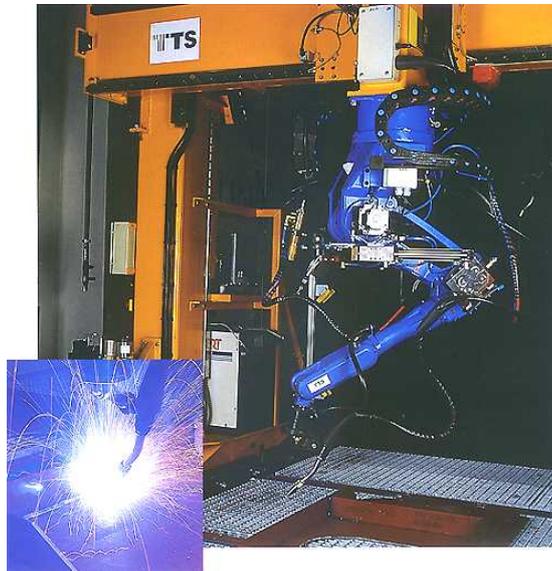


Figura 1.4: Robot de soldadura.

Los robots también aparecen en otros escenarios fuera de la fábrica. También se utilizan en entornos altamente peligrosos para las personas. Por ejemplo, se utilizan robots teleoperados para explorar zonas de alta radiación en el interior de reactores nucleares, para rastrear en las profundidades oceánicas, para sondear volcanes o pirámides, para inspeccionar y desactivar bombas o minas. En estos casos el robot suele estar equipado con cámaras de vídeo y es guiado remotamente por un operador humano.

En los últimos años está creciendo el uso de robots para tareas agrícolas. Las tareas de fumigación o cosecha son potencialmente automatizables porque son repetitivas y tediosas, e incluso conllevan cierto riesgo químico para el operario humano. Por ejemplo, hay robots recolectores de tomates, melones, pepinos o champiñones, que utilizan visión artificial para identificar el fruto adecuado. Otra aplicación relevante en esta línea son los cosechadores automáticos de cereales, que recorren de modo semiautónomo los inmensos campos de cereales segando y recogiendo el grano. Estos robots agrícolas suelen ser vehículos comerciales a los que se ha dotado de autonomía incorporándoles los sensores, los actuadores, los microprocesadores y la programación necesaria para la toma de decisiones.

Las exploraciones espaciales suponen otro campo de aplicación de la robótica. Se han utilizado robots en varias misiones. Por ejemplo, la misión Pathfinder puso al robot Sojourner en la superficie de Marte en 1997. El robot tenía 6 ruedas y básicamente se movía teleoperado desde la Tierra. Tomó varias imágenes, analizó el suelo y algunas rocas del planeta rojo, enviando los resultados a la Tierra. En esta línea la NASA ha invertido y sigue invirtiendo mucho en proyectos que desarrollan robots capaces de desenvolverse en entorno hostil tan lejano (como el programa de telerobótica espacial).

Uno de los campos insospechados por donde está creciendo la oferta robótica es el ocio y el entretenimiento. Por ejemplo, el perrito Aibo de Sony en la figura 1.5(a) se vende como mascota y los robots humanoides desarrollados por Honda, Toyota y Sony en los últimos años avanzan en esta línea. También el ladrillo de Lego [Noga, 1999] se vende como juguete imaginativo.

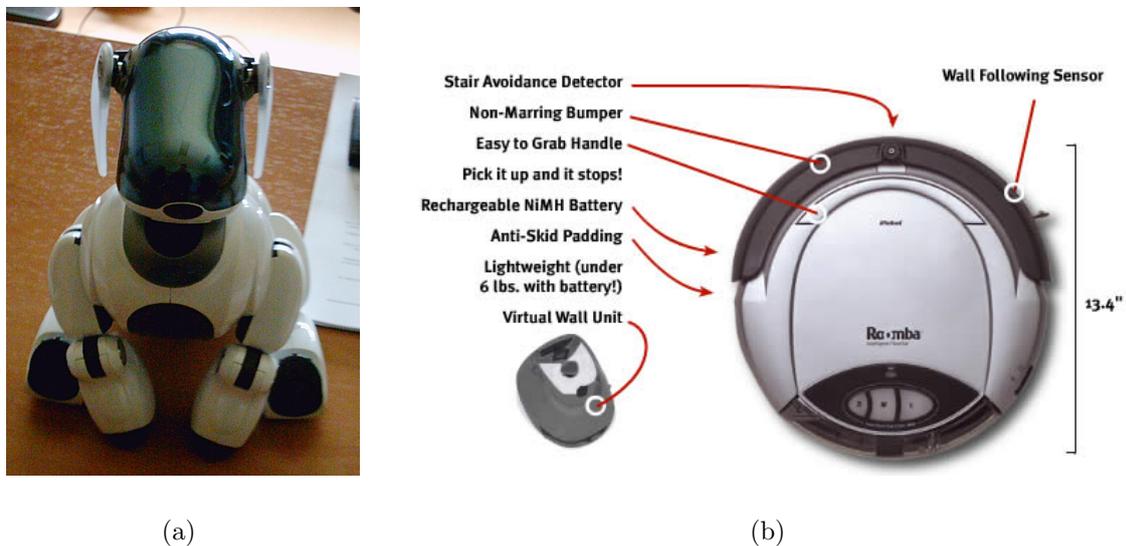


Figura 1.5: Mascota AIBO de Sony (a) y aspiradora Roomba (b)

Otros tímidos intentos se han dirigido a introducir los robots en el hogar para automatizar tareas domésticas. Una muestra es el aspirador robótico Roomba [iRobot, 2005] que deambula por una habitación de forma autónoma sin chocar con las paredes, tragando pelusas y deslizándose debajo de las camas y los sofás. Sus sensores evitan que choque contra las paredes o los muebles, o se caiga por el hueco de las escaleras.

Su precio es inferior a los 200 \$ y es un ejemplo de sistema robótico comercial que se puede ver en la figura 1.5(b).

El campo de la automoción es otro claro ejemplo de la influencia de la robótica en la industria y productos de naturaleza comercial. El campo de la automoción adopta características de la robótica para el desarrollo de sistemas de seguridad y comodidad en los vehículos, como los anteriormente comentados. La característica principal de estos sistemas incorporados a los vehículos es la autonomía en la toma de decisiones. Es decir, en la mayoría de estos sistemas presentados en la sección anterior: control de tracción, de estabilidad, etc. no es el conductor del vehículo el que decide cuando ha de ponerse en funcionamiento el sistema, sino que el propio sistema, gracias a microprocesadores y la programación de éstos, determina cuando ha de hacerse. Estas decisiones se toman en función de la información que capturan los sensores u otros parámetros que le sirvan de ayuda para tomar la decisión. Los sensores que suelen incorporar estos sistemas son ultrasonidos, radar, cámaras, etc. Con esto se dota de cierta inteligencia a los vehículos como sucede en los robots.

Pero no sólo la autonomía de estos dispositivos es una característica compartida con la robótica, otros elementos de estos sistemas como los sensores y actuadores también son reflejo de la influencia de la robótica en la automoción. Sistemas como el de ayuda al aparcamiento cuentan con sensores para percibir el obstáculo y determinar la cercanía a éste, ya sea a través de sensores de ultrasonido o visión trasera.

Todo esto en cuanto a los vehículos comerciales, en el caso de prototipos de investigación la influencia de la robótica es aún mayor, ya que en la mayoría de los prototipos se persigue la construcción de vehículos inteligentes que sean capaces de circular sin conductor (robot Klaus comentado anteriormente, etc.).

En general, el uso de los robots va creciendo paulatinamente, aumentando el rango de sus aplicaciones a medida que estos aumentan en autonomía y funcionalidad, debido a los avances en el campo de la electrónica y la informática.

1.3. SIA (Sistema Inteligente de Adelantamiento)

Además de estos avances en la automoción, realizados para facilitar la conducción y aumentar la seguridad, también hay veces que el hombre se ve obligado a desarrollar otros para adecuarse a cambios en las normas viales.

En la directiva europea 92/6 se establece una limitación de velocidad para autobuses y camiones pesados, así como la imposibilidad de circular si no están equipados con un dispositivo limitador. En concreto la directiva fija una velocidad máxima de 100 km/h para los autobuses pesados (categoría M3), y de 90 km/h para los camiones pesados (categoría N3).

Estos limitadores están diseñados para aumentar la seguridad vial, pero dificultan la fluidez del tráfico y la seguridad en al menos una situación: cuando un vehículo pesado circulando a una velocidad cercana a la máxima quiere adelantar a otro vehículo pesado transitando a una velocidad ligeramente inferior. En esta situación la maniobra se hace lenta, a pesar de que los motores actuales, sin limitador, pueden proporcionar la potencia necesaria para que la maniobra de adelantamiento se hiciera de manera más rápida.

En este escenario resulta útil el desarrollo de un dispositivo que permita identificar automáticamente tal situación y elevar temporalmente el límite, justo mientras se realiza el adelantamiento. Con ello se agilizaría la maniobra, ganando en fluidez del tráfico y en seguridad.

Un sistema que intenta resolver esta situación es el que se ha desarrollado en este proyecto fin de carrera. Este proyecto es un ejemplo de tecnología en vehículos. En él se han empleado elementos propios de la robótica como sensores para percibir las situaciones en las que se puede encontrar el vehículo, así como toma de decisiones autónomas.

Se trata de un proyecto de naturaleza comercial elaborado en la universidad Rey Juan Carlos para el INTA (Instituto de Técnica Aeroespacial), a través de mediación de la empresa Consultrans. El proyecto SIA es un proyecto real, con plazos de entrega y en el que existían reuniones periódicas con el cliente.

La organización del resto de la memoria es la siguiente: En el capítulo posterior se fijan los objetivos de este proyecto, así como los requisitos de partida que han condicionado el desarrollo del sistema. En el tercer capítulo se detalla la primera fase que consiste en el estudio de sensores adecuados en función de los requisitos. El cuarto capítulo se encarga de analizar y justificar el diseño elegido a la hora de realizar tanto el *hardware* como el *software*. En el quinto capítulo se profundiza en la descripción informática del sistema, comentando el *software* de bajo y alto nivel desarrollado. Las pruebas experimentales realizadas están comentadas en el sexto capítulo. Para finalizar, en el último capítulo se exponen las conclusiones extraídas del desarrollo del proyecto, así como las siguientes líneas de investigación en las que se podría continuar.

Capítulo 2

Objetivos

En este capítulo se definen los principales objetivos que se han perseguido al desarrollar este proyecto. Hay que tener en cuenta, además, que existen una serie de requisitos y tareas adicionales debido al carácter precomercial del sistema, que también deben ser alcanzados y a los que creemos interesante habernos enfrentado porque no es lo habitual en este tipo de trabajos.

2.1. Objetivos

La finalidad principal de este proyecto es el diseño, programación y construcción de un prototipo para la agilización de la maniobra de adelantamiento en vehículos pesados, ya que actualmente éstos tienen limitada electrónicamente la velocidad a 90 km/h, por la directiva europea 92/6. El sistema debe percatarse de la situación de comienzo (figura 2.1) y fin del adelantamiento, para subir temporalmente (hasta 110 km/h) y volver a bajar el límite legal de velocidad respectivamente. La construcción de un prototipo servirá para validar el diseño realizado y estimar la adecuación al producto final a comercializar.

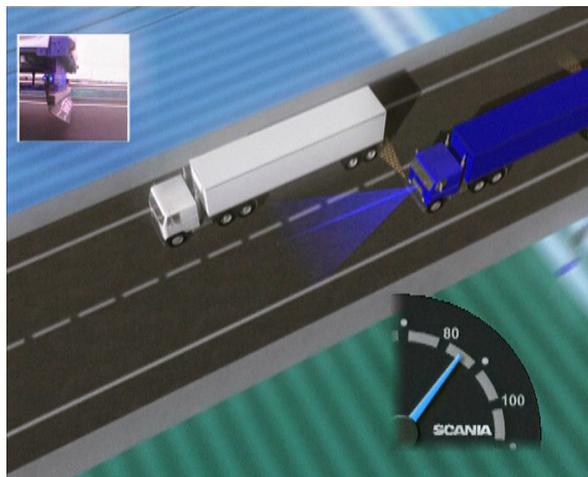


Figura 2.1: Sistema detectando el comienzo de un adelantamiento

Asímismo, se han identificado una serie de subobjetivos que se describen a continuación:

1. **Estudio sensorial:** En una primera fase de diseño es necesario estudiar de manera profunda los diferentes tipos de sensores de distancia disponibles en el mercado (ultrasonidos, láser puntuales, láser de barrido, láser de barrido multiplano, radares). Entre los múltiples factores a comparar destacan el rango disponible de medidas, la apertura angular (en caso de sensores de barrido), la precisión con la que miden; así como factores externos a los funcionales: Precio, intervalo de temperatura de funcionamiento, peso del sensor, disponibilidad, . . . que solo debido al carácter precomercial de la aplicación se han tenido que valorar para su correcta adaptación al proyecto.



Figura 2.2: Sensor radar Eaton Vorad que además de medir distancias es capaz de estimar la velocidad de los obstáculos hasta los 200 Km/h

Por tanto, un objetivo importante es la búsqueda de los mejores sensores de distancia que proporcionen un equilibrio entre funcionalidad y precio. En capítulos sucesivos se profundizará en las características de los dispositivos empleados.

2. **Diseño eléctrico y lógico:** Nuestro sistema está compuesto de varios elementos principales: Sensores, subsistema de gestión sensorial y ordenador de proceso. Todos ellos forman parte de la aplicación y, por tanto, tienen que residir en el prototipo donde se instale el sistema. Dicho prototipo es un vehículo pesado que únicamente contiene una fuente de alimentación: Su/s batería/s. La heterogeneidad de los dispositivos empleados necesita del diseño de una etapa previa de división de tensión, para que el conjunto de componentes reciba una apropiada alimentación e intensidad proveniente de las baterías del vehículo. A su vez, hay que realizar un diseño modular y dividido en subsistemas independientes que definirán la arquitectura lógica de la aplicación.
3. **Programación de bajo nivel y recepción de los datos sensoriales:** Una vez escogidos los sensores que nos van a ofrecer su particular visión del mundo, es tarea indispensable crear un subsistema intermedio de gestión para gobernar el funcionamiento de los sensores y gestionar su activación, ajuste y detención. En

particular, los sensores de ultrasonido ofrecen un interfaz de muy bajo nivel y es preciso diseñar y programar un circuito de control para los mismos, que permita al ordenador central una lectura más fácil y flexible de los datos. En cuanto al láser, sus datos están disponibles de una manera más cercana al programador, por lo que únicamente se han utilizado librerías *software* para requerir la distancia a los objetos desde el láser.

4. **Programación de alto nivel:** Después de diseñar la arquitectura general del sistema, hay que identificar y caracterizar todos los casos susceptibles de pertenecer al inicio y fin de adelantamiento, analizando los datos procedentes de los sensores. Una vez caracterizada la multitud de escenarios y entornos más frecuentes, hay que pasar a programar el sistema para que se activen las señales encargadas de modular el regulador de velocidad. Además, sería recomendable programar una aplicación para visualizar y chequear el correcto funcionamiento del sistema, de tal manera que se permitiera comprobar visualmente las distancias a objetos y desde donde se pudieran ajustar algunos parámetros sensoriales.

2.2. Requisitos

Se han identificado una serie de requisitos para que la aplicación funcione de manera adecuada y responda ante todas los casos que se pueden plantear de manera satisfactoria:

- El sistema será totalmente autónomo y no será necesaria ninguna intervención humana. De esta forma toda la atención del conductor permanecerá en la carretera.
- La aplicación estará protegida de picarescas, activándose automáticamente y evitando un control manual para aumentar la velocidad de manera voluntaria y sin causa justificada.
- Los sensores deben ser robustos a los cambios climáticos. Por ejemplo han de funcionar con niebla, lluvia, en altas y bajas temperaturas, etc.
- El funcionamiento debe ser correcto en todos los tramos que pueden presentarse en una carretera: Curvas, rectas, carreteras de un sentido, de dos, etc..
- La arquitectura escogida debe contener una serie de sensores conectados a un ordenador convencional a través de un medio de conexión altamente extendido y de fácil adquisición. El sistema operativo escogido para dicho ordenador ha de garantizar alta disponibilidad y robustez, pues el programa principal encargado de procesar los datos y que tomará todas las decisiones ejecutará sobre él.
- Un requisito importante es que los datos sean enviados al ordenador de control con la vivacidad suficiente para ser procesados en tiempo real. Además, no solo se exigirá que el envío sea rápido, sino que sea lo suficientemente rico en información como para decidir adecuadamente la tarea a realizar.

- El prototipo a desarrollar deberá ir empotrado en un vehículo pesado y se deberá tener en cuenta este aspecto a la hora de diseñar el sistema de alimentación (12V./24V.) y de elegir el sensor conveniente (temperatura de funcionamiento, tamaño, peso). Por ello los sensores elegidos deben exhibir la robustez típica de los equipos de exteriores: vibraciones continuas, temperaturas extremas, etc.
- Como en cualquier proyecto precompetitivo el coste económico es un condicionante importante, cara a la comercialización del sistema final. En este sentido se buscará un compromiso entre calidad y precio, superando cierto mínimo que garantice la funcionalidad del sistema.

2.3. Metodología

El modelo de desarrollo seguido ha constado de varias fases, empezando por una correcta definición y planteamiento del problema a resolver. A continuación se evaluaron las posibles alternativas sensoriales valorando sus prestaciones, precio y adecuación a la aplicación. Después se realizó el diseño global del proyecto estableciendo los diversos subsistemas que compondrían el prototipo final.

Pasada esta fase inicial se comenzó a trabajar en la implementación de la aplicación propiamente dicha y se dividió su realización en dos etapas:

- **Desarrollo *hardware* y *software* de bajo nivel**, necesario para que el ordenador de control tenga disponibles todos los datos con una frecuencia aceptable.
- **Desarrollo del *software* de alto nivel**, que analiza toda la información proporcionada por los sensores y realiza el análisis necesario para distinguir entre situaciones de adelantamiento o no.

A su vez, cada una de estas dos fases debe seguir un modelo en espiral ilustrado en la figura 2.3, que implique análisis, diseño, implementación, pruebas y validación, desembocando en la construcción de un prototipo. Asimismo al terminar la construcción de cada prototipo se debe documentar todo el proceso seguido.

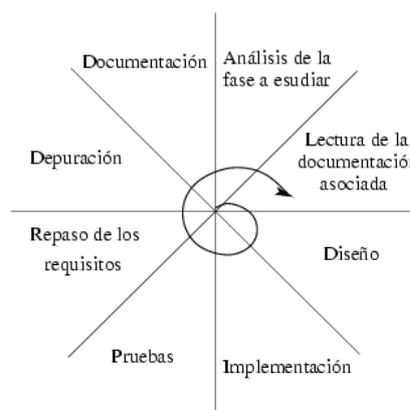


Figura 2.3: Metodología en espiral

Debido a la naturaleza comercial del proyecto, hay que tener en cuenta los hitos temporales que marca la empresa contratante y definir eficazmente la funcionalidad que debe incluirse en cada etapa o prototipo, así como planificar las demostraciones tanto en recintos interiores como en carretera exterior y las múltiples reuniones con el cliente.

Capítulo 3

Estudio Sensorial

El diseño inicial del sistema se divide en un sistema sensorial que ofrece información del entorno y un sistema de control, que en función de esa información toma decisiones (detectar inicio y fin de adelantamiento). El sistema sensorial se compone de los sensores y del sistema de gestión de estos, que se encarga de recopilar la información ofrecida por los mismos. Los sensores se disponen en el frontal del vehículo pesado para detectar el vehículo precedente a rebasar y en el lateral para detectar cuando se le ha rebasado por completo. De este modo en la figura 3.1 se puede observar la disposición de los sensores en el vehículo. Aunque todo este diseño del sistema se detalla en el siguiente capítulo, la decisión de qué sensores montar en el sistema tiene tanta importancia que le hemos dedicado un capítulo propio para el estudio de las distintas alternativas sensoriales.

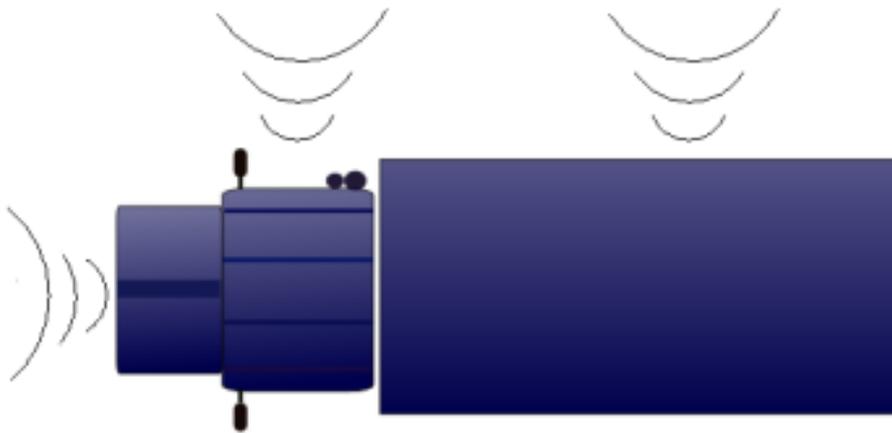


Figura 3.1: Esquema que muestra la situación de los sensores en el vehículo

Para que nuestro sistema funcione de modo autónomo y fiable es necesario dotarle de un conjunto de sensores que le permita discriminar la situación previa al adelantamiento, así como el momento en el que la maniobra se puede dar por terminada. Esto hace posible que el aumento del límite de velocidad tenga lugar *exclusivamente* mientras dura la maniobra, de modo que no se pervierta el espíritu de la limitación legal.

La autonomía de funcionamiento es un requisito esencial del dispositivo, que debe funcionar sin intervención alguna del conductor del vehículo, de modo automático y objetivo. En este sentido, la elección sensorial resulta crítica para la calidad del sistema, pues literalmente toda la información de que éste dispone para decidir aumentar el límite o no, proviene de los sensores. Esto es decisivo en la autonomía del sistema ya que debe tener una fuente de información del exterior buena y si ésta no es buena se tomarán malas decisiones.

Las necesidades sensoriales del sensor frontal son distintas a las del sensor lateral. El sensor frontal es el encargado de detectar el comienzo del adelantamiento y esto implica que debe detectar al vehículo precedente con anterioridad, y a una de entre 30 y 50 m. Por tanto, el sensor elegido para el frontal del vehículo debe tener un rango de medida de hasta 50 m. Además sólo debe detectar el vehículo que se encuentra delante y en el mismo carril, ya que es al que va a rebasar. El sensor lateral se usa para la detección del fin del adelantamiento, que se produce cuando se ha rebasado por completo al vehículo. Por lo tanto la finalidad de éste es que nos diga si hay o no vehículo en nuestro lado derecho. La distancia cuando se está rebasando a un vehículo es raramente superior a los 3 m., lo que hace que el rango de distancia del sensor lateral sea mucho menor que el frontal.

En cuanto a las tecnologías, el mercado de sensores ofrece diferentes opciones: visión ([Dickmanns, 1997]), sensores de distancia infrarrojos, ultrasonidos, láser ([Wang y Thorpe, 2002]), radar, etc.. Todas ellas han sido evaluadas y en este capítulo describiremos las que mejor se adaptan al proyecto. El criterio de elección de los sensores es que doten al sistema de buena información del entorno estableciendo a la vez un compromiso en el precio.

3.1. Sensores de ultrasonido

Los sensores de ultrasonido se usan fundamentalmente para medir distancias a objetos, y principalmente en aplicaciones de robótica móvil. Su funcionamiento consiste en la emisión de una onda acústica cuyo rebote con un objeto captura un micrófono, estimando así la distancia al objeto como se puede observar en la figura 3.2. El tiempo que tarda la onda acústica en retornar se denomina tiempo de vuelo y es proporcional a la distancia recorrida por ésta.

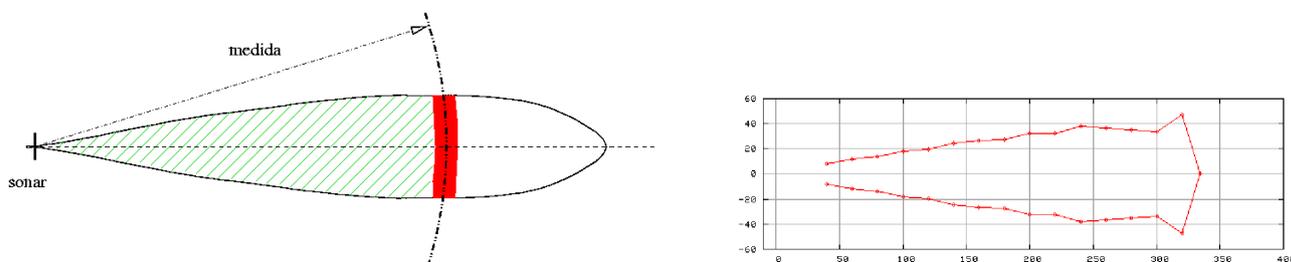


Figura 3.2: Interpretación de la lectura sónica (izquierda) y alcance experimental (derecha)

El inconveniente fundamental es la incertidumbre angular, ya que la onda emitida se va abriendo a medida que avanza en el espacio. Los rebotes también afectan negativamente a estos sensores, ya que se obtienen medidas irreales del entorno.

Dadas las necesidades del sensor frontal, el sensor de ultrasonidos no es el más indicado ya que su rango de distancia no es mayor a los 10 m. Además la incertidumbre angular no lo hace viable ya que detectaría otros objetos (medianas, señales, árboles, vehículos en otro carril, etc.), incluso cuando no hay vehículo al que adelantar. Aún así, quizá resulte conveniente para los laterales ya que en este caso la distancia a medir es menor y la incertidumbre angular no afecta de manera tan acusada.

Se realizó un extenso y profundo estudio sobre los modelos de sensores de ultrasonido existentes en el mercado. Se estudiaron en concreto el modelo 5000-S12 de *Cleveland Motion Controls*¹ que fueron descartados por su escaso rango de medida (15 cm. a 1.5 m.). También se estudio el modelo SR50 Ultrasonic Distance Sensor de *Campbell Scientific*² que posee un rango de medida de hasta 10 m. pero que fue descartado por sus grandes dimensiones (31 cm. de largo y 7.5 cm. de diámetro), además de por su elevado precio. Por estos motivos y dadas las necesidades de la aplicación, poco a poco se iban descartando la mayoría y se decantó la elección al *Polaroid Smart Sensor*, que se muestra en la figura 3.3.

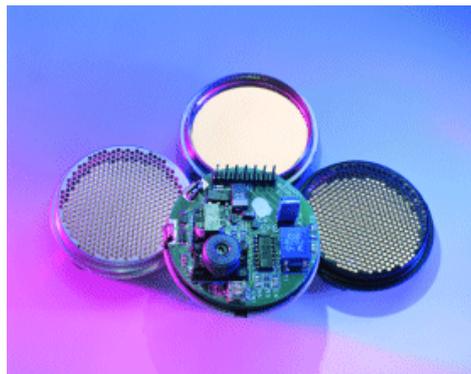


Figura 3.3: Polaroid Smart Sensor.

Este modelo de sensor ultrasónico cumple a la perfección las necesidades de los sensores laterales usados en la aplicación. El sensor lateral sólo debe detectar la presencia del vehículo al que se está rebasando pero no necesita saber la distancia exacta. Las características del *Polaroid Smart Sensor* son las siguientes:

- Un rango de medidas de 0.4 a 10 m., muy acorde con las necesitadas para la aplicación.
- Reducido tamaño entorno a los 4.5 cm. de diámetro y 2.7 cm. de espesor. Estas dimensiones lo hacen más flexible para su posible configuración como sensor lateral, sin necesitar hacer grandes modificaciones en los vehículos para su instalación.

¹<http://www.cmcontrols.com>

²<http://www.campbellsci.com>

- Un precio asequible (50 euros) que hace viable la comercialización del sistema a un precio competitivo.
- La capacidad de trabajo en exteriores y bajo condiciones climáticas extremas, gracias a un baño que les dota de impermeabilidad. Estas características las posee el modelo para trabajo en exteriores *Open Face Smart Sensor de Polaroid*.

El manejo de estos sensores consiste en la activación de un pulso de *INIT* para iniciar un ciclo de medida. Esta activación produce que el sensor emita la onda acústica. Cuando ésta regresa al rebotar con un objeto, el sensor activa un pulso de *ECHO*. El tiempo transcurrido entre los dos pulsos es proporcional a la distancia al objeto detectado. Si el objeto a detectar se encuentra fuera del rango de media del sensor (0-10 m.) o no hay objetos, el pulso de *ECHO* no se produce. Además de estas señales el sensor posee otras como el *BINH* que sirve para iniciar un ciclo de estabilización del sensor, es decir, para esperar a que el *hardware* esté estabilizado para realizar un nuevo ciclo de medida.

3.2. Sensores láser

Los sensores láser son sensores de distancia a obstáculos que han ganado cuota de mercado en los últimos años. Se usan en aplicaciones industriales para crear zonas de seguridad donde no se quiere que haya personas, etc. Sus fundamentos físicos son muy parecidos a los de ultrasonidos, pero empleando luz estructurada como onda que se emite y se recibe. Normalmente el rango de frecuencias que se utiliza es luz invisible al ojo humano.

Son sensores de gran precisión tanto radial como angular. En distancia llegan al orden de milímetros y en resolución angular llegan al medio grado y al cuarto de grado. Al utilizar la luz, la onda tiene una dispersión angular mínima. La distancia máxima a la que detecta un objeto depende del color de ese objeto en el que refleja la luz. Para su evaluación en este proyecto nos hemos fijado en el caso peor: en los rangos para objetos negros y con baja reflectividad, como las ruedas de los camiones. Hemos encontrado modelos que llegan en ese caso hasta 30 metros, que está dentro del rango de medida necesario para el sensor frontal.

Comparativamente son sensores caros, su precio es medio-alto, alrededor de los 4000 euros. Hay dos modalidades de sensores láser: los *puntuales*, que miden la distancia al obstáculo más próximo en una única dirección, y los *de barrido*, que miden la distancia a todos los obstáculos en un abanico de unos 180 grados. Los sensores de barrido tienen igualmente un único haz láser, y un espejo giratorio que permite orientarlo para barrer los 180 grados.

Los sensores puntuales fueron descartados debido a la reducida región que cubren. En nuestra aplicación es necesario obtener una información más amplia de distancia a obstáculos para cubrir todo el abanico casuístico al que nos enfrentamos. La alternativa de emplear varios sensores láser puntuales fue descartada debido a su elevado precio, pues una pareja de estos sensores ya cuesta más que un sensor de barrido (mucho más rico en información). Los sensores de barrido son dispositivos más completos y ofrecen mayor riqueza sensorial que los modelos puntuales. En la figura 3.4(b) se pueden observar las medidas obtenidas por el láser de barrido. Las medidas corresponden a una

sala donde hay una serie de objetos dispuestos en la pared lateral derecha. Los modelos de exteriores incorporan mejoras como calefacción y antivaho, así como amortiguación y carcasa estanca para protegerse de vibraciones y condiciones climatológicas adversas.

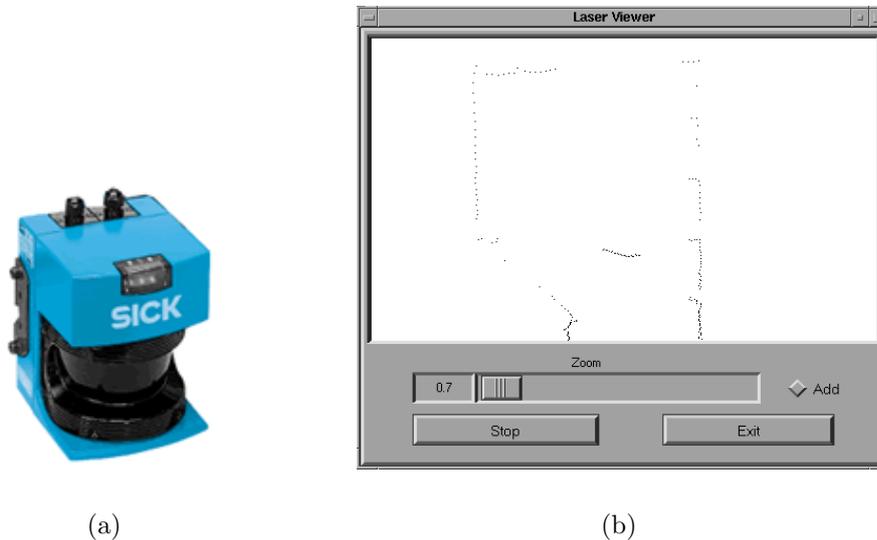


Figura 3.4: Imagen del sensor láser (a) y medidas obtenidas por un modelo de barrido (b)

Estos dispositivos industriales han sido analizados con detalle, porque nos parecen adecuados para las necesidades sensoriales del proyecto. Su conexión al ordenador principal es sencilla, a través de puerto serie, por lo que no requieren el uso de ninguna tarjeta especial de adquisición de datos, ni ningún subsistema intermedio como el caso de los ultrasonidos. La velocidad de transmisión del láser es configurable de 9600 a 38400 baudios. Entregan un barrido cada 100 milisegundos, frecuencia suficiente para la detección del vehículo delantero.

La desventaja principal de los sensores láser es su elevado precio. Posiblemente para un producto final, este elemento encarecería mucho la cuantía total. En nuestro caso, el fin es realizar un prototipo funcional y su incorporación sí es viable y adecuada.

Se estudiaron unos cuantos modelos de los disponibles en el mercado. Entre ellos el de Ibeo, en concreto su modelo *LD Automotive* que se trata de un láser avanzado usado en aplicaciones de automoción. Este láser barre en horizontal 180° y en vertical en 4 posiciones distintas. La desventaja de este modelo es su elevado precio que pasa de los 30.000 euros. Otros modelos analizados fueron los de la casa SICK, que presenta numerosos dispositivos láser. Todos ellos tienen una apariencia similar al de la figura 3.4(a). Entre ellos hemos analizado tres modelos especialmente adaptados a entornos duros de exteriores: LMS-211, LMS-221, LMS-291. Para nuestro proyecto hemos seleccionado el modelo LMS-221 por contar con excepcionales características para el uso en exteriores. Las características principales del modelo LMS-221 se resumen en la tabla 3.1.

dimensiones en mm (ancho, alto, profundo)	352 x 266 x 196
Laser Class	1
distancia máxima	80m
interfaz de datos	RS422 / RS232
velocidad de transferencia	9.6/19.2/38.4/500 kBaud
vida media	unas 50000 horas de funcionamiento

Tabla 3.1: Características del modelo láser LMS-221 de Sick

3.3. Sensores radar

El radar es considerado un sensor remoto, es decir, un elemento capaz de proveer información sobre objetos a distancias alejadas de él. Esta característica le hace apropiado para usarse como sensor frontal del sistema. La apariencia física de un sensor radar se puede observar en la figura 3.5.



Figura 3.5: Sensor radar

El radar tiene dos propiedades importantes:

1. Emplea una onda electromagnética que se encuentra en la porción del espectro correspondiente a la microonda, con longitudes de onda que van de 1 mm a 1 m.
2. Es de técnica activa, como el láser y los ultrasonidos, ya que emite radiación y luego percibe el objeto por medio de la radiación reflejada. Este hecho contrasta con los métodos "pasivos" que dependen de la fuente natural de radiación, como puede ser la emisión térmica o la luz visible.

Esta última característica representa una de las ventajas del radar: No depende de la fuente de radiación externa, dado que él ilumina el objeto de interés.

Consecuentemente el radar puede ser usado tanto de día como de noche. Las microondas penetran la niebla, nieve, lluvia y granizo, estos dos últimos en menor grado, es decir que funciona correctamente bajo condiciones climáticas adversas.

Puede medir largas distancias a objetos y determinar la velocidad de su desplazamiento, observando las distintas posiciones que toman en el tiempo.

La desventaja del radar es la limitada habilidad para resolver detalles del objeto, si se lo compara con los métodos ópticos y por supuesto la imposibilidad de percibir color, además del elevado precio de éstos. Otros motivos desfavorables son que no poseen una interfaz de programación y son de difícil distribución.

Todos estos motivos llevaron a descartar este dispositivo como posible sensor frontal. Los motivos son que encarecería el valor del sistema comercial, lo que no lo haría competitivo frente a otros y dificultaría el desarrollo al no poseer interfaz de programación.

3.4. Visión

Otra alternativa barajada fue el usar cámaras como sistema sensorial. Las cámaras dan una secuencia de imágenes en un determinado formato. Estas imágenes tienen gran volumen de datos por lo que es muy costoso su procesamiento. Dentro del contexto de nuestra aplicación no parece muy indicado su uso ya que el entorno de trabajo no está controlado y pueden darse numerosos casos. Estas situaciones pueden ir desde encontrar vehículos de diferentes morfologías, remolques cuadrados, planos, etc., hasta condiciones meteorológicas adversas, etc. Estos motivos hicieron descartar el uso de la visión para nuestra aplicación.

Capítulo 4

Diseño

Una vez planteado el problema a resolver, la construcción de un sistema inteligente de adelantamiento, se necesita elegir un diseño apropiado que cumpla con los requisitos especificados. Este capítulo expone los detalles de la fase de diseño del proyecto. En primer lugar se dará una visión global del sistema, distinguiendo los elementos más importantes del mismo. A continuación, desgranaremos las particularidades de los subsistemas que componen la aplicación. Para finalizar, se detallarán los últimos aspectos de la arquitectura global, donde se unen todas las piezas que componen el proyecto, y se describirán puntos más concretos como el diseño eléctrico del hardware construido.

4.1. Diseño global

Existen dos etapas fundamentales y bien diferenciadas en el proyecto. Por un lado, necesitamos una fase perceptiva, a través de la cuál recogemos la información del tráfico y los estímulos externos (intermitentes, velocidad del propio vehículo, etc.). Por otro lado, necesitamos procesar toda la información recibida y decidir en consecuencia de manera correcta y autónoma.

Para la etapa perceptiva, necesitamos un diseño lo suficientemente robusto como para abarcar la práctica totalidad de los casos con los que nos vamos a encontrar. En nuestro caso, se ha decidido repartir los sensores en dos áreas: Parte frontal y lateral derecha del vehículo. En la figura 4.1 podemos observar la disposición sensorial. Se ha creído conveniente colocar un sensor frontal para que identifique los vehículos y objetos antes de ser rebasados. Este sensor deberá tener un amplio rango de funcionamiento (30-60 m.) para poder detectar los vehículos con suficiente antelación y además, debe proporcionar bastante riqueza sensorial pues se va a encontrar con gran cantidad de objetos en su camino (vehículos, medianas, árboles, personas, etc.). Para la parte lateral, se utilizan varios sensores de menor coste y por tanto, menos ricos en información. El motivo por el que no se necesitan en este punto sensores más caros es porque el sensor frontal ya habrá puesto previo aviso a la aplicación de que comienza un adelantamiento. Por tanto, prácticamente solo es necesario sensores de barrera que detecten si hay un camión aún o por el contrario, ya se le ha adelantado.

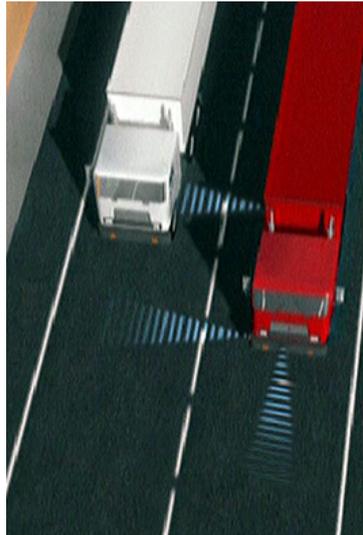


Figura 4.1: Mapa con las dos zonas cubiertas por los sensores

La segunda etapa de control o de proceso se ha resuelto utilizando un ordenador convencional que recibe toda la información procedente de los sensores. Este ordenador estará ejecutando un software de análisis de datos y actuará sobre el limitador de velocidad según sea conveniente o no. Es muy importante que los datos sean recibidos por el PC de manera vivaz, para que la aplicación funcione en tiempo real. Por último, también hemos creído conveniente que desde el PC se pueda modular el funcionamiento de los sensores. Así, la comunicación es bidireccional y desde el PC también se pueden enviar comandos de control a los dos subsistemas sensoriales (frontal y lateral) para ajustar alguno de sus parámetros.

4.1.1. Fase perceptiva

El diseño sensorial es muy importante ya que los sensores son los encargados de capturar la información relevante para que el sistema identifique si se está en situación de adelantamiento o no. Si el sistema sensorial no funciona correctamente nos encontraremos con falta de robustez en un escenario real y peligrosidad debida a la circulación de los vehículos a alta velocidad.

Por este motivo los sensores deben ser robustos y trabajar correctamente en condiciones adversas, etc. Se necesita un sistema sensorial rápido, es importante obtener de los sensores cuanta más información mejor y en poco tiempo. De este modo la decisión que tome el sistema dispondrá de más muestras sensoriales para extraer mejor las características del entorno donde se desenvuelve el vehículo en cada momento.

Además hay que tener en cuenta que este sistema irá empotrado en el vehículo y por lo tanto la alimentación del sistema saldrá del propio vehículo. Lo que implica que todos los elementos electrónicos utilizados en el sistema tendrán que poder alimentarse de la batería del vehículo.

Otro aspecto a tener en cuenta es que la naturaleza de este proyecto es comercial, por lo tanto el diseño sensorial también debe estar enfocado a que el sistema tenga un precio competitivo. Esto condiciona el tipo de sensores, *hardware* y otros elementos para el desarrollo del sistema.

Sensor frontal

El sensor frontal se utiliza para la detección del vehículo que será adelantado por el que lleva el sistema. Este sensor debe percibir características que identifiquen que hay un vehículo delante y próximo. Para este cometido pueden usarse sensores de ultrasonidos, cámaras, radar, láser, etc.

Los sensores de ultrasonido se barajaron pero fueron descartados debido a su corto alcance (máximo 10 m.) y la gran apertura angular con la que se propaga la onda emitida y que crearía gran ambigüedad, pues no se sabría con certeza la posición de los objetos.

El usar cámaras de vídeo para el sensor frontal implica que hay que procesar imágenes para extraer de ellas la característica relevante para esta aplicación, y ésta es un vehículo situado delante. Este procesamiento de imágenes a través de filtros, etc. implica gran capacidad de cómputo y mucho tiempo de proceso. Estos aspectos ralentizan el número de muestras en el tiempo, de manera que el sistema pierde vivacidad, cuando esta es imprescindible para hacer más robusto el sistema. No sólo esta cuestión es crítica a la hora de decantarse como sensor frontal, además existen otros condicionantes como el hecho de que no todos los vehículos tienen la misma fisonomía, lo que complica la extracción de una característica relevante e unívoca. Además de que las condiciones de visibilidad no serán siempre las más óptimas, puede llover, estar nublado, sol, noche, etc. lo que dificultaría la captura de imágenes al no distinguirse perfectamente el vehículo precedente. Estas desventajas desembocarían en una aplicación poco robusta y muy dependiente de aspectos externos. Aunque este tipo de sensores son bastante asequibles, no es aconsejable su uso como sensor frontal por los motivos esgrimidos anteriormente, por lo que se descartó su utilización.

Otro tipo de sensores son los radares, considerados sensores remotos, esto es, sistemas que son capaces de proveer información sobre objetos a distancias alejadas del equipo sensor. El radar puede ser usado tanto de día como de noche y en condiciones ambientales adversas lo que resuelve un inconveniente de las cámaras. Por contra, el coste de un radar capaz de medir distancias entorno a los 40 m. es muy elevado. La gran capacidad de cómputo necesaria para procesar la información obtenida por un sensor radar también es un punto en contra para estos sensores. Estos motivos también hicieron que descartásemos esta familia de sensores para la parte frontal de nuestro vehículo.

Por último otro tipo de sensores barajado para el diseño del sistema como sensor frontal fue el sensor láser. El láser es un sensor que emite un haz de luz para estimar distancias a objetos. Es mucho más preciso que los sensores de ultrasonido, ya que no emite una onda sino un haz. Existen diferentes configuraciones de sensores láser, sensores láser puntuales que emiten un solo haz, láser de barrido que barren una zona con varios haces, etc. El coste de este tipo de sensores es inferior al radar y se ajusta a las estimaciones de coste para el sistema. Además existen modelos diseñados para trabajo en exteriores que son estancos y poseen incluso un calefactor interno, lo cual les hace idóneos como sensor frontal para nuestra aplicación.

Sensor lateral

La función del sensor lateral es la detección de comienzo y fin de la maniobra de adelantamiento. Para lograr esta labor hay que colocar al menos dos sensores laterales dispuestos longitudinalmente en el flanco derecho del vehículo, tanto delante como detrás de éste. La necesidad de disponer de una batería de sensores con al menos dos unidades es para asegurar el correcto seguimiento del vehículo adelantado (pasando primero a ser detectado por los sensores laterales delanteros, después por los centrales, y finalizando con todos los sensores sin detectar obstáculo). El coste de cada uno de estos sensores debe ser asequible para el precio total del sistema.

Estos sensores deben medir un rango máximo de distancia entorno a los 6 m. ya que por lo general la distancia lateral entre vehículos en carretera no supera los 3 m. De esta manera se barajaron dos tipos de sensores de distancia: láser y ultrasonido.

El sensor láser, como se comentó antes, es muy preciso pero el coste de cada sensor y el tener que utilizar varios para el lateral del vehículo, hizo que se descartara éste como sensor lateral.

Por otro lado los sensores de ultrasonido poseen un rango de distancia de hasta 10 m., distancia más que aceptable para usarse como sensor lateral. Además, tienen un precio muy asequible y una capacidad de trabajo en condiciones climáticas adversas muy aceptable. La facilidad de proceso de la información capturada también es una ventaja, pues se trata de información simple (un número). También cabe destacar que su apertura angular era un problema cuando se utilizaba como sensor delantero, sin embargo, aquí es una ventaja. Al necesitar un sensor de barrera que indique si hay o no vehículo, el cono que se produce al propagarse la onda nos resulta beneficioso pues abarca mayor superficie de barrido y no se concentra en un único punto. Estas características hacen muy recomendable al ultrasonido como sensor lateral por lo que fue elegido para el proyecto.

4.1.2. Fase de análisis y control

Una vez diseñada una infraestructura para adquirir la información del entorno, necesitamos otra etapa donde se analicen los datos y se envíen las órdenes de control correspondientes. Esta fase se realiza en un ordenador convencional que deberá ir implantado en el vehículo. Hemos creído positivo el uso de un PC frente a una solución empotrada y a medida, ya que su precio es menor, su potencia es elevada y dispone de puertos y *slots* de conexión muy variados y ampliamente extendidos. En nuestro caso, hemos utilizado el protocolo RS232 sobre una pareja de cables serie para conectar los sistemas sensoriales frontal y lateral al ordenador. De nuevo nos hemos ayudado de un sistema de interconexión muy extendido, conocido, barato y funcional para comunicar ambos subsistemas.

Para una mejor depuración y ajuste de los sensores, también se ha de diseñar un interfaz gráfico que permita visualizar los datos recibidos por los sensores. Desde aquí se simularán los intermitentes y se mostrarán las señales de subida o bajada del limitador de velocidad. A su vez se utilizará también para enviar comandos de gestión a los subsistemas sensoriales y modificar su comportamiento.

4.2. Subsistema sónar

Los sensores de ultrasonido empleados como sensor lateral proporcionan una interfaz de utilización muy simple y escasa en información. Para iniciar una medida es necesario introducir una determinada tensión por una de sus patillas y esperar la terminación de la medición por otra. Es indispensable, por tanto, disponer de circuitería extra para contabilizar el tiempo de vuelo de la señal sonora y poder calcular la distancia al objeto. Además necesitamos un mecanismo para poder seleccionar el sensor adecuado de entre la batería de sensores colocados en la parte lateral del vehículo. Los objetivos de este subsistema serán el lanzamiento ordenado de los diferentes sensores, su posterior captura para el procesado final en el PC y el control de ciertos parámetros del conjunto de sónares.

Antes de llegar al diseño final del subsistema de control sónar se barajaron varias alternativas. La primera de ellas consistía en utilizar una lógica de control basada en puertas lógicas, registros, multiplexores, comparadores, etc. Esta solución no exigía aprendizaje extra pues el funcionamiento de estos elementos es sencillo y se conocía. El problema es la poca flexibilidad que se logra y la dificultad para realizar tareas complejas y no siempre similares. Por ejemplo, sería relativamente fácil conseguir que se dispararan periódicamente un número fijo de sónares, pero no lo sería tanto poder seleccionar de una manera flexible el número de sensores que queremos disparar periódicamente.

Otra alternativa posible era utilizar una tarjeta de adquisición de datos como la mostrada en la figura 4.2. Este tipo de tarjetas comerciales disponen de entradas donde conectar sensores y salidas desde donde se ofrecen los datos de una manera más amigable para ser estudiados por el ordenador central. Entre las ventajas que ofrecen encontramos un ahorro en el tiempo de desarrollo, al evitar tener que realizar el diseño eléctrico y funcional de la tarjeta, pues simplemente hay que comprarla. Como es de esperar también tienen una serie de desventajas: Principalmente el tiempo de aprendizaje para utilizar el software de manejo de la tarjeta y su elevado precio.



Figura 4.2: Tarjeta de adquisición PCI

Estas penalizaciones nos hicieron pensar que podríamos encontrar una solución mejor. El tiempo que podríamos perder en aprender a utilizar la tarjeta lo podríamos emplear en realizar un subsistema de gestión de sensores a nuestra medida, basado en un microcontrolador (el funcionamiento general de estos microprocesadores se explica en la siguiente subsección). Ganaríamos en flexibilidad frente a la primera solución basada en elementos digitales fijos, ahorraríamos en coste frente a la alternativa comercial y

además nos beneficiaríamos de la alta potencia de estos microprocesadores de bolsillo. Por citar alguna de sus virtudes destacan la incorporación del protocolo RS232 en el propio chip, contadores, entradas y salidas analógicas y digitales, interrupciones, etc. Por contra, perderíamos tiempo en aprender a utilizar el microcontrolador y en diseñar los aspectos eléctricos de la placa. Esta última alternativa nos pareció mas apropiada para el proyecto y nos decantamos por ella.

4.2.1. Microcontroladores

A continuación expondremos las principales características de los microcontroladores, para comprender el porqué de su utilización en la placa que gobierna los sensores de ultrasonido.

Un microcontrolador es un circuito integrado que contiene toda la estructura (arquitectura) de un microcomputador, o sea CPU, RAM, ROM y circuitos de entrada y salida. Algunos microcontroladores más especializados poseen además convertidores analógico digital, temporizadores, contadores, sistemas de comunicación RS232, salidas PWM, etc.

A continuación se explicarán los componentes principales que conforman la arquitectura de un microcontrolador:

- Una CPU (*Central Processor Unit* o Unidad de Procesamiento Central) que interpreta las instrucciones de programa.
- Una memoria PROM (*Programmable Read Only Memory* o Memoria Programable Solamente para Lectura) encargada de memorizar permanentemente las instrucciones de programa. Otros modelos de microcontroladores tienen memoria de programa de tipo EEPROM o FLASH
- Una memoria RAM (*Random Access Memory* o Memoria de Acceso Aleatorio) utilizada para memorizar las variables utilizadas para el programa.
- Una serie de LINEAS de E/S para controlar dispositivos externos o recibir pulsos de sensores, switches, etc.
- Una serie de DISPOSITIVOS AUXILIARES para su funcionamiento, como pueden ser generadores de reloj, buses, contadores, etc.



Figura 4.3: Ejemplo de microcontrolador Hitachi H8

El mercado de los microcontroladores es un mundo muy activo pues hay multitud de aplicaciones donde están presentes estos pequeños chips. La industria automovilística, la robótica e incluso los juguetes utilizan estos ordenadores que caben en la palma de la mano para realizar alguna tarea concreta. Existen multitud de fabricantes (Hitachi, Intel, Microchip [Microchip, 2001], . . .) y cientos de modelos distintos. En nuestro caso decidimos utilizar la familia de microcontroladores (*PIC*) de la empresa *Microchip* porque existe gran cantidad de documentación al respecto, tanto oficial como no oficial. Además, el fabricante proporciona un entorno de desarrollo gratuito denominado MPLAB [Microchip, 2004]. Existen multitud de microcontroladores de *Microchip*, por lo que prácticamente está asegurado que exista alguno con las características deseadas para un determinado proyecto. Debido a la amplia difusión de los *PIC's* es más fácil también encontrar el programador necesario para descargar el código desarrollado en el chip.



Figura 4.4: Detalle del programador empleado en el proyecto SIA

4.3. Subsistema láser

El subsistema perceptivo está formado por dos tipos de sensores: Los sensores de ultrasonido, que se reparten por el lateral del vehículo y el sensor láser de barrido, que se sitúa en la parte frontal del mismo. En esta sección explicaremos el diseño realizado para que las medidas captadas por el láser sean recibidas en el PC para ser procesadas.

A diferencia de la interfaz de funcionamiento ofrecida por los sensores de ultrasonido, el sensor láser empleado (figura 4.5) proporciona una interfaz de utilización mucho más amigable y de alto nivel. A través de un cable serie se envían comandos de gestión, para que el sensor envíe de manera continuada medidas hasta el PC. De la misma manera se le puede solicitar al sensor que detenga o comience su funcionamiento.



Figura 4.5: Modelo láser empleado en el proyecto SIA

Todos estos comandos se han encapsulado en una biblioteca de alto nivel escrita en C. A continuación se muestran las cabeceras de las principales funciones contenidas en la biblioteca:

```
void sick_stop_continuous_mode(sick_laser_p laser);
void sick_start_continuous_mode(sick_laser_p laser);
void sick_start_laser(sick_laser_p laser);
void sick_stop_laser(sick_laser_p laser);
void sick_handle_laser(sick_laser_p laser);
```

La aplicación de control que reside en el PC y gobierna los datos y el interfaz gráfico que los muestra en pantalla, incluyen esta biblioteca en su código para acceder a los datos del láser. Los datos recibidos en cada haz de medidas son enviados en una estructura que contiene 180 medidas (una medida por cada grado barrido por el láser). De esta manera, los datos ya están disponibles para su análisis y visualización tanto gráfica como numérica.

4.4. Unidad de proceso

El control de la aplicación tiene como objetivo el recopilar la información capturada por los sensores, el procesado de esta información para extraer las características relevantes del entorno y posteriormente la toma de decisiones que incluye activar un actuador como es el limitador de velocidad. Como añadido y sólo debido al carácter de prototipo también es importante disponer de un interfaz gráfico donde visualizar la información recibida y simular las decisiones tomadas (subidas/bajadas del limitador de velocidad). En el capítulo siguiente se detallarán los aspectos relativos a la programación del *software* de análisis de los datos y de su visualización.

La aplicación de control para este prototipo es una versión preliminar, que controla un caso sencillo de adelantamiento. Este caso sencillo consiste en un adelantamiento en línea recta y a un único vehículo. El diseño para esta versión preliminar se divide en dos

partes claramente diferenciadas: una parte perceptiva de recepción y procesado de los datos sensoriales y una de control para analizar estos datos y actuar en consecuencia.

La parte perceptiva tendrá como objetivo recabar información del entorno para saber en cada momento en qué situación se encuentra el vehículo. Esta información será si hay un vehículo que precede al que incorpora el sistema, en el caso de la maniobra de comienzo de adelantamiento (habrá que buscar perfiles rectos que se asemejen a camiones entre los objetos detectados, eliminando ruido y aplicando técnicas de segmentación). En el caso de fin de adelantamiento habrá que identificar si hay obstáculo o no en el lateral derecho, es decir, si aún estamos rebasando a un vehículo.

La parte de control tiene como misión establecer las órdenes de control sobre el limitador. Estas órdenes dependen en cada momento de la información obtenida en la fase perceptiva.

Además teniendo en cuenta la naturaleza de la maniobra de adelantamiento, éste se divide en dos fases claramente diferenciadas. Una fase inicial en la que se estima si hay que adelantar porque existe un vehículo precedente, a velocidad más baja que la del vehículo que va a rebasarlo. Y una fase final en la que se ha rebasado por completo al vehículo y por lo tanto se puede concluir el adelantamiento. De este modo no se puede dar la situación de fin del adelantamiento si antes no se ha producido la de inicio. Y para percatarnos del inicio hace falta observar que hay un vehículo precedente más lento.

De esta consideración se extrae la conclusión de que no es necesario observar continuamente si se ha rebasado ya al vehículo que se va a adelantar si aún no se ha comenzado la maniobra de adelantamiento. Por este motivo se debe recibir siempre información sensorial del láser para detectar el vehículo precedente al que adelantar. Para de este modo percatarse el sistema de la situación de comienzo del adelantamiento. Y por contra no es necesario que este continuamente activo el sistema encargado de los sensores de ultrasonido si aún no ha comenzado el adelantamiento. Estos sensores están para estimar cuándo se ha rebasado por completo al vehículo que se está adelantando y por lo tanto, se está en situación de fin del adelantamiento.

Teniendo en cuenta las consideraciones anteriores, el diseño global de la aplicación de control se basa en un autómata finito de estados. Este autómata consta de varios estados que se corresponden a las situaciones de detección frontal de un vehículo, detección lateral del vehículo a rebasar y fin del adelantamiento. El paso de un estado a otro viene provocado por las distintas situaciones que hay en la maniobra (detección o no de un vehículo precedente, etc.). Este autómata se explica en detalle en el siguiente capítulo.

4.5. Arquitectura final

A lo largo de este capítulo y el anterior hemos descrito cada uno de los elementos que forman parte del sistema (sensores, subsistema de control s3nar, subsistema de control de la aplicaci3n). Ahora es el momento de dar una visi3n global de la aplicaci3n y desde el punto de vista del dise1o, explicar c3mo se unen todas estas piezas que por s3 solas no cumplir3an la tarea final.

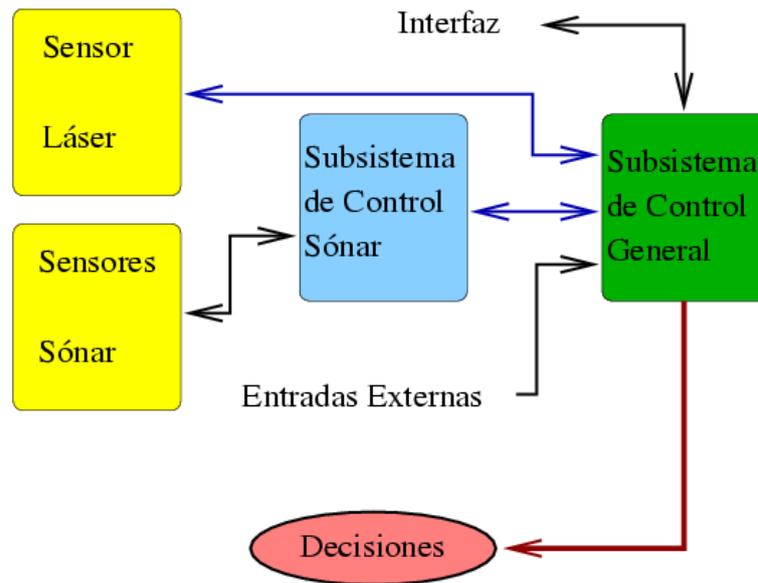


Figura 4.6: Unidades funcionales que componen la aplicación

La figura 4.6 ilustra la arquitectura global del proyecto. Las cajas amarillas representan los sensores. El sensor láser se conecta al subsistema de control general por medio de cable serie RS232 (flecha azul). De la misma manera, el PC y el subsistema de control sónar utilizan el mismo medio de interconexión. El objetivo que se persigue es que el diseño sea lo más sencillo que se pueda y en la medida de lo posible esté formado por elementos estándar y que podamos encontrar habitualmente, de ahí el utilizar un PC convencional y cable serie RS232. Los sensores sónar se enchufan directamente a la placa que gobierna su funcionamiento por medio de un cable apantallado, para minimizar las interferencias.

El funcionamiento de los sensores de ultrasonido debe ser configurable desde la interfaz gráfica. Esto permite mayor flexibilidad para hacer las pruebas y además, ayuda en la fase de depuración y corrección de errores. Se ha diseñado un protocolo de control entre el subsistema de control sónar y el PC. Así, es posible iniciar/detener el disparo de todos sensores o pausar un determinado sensor si se desea. A continuación se muestra una tabla con los comandos especificados en el protocolo.

Debido a que lo que se pretende es construir un prototipo, es recomendable diseñar también un interfaz gráfico desde el que se pueda observar el funcionamiento de la aplicación. Desde este interfaz también se podrá interactuar para regular el funcionamiento de todos los sensores y comprobar la exactitud de las medidas entregadas.

Además el ordenador principal deberá tener en cuenta otras entradas externas que serán simuladas y que representarán intermitentes, velocidad del vehículo, etc., en definitiva, factores que influyen en la decisión final de elevar la limitación de velocidad pero que son externos a nuestra aplicación.

El subsistema de control general deberá tomar todas las entradas y hacer un análisis de casos para tomar la decisión adecuada. En nuestro prototipo esta decisión se simulará y se mostrará en la interfaz gráfica.

<i>Comando</i>	<i>Descripción</i>
Encendido	Comando que activa el sistema de sensores para que envíen medidas, en el caso que haya algún sensor activo.
Apagado	Comando que desactiva el envío de medidas al PC
Activación selectiva	Comando para activar un sensor de ultrasonidos concreto. Hay que proporcionar el número de sensor a activar.
Desactivación selectiva	Comando para desactivar un sensor de ultrasonidos concreto. Hay que proporcionar el número de sensor a desactivar.
Reset	Comando que reinicia todo el sistema sensorial s3nar.
Configuraci3n	Comando que sirve para que el PC solicite la configuraci3n del sistema de sensores de ultrasonido (frecuencia de reloj y <i>slots</i> de la placa con sensores conectados).

Tabla 4.1: Comandos de control

<i>Sintaxis</i>	<i>Funci3n</i>
Envío de datos	Envío de una medida de un sensor de ultrasonidos del microcontrolador al PC. En la informaci3n enviada est3 contenido el identificador del sensor al que corresponde la medida.
Envío de configuraci3n	Comando utilizado para que el PC reciba del microcontrolador la frecuencia de reloj a la que trabaja y el número de <i>slots</i> de la placa que tienen un sensor conectado.

Tabla 4.2: Comandos de datos

4.6. Diseño el3ctrico

Una vez decidido que el cerebro del subsistema s3nar es un microcontrolador, necesitamos especificar las característic3s que debe reunir la placa de control de los ultrasonidos. Ésta constar3 de varios elementos diferenciados y con funciones independientes:

- **Microcontrolador:** Lleva el control de la secuencia de disparo de los s3nares. Es capaz de recibir comandos de control desde el PC para modular el funcionamiento de los sensores. Asimismo envía las medidas de los ultrasonidos al ordenador de control. Todo esto debe lograrse de manera vivaz y sin suponer un cuello de botella para el sistema.
- **Z3calos de conexi3n para los sensores:** Se ha incluido en el diseño una zona con varios conectores para enchufar los sensores que se desean emplear. En nuestro caso vamos a utilizar cuatro sensores como máximo, pues creemos que es un número suficiente para cubrir el flanco de un vehículo pesado.
- **Z3calo de conexi3n con el PC:** La comunicaci3n con el ordenador se realiza usando un cable serie y el protocolo RS232. La tecnología empleada por el microcontrolador es TTL (5v.), sin embargo, el puerto serie utiliza $\pm 12v$. Es necesario incluir una pastilla que realice el ajuste de las tensiones. El chip empleado es un Maxim 232 al que se le han ańadido cinco condensadores como sugieren en la secci3n de instalaci3n del fabricante [Maxim, 2004].

- **Electrónica de alimentación:** La alimentación de la placa es un punto importante a diseñar puesto que se incorporan varios elementos y todos ellos con distintos requerimientos eléctricos. En este punto, hay que señalar que es necesario alimentar los sensores de ultrasonido, el microcontrolador y la circuitería extra que es necesaria (multiplexores, decodificadores, etc.). Los sónicas requieren una elevada intensidad de pico para su correcto funcionamiento y una tensión diferente a la necesitada por el microcontrolador por ejemplo. Hay que realizar una etapa de estabilización de tensión y hacer llegar a todos los elementos del subsistema la tensión e intensidad correctas. Para la alimentación de la placa se ha utilizado un regulador 7805 que proporciona una tensión de salida de 5 voltios, suficiente para el microcontrolador y los componentes electrónicos. Para alimentar los sensores de ultrasonido hemos empleado otro regulador, en concreto el modelo 78H12, que consigue entregar hasta 5 Amperios en régimen continuo. De este modo aseguramos la correcta excitación de las membranas de los sensores y disponemos de un margen de seguridad suficientemente amplio.



Figura 4.7: Detalle de los reguladores y bornas de alimentación

La utilización de esta etapa previa de adecuación de tensión a cada elemento persigue varios objetivos:

1. Adecuar la tensión de entrada a las necesidades de la placa que contiene el microprocesador y los sónicas.
 2. Permitir un rango variable de tensión de entrada para que el diseño sea lo más genérico posible y pueda funcionar tanto en camiones con una batería (12 V.) como con dos (24 V.).
 3. Proporcionar una alimentación estabilizada y libre de ruido o picos de tensión provocados por el encendido u otras causas.
 4. Servir como elemento intermedio de protección para evitar la rotura de la electrónica o los sónicas.
- **Electrónica extra:** Hemos utilizado un oscilador de cuarzo de 4MHz para fijar la velocidad a la que ejecutarán las instrucciones en el microcontrolador. Este oscilador externo permite mayor estabilidad y velocidad que un generador de

señal interno. También hemos empleado multiplexores y decodificadores para poder direccionar cada uno de los sensores pinchados en la placa y recoger la señal de *ECHO* adecuada en cada caso.

Las figuras 4.9 y 4.8 muestran una fotografía de la placa construida y el diseño esquemático de la misma.

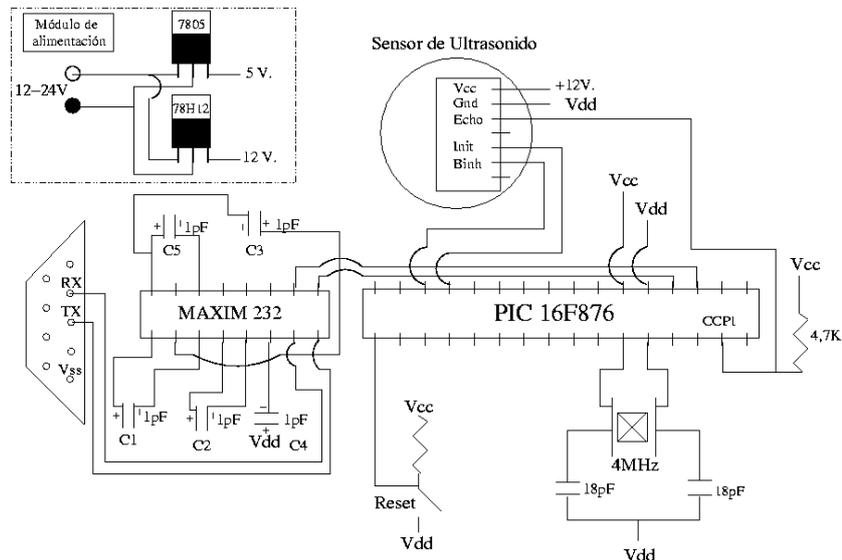


Figura 4.8: Diagrama esquemático de la placa de control s3nar

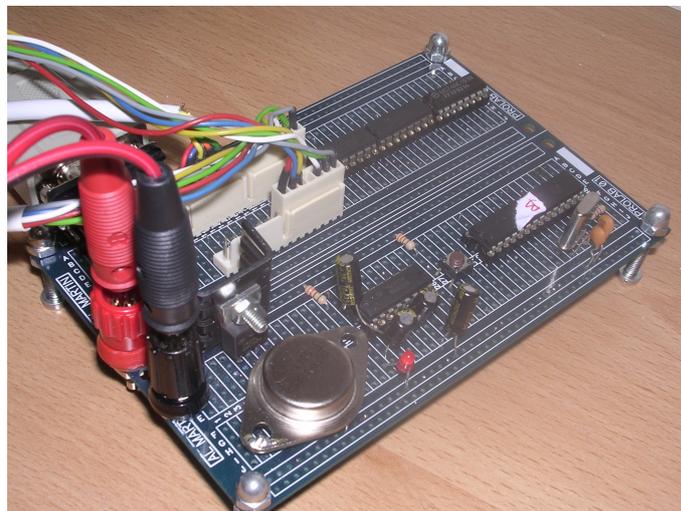


Figura 4.9: Fotografía de la placa de control s3nar

Capítulo 5

Descripción informática

Si en capítulos anteriores hemos descrito el diseño *hardware* del sistema, en el presente capítulo abordamos todos los aspectos relacionados con la programación del proyecto. El *software* se divide en tres programas diferentes: Código del PIC, Monitor y aplicación de control. En este capítulo detallaremos cómo hemos programado el microcontrolador que gestiona el funcionamiento de los sensores de ultrasonido, el Monitor que permite visualizar e interactuar con los sensores y el programa que controla las decisiones actuando sobre el limitador de velocidad. Estos dos últimos programas comparten la sección de captura sensorial, ya que ambos necesitan disponer de los datos capturados por los sensores. La programación de la aplicación se puede dividir en dos partes: Bajo nivel (orientado a la captura sensorial) y alto nivel (orientado a la visualización y el control). El microcontrolador se ha programado en ensamblador para maximizar la velocidad y favorecer la vivacidad del sistema. El monitor y la aplicación de control se han programado en lenguaje C, ya que necesitamos código de alto nivel y rápido. Este lenguaje, aunque carece de orientación a objetos y de las estructuras de datos más elaboradas de los últimos lenguajes, como podrían tener Java o C#, ofrece una elevada velocidad que resulta indispensable para una aplicación de tiempo real como la nuestra. Además, se disponía de algunas bibliotecas escritas en C y C++ para el control del láser y del puerto serie, que ayudaron en la decisión de elegir C como lenguaje, gracias a la posibilidad de reutilizar dicho código.

Estos tres programas (código del PIC, Monitor y aplicación de control) conforman la implementación informática del proyecto y en las siguientes páginas se desgrena cada

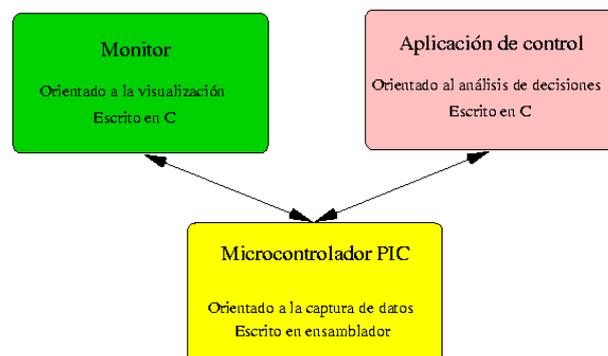


Figura 5.1: Programas que componen el proyecto SIA

uno de ellos.

5.1. Subsistema sónar

Según vimos en el capítulo de diseño, la interfaz de utilización del sensor sónar es muy cruda, muy física, muy de bajo nivel. Las medidas de los sónares han de llegar al PC para su posterior análisis y para lograr este objetivo se ha construido la placa que gestiona dichos sensores.

El microcontrolador PIC 16F876 es el cerebro de esta placa de adquisición de datos. Es capaz de recibir peticiones de toma de medidas por parte del PC que controla toda la aplicación, de manejar los ultrasonidos para iniciar medidas, de enviar los datos obtenidos al PC, etc.

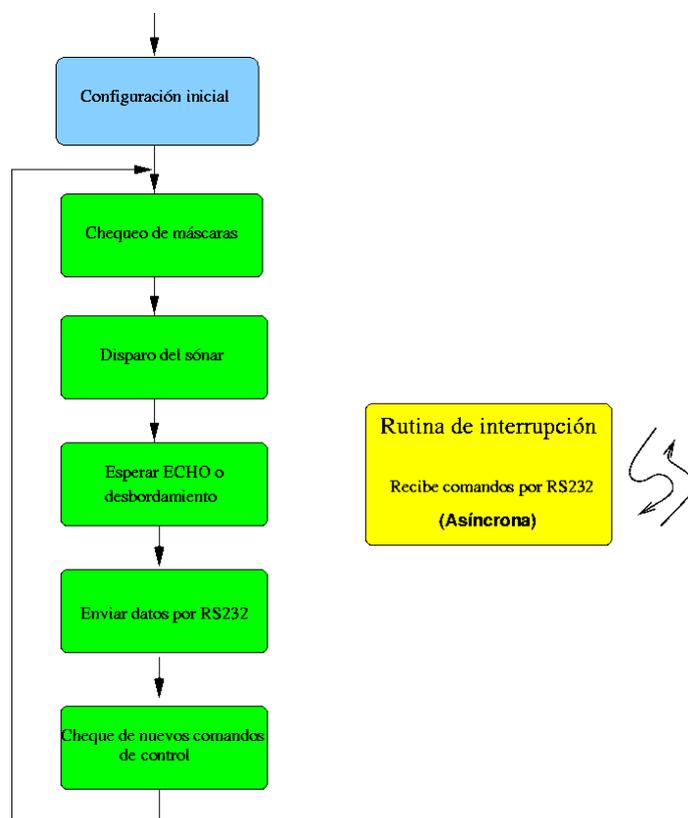


Figura 5.2: Diagrama de flujo del programa del PIC 16F876.

Las tareas principales que realiza el programa contenido en el PIC, cuyo diagrama de flujo se muestra en la figura 5.2, son las siguientes:

1. Configuración del PIC: Inicializar elementos propios del microcontrolador como contadores, interrupciones, etc.
2. Chequeo de máscaras: Existen diferentes máscaras que marcan si un determinado sensor ha de ser disparado o no (por ejemplo si se ha pausado su funcionamiento desde el Monitor). Antes de iniciar una medida se chequea el estado de estas

máscaras individuales y también el estado del interruptor general, que activa o desactiva el funcionamiento global del subsistema sónar.

3. Inicio de medidas: Disparar secuencialmente la señal de inicio de los sensores pinchados en la placa.
4. Recepción de ECHO o de desbordamiento: Percibir la señal de ECHO y manejar correctamente el contador asociado al tiempo de vuelo de la onda. También se gestiona el desbordamiento del contador en caso de sobrepasar un tiempo de espera máximo y no haber recibido la señal de ECHO correspondiente.
5. Envío de datos al PC: El microcontrolador emplea un protocolo de comunicación que se ha creado, para hacer llegar los datos al PC cada vez que una medida está disponible.
6. Gestión de comandos: El microcontrolador acepta comandos de gestión enviados desde el PC. Cuando se recibe alguno de ellos se genera una interrupción. Se ha realizado una rutina de tratamiento de interrupción que extrae el comando enviado y lo almacena para su posterior tratamiento.

Todas estas tareas programadas en el microcontrolador PIC se describen con detalle en los siguientes apartados.

5.1.1. Configuración del PIC

Como vimos en el capítulo 4, el PIC 16F876 dispone de gran cantidad de elementos útiles para la gestión del sistema sónar. Contiene entre otros elementos tres temporizadores programables, una unidad de conversión analógico-digital, interrupciones y una unidad USART para comunicación bidireccional a través de puerto serie. Antes de programar el PIC para que realice las tareas adecuadas, es necesario configurar algunos de estos elementos para ajustarlos a las necesidades concretas de la aplicación.

En concreto, nuestro programa utiliza la unidad de puerto serie para recepción de órdenes por parte del PC y contadores para estimar el tiempo transcurrido desde que se disparó un ultrasonido hasta que se recibió su rebote (o bien venció el plazo máximo de espera). También han de habilitarse las interrupciones para la recepción vía puerto serie (el flujo de ejecución saltará a la rutina de tratamiento donde se extraerá el comando y se almacenará para su posterior tratamiento) y los pines correspondientes de entrada/salida para comunicarse con el resto del *hardware*. Por último han de inicializarse las variables manejadas en el programa. A continuación se detalla la configuración de estos elementos. El código exacto está disponible en el CD adjunto a esta memoria.

En primer lugar, se configuran los puertos del PIC estableciéndose los pines RC7 y RC2 (como se puede ver en el esquema 4.8 del capítulo anterior) como entradas, por los que se conectará la señal de reloj externo y la señal de interrupción externa (la de recepción del puerto serie) respectivamente. El resto de puertos (RA, RB) se definen como salidas para comunicación con el resto del *hardware* (MAX232, multiplexor, etc.).

```

BANKSEL TRISA
movlw 0x00
movwf TRISA ; Se configura RA como salidas
BANKSEL ADCON1
movlw 0x06
movwf ADCON1 ; Se configura RA como puerto digital

BANKSEL TRISB
movlw 0x00
movwf TRISB ; Se configura RB0 como salidas excepto
; RB0 (INT externa)

BANKSEL TRISC
movlw 0x84 ; Se configura RC7 y RC2 (CLK y CCP1)
; como entrada y el resto como salidas
movwf TRISC
BANKSEL ADCON1
movlw 0x07 ; Puertos digitales
movwf ADCON1

```

A continuación se seleccionan los parámetros de funcionamiento de la unidad USART para su comunicación RS232, poniéndola en modo asíncrono y en modo *alta velocidad* para comunicación a 9600 baudios. Seguidamente se activa esta unidad para funcionar en modo continuo.

```

BANKSEL TXSTA
bsf TXSTA, TXEN ; Se activan las transmisiones
bcf TXSTA, SYNC ; Se configura en modo asíncrono
bsf TXSTA, BRGH ; Se configura en alta velocidad

BANKSEL RCSTA ; Se cambia al banco 0
bsf RCSTA, SPEN ; Se activa el puerto serie
bsf RCSTA, CREN ; Modo de recepción continua
bcf RCSTA, ADDEN

BANKSEL SPBRG ; Se cambia al banco 1
movlw 0x19 ; Velocidad USART: 9600 baudios
movwf SPBRG

```

También hay que establecer las interrupciones que nuestro programa necesita manejar. Es necesario habilitar las interrupciones globales y las de periféricos para que pueda producirse la de recepción del puerto serie (que también se habilita). A pesar de que únicamente se va a utilizar una interrupción (recepción serie) es necesario activar todas las interrupciones anteriores. La alternativa al uso de interrupciones para recibir datos es utilizar espera activa. La desventaja de la espera activa es que es bloqueante y no permitiría que continuaran disparándose los sensores de ultrasonido hasta que no se recibieran los datos por parte del PC. En nuestra aplicación esta técnica es totalmente inviable pues la principal prioridad es el disparo de los sensores. La aportación de las

interrupciones es que permiten una ejecución normal y continuada de ciclos de medida sónar y sólo en el momento de recibir un comando, la ejecución salta a la rutina de tratamiento. Aquí se extrae el comando y se devuelve la ejecución al punto donde se quedó antes de la interrupción. De esta manera, el tiempo empleado en tratar la interrupción es mínimo y el aprovechamiento del tiempo de proceso es máximo. Para el envío de los datos percibidos al PC no es necesario el uso de interrupciones pues se conoce el momento exacto en el que hay que realizar el envío. Esta tarea se realiza justo después de recibir el ECHO o al vencer el tiempo de espera máximo acordado, como se indicó en la figura 5.2.

```

BANKSEL INTCON
bsf     INTCON,GIE      ;activación Interrupciones globales
bsf     INTCON,PEIE    ;activación Interrupciones periféricos
BANKSEL PIE1           ;configuración Interrupción Rx
bsf     PIE1,RCIE      ;activación Interrupción Rx

```

En esta primera fase hay que ajustar un contador encargado de medir el tiempo de vuelo de la onda emitida por el ultrasonido periódicamente. Por tanto, se activa el TIMER1 de 16 bits configurándolo con una resolución de 65 mseg. por cada pulso de reloj. La velocidad de éste está determinada por el oscilador de cuarzo externo incluido en la placa desarrollada. Esta velocidad es fija y la inicialización de los contadores es la adecuada para esta velocidad. En caso de alterar la velocidad del oscilador habría que variar la configuración del contador. A pesar de que la velocidad del reloj es fija, los contadores permiten emplear un divisor de frecuencia para contar a un ritmo más lento que el marcado por el cristal de cuarzo. En nuestro caso no hemos utilizado el divisor de frecuencia o *prescaler*. La resolución de 65 mseg. empleada permite que el tiempo de desbordamiento del contador coincida aproximadamente con el tiempo de vuelo máximo percibido por el sensor de ultrasonido. De este modo cuando se desborda el TIMER1 se sabe con seguridad que no se ha recibido ECHO y por tanto, la distancia máxima hasta un objeto se ha sobrepasado.

```

BANKSEL T1CON
clrf    T1CON          ;TMR1 apagado,Prescaler = 1
                          ;tiempo captura= 65 msec.

```

El siguiente elemento a inicializar es la unidad CCP (*Capture Compare PWM module*) del microcontrolador. Esta unidad es útil para nuestra aplicación y agiliza algunas tareas. Dispone de tres modos de funcionamiento: Modo *captura*, modo *comparación* y modo *PWM*. El modo *captura* permite congelar el valor del TIMER1 cuando se produce un evento por el pin RC2. Este evento puede ser un flanco de subida/bajada o el cuarto o decimosexto flanco de subida por ese pin. La señal de ECHO de los sónares se ha conectado al pin RC2 y se ha configurado el evento como flanco de subida. Así, conseguimos congelar de una manera automática y rápida el valor del temporizador que marca el tiempo de vuelo cuando se recibe el ECHO.

El modo *comparación* permite cotejar el valor del temporizador con un valor predeterminado y actuar sobre el pin RC2 si la comparación es positiva. El modo de funcionamiento *PWM* (*Pulse Width Modulation*) se emplea para controlar ciertos

dispositivos (por ejemplo algunos motores) cuyo control se realiza variando la anchura de los pulsos emitidos hasta los mismos. Estos dos últimos modos no se han utilizado en nuestro programa y la unidad CCP se ha ajustado en modo *captura* permanentemente. El modo de funcionamiento se regula en el registro CCP1CON.

```
BANKSEL CCP1CON
bsf      CCP1CON, CCP1M2
bsf      CCP1CON, CCP1M0
```

Por último se inicializan algunas de las variables usadas en el programa como el número de sensores activos en el sistema, el *buffer* usado para ir acumulando comandos de control recibidos desde el PC, los punteros para gestionar dicho *buffer*, etc.

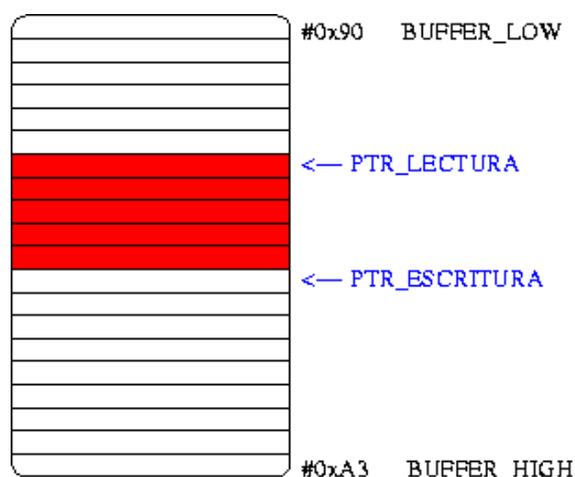


Figura 5.3: Ejemplo de uso del *buffer* de comandos recibidos

La figura 5.3 muestra un ejemplo para comprender cómo se realiza la gestión del *buffer* de comandos y las variables y constantes que intervienen. Las constantes (en letra negra) BUFFER_LOW y BUFFER_HIGH delimitan el tamaño y posición del *buffer* en memoria. Se utilizará el banco 3 que dispone de gran espacio libre para uso general. Se ha reservado un *buffer* de longitud 20 que creemos razonablemente grande para nuestra aplicación. En color rojo se ilustra la zona del *buffer* que contendría comandos y en blanco la zona libre. Las variables (en letra azul) PTRLECT (puntero de lectura) y PTRESC (puntero de escritura) indican respectivamente la posición donde se encuentra el inicio del siguiente comando a tratar y la siguiente posición libre del *buffer* donde se escribirá el siguiente comando recibido. Así, cada vez que se consuma un carácter del comando se deberá avanzar PTRLECT y cada vez que se copie una letra de un nuevo comando recibido se deberá avanzar PTRESC. El *buffer* es circular y los comandos no se borran directamente cuando se consumen, sino que sólo se trabaja sobre los punteros mencionados, consiguiendo una gestión bastante eficiente del *buffer*. Además se dispone de una variable que indica la ocupación del *buffer* y de otra que indica el número de comandos completos recibidos. En la fase de inicialización, el valor de las variables PTRLECT y PTRESC es el que contiene la constante BUFFER_LOW.

Para la gestión de los sensores que están activados/desactivados se utiliza una única variable llamada `SENSOR_MASK`. A cada sensor de la placa le corresponde un bit de la variable. El número de bits de la variable coincide con el número máximo de sensores que admite la placa (8 en nuestro caso). Si un bit de la variable está puesto a 1 significa que el sensor correspondiente está activo y deberá ser disparado. Si de lo contrario el bit está a 0 ese sensor no será disparado. Esto permite, como luego veremos, una activación y desactivación selectiva de los sensores que realmente están midiendo en cada momento.

5.1.2. Ciclo de medidas del ultrasonido

El fragmento de código encargado de los ciclos de medida de los sónares se compone de una serie de comprobaciones, el envío de la señal de `INIT` al sensor y a la vez la activación del contador, esperar el `ECHO` del sensor para detener el contador y por último enviar el tiempo de vuelo medido al PC a través del puerto serie. El flujo de control de este ciclo de medidas se resume en la figura 5.4.

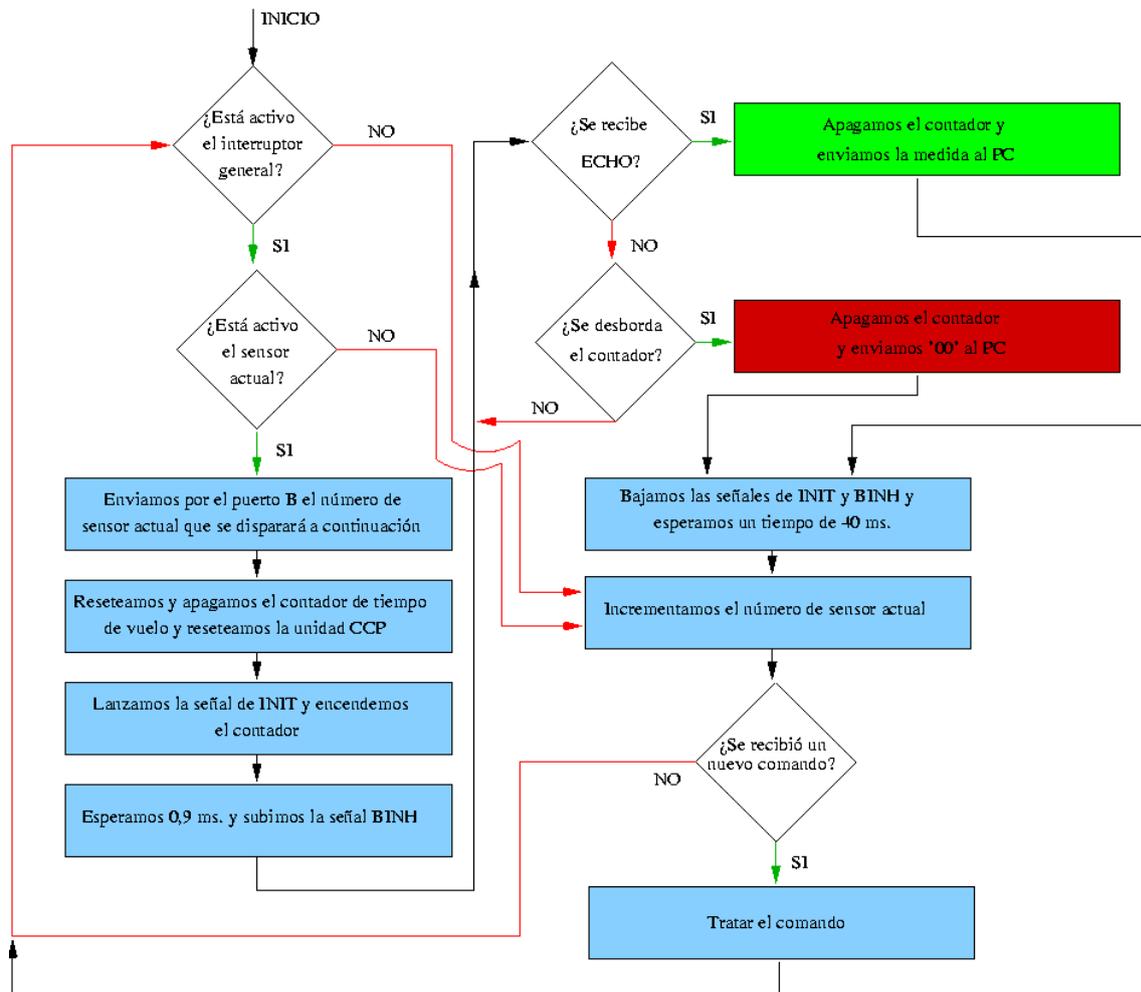


Figura 5.4: Diagrama de flujo del ciclo de medidas s3nar

Al comienzo del ciclo de medidas se comprueba si está activo el interruptor general del sistema. Como se puede observar en el diagrama de la figura 5.4, si no está activo el sistema, el PIC no realiza ningún ciclo de medida aunque existan sensores individualmente activos. Si está activo el sistema, el PIC se asegura de que haya algún sensor concreto habilitado y lanza el apropiado en cada instante. En caso de que en una iteración no se realice el ciclo de medida (bien por estar desactivado el interruptor principal o bien porque el sensor actual está inhabilitado), se incrementa el número de sensor actual y se comprueba si hay que tratar algún comando pendiente.

Si el interruptor global está habilitado y el sensor actual está activo, el PIC envía la dirección del sensor (del 1 al 8) al decodificador y éste envía la señal de INIT al sensor concreto. Una vez disparado el sónar, se realiza una espera activa bloqueante hasta recibir el ECHO o hasta que se produzca un desbordamiento en el contador.

Una vez recibido el ECHO, se detiene el contador y la unidad CCP congela los 16 bits de su valor en dos registros de 8 bits. A continuación se almacena la información de estos dos registros en dos variables del programa que contienen la medida de distancia expresada en un número proporcional de *tics* de reloj. El registro utilizado por la USART para el envío RS232 es de 8 bits, por lo que hay que enviar la información en dos veces. Se compone un mensaje como ilustra la figura 5.5, formado por un identificador de mensaje ('D') que indicará al PC que a continuación llegarán datos sensoriales. Después se incluye el número de sensor del que provienen los datos y finalmente el valor de los dos registros de 8 bits (la parte alta y baja del contador).



Figura 5.5: Formato del mensaje enviado al PC en caso de recepción de ECHO

En caso de que se desborde el contador por no haber recibido el ECHO, se enviará al PC el valor 0 por cada registro (figura 5.6).

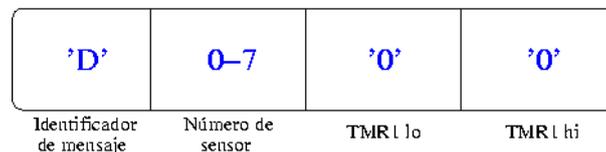


Figura 5.6: Formato del mensaje enviado al PC en caso de desbordamiento del contador

Finalmente se detiene el contador, se desactivan las señales de INIT y BINH y se prepara la siguiente iteración incrementando el número de sensor. Además hay que tener en cuenta que los sensores de ultrasonido son elementos *hardware* y necesitan un tiempo para estabilizarse después de cada ciclo de medida. Por tanto, tras cada medida se espera un tiempo de 40 mseg. para asegurar la estabilidad del sensor en la siguiente medida.

Los retardos se han realizado mediante subrutinas *software*. Se ha calculado el número de instrucciones que se ejecutan durante el tiempo que se pretende estar esperando y se ha construido un bucle que ejecute exactamente ese determinado número. A continuación se muestra el fragmento de código que causa el retardo de 40 mseg.

```
DEL_40
    movlw    0x34
    movwf    COUNT2
    clrf     COUNT1
del_low_40
    incfsz   COUNT1, F
    goto     del_low_40
    decfsz   COUNT2, F
    goto     del_low_40
    nop
    return
```

La rutina de envío por el puerto serie consta de tres pequeñas partes:

1. Espera hasta que se terminen de enviar datos anteriores (si los hubiera).
2. Envío de los datos. Como nuestro microprocesador dispone de una unidad USART esta tarea es muy sencilla. Únicamente es necesario copiar el registro a enviar a un registro especial TXREG de la USART. Todo el protocolo RS232 es realizado por esta unidad sin ninguna intervención nuestra.
3. Se espera en la subrutina hasta que el envío se completa.

A continuación mostramos la rutina mencionada anteriormente:

```
SendRS232
    nop
WaitTX
    BANKSEL PIR1
    btfss   PIR1, TXIF
    goto    WaitTX
    BANKSEL TXREG      ; Se cambia al banco 0
    movwf   TXREG      ; Se copia en TXREG el byte a enviar
    BANKSEL TXSTA      ; Se cambia al banco 1
wait_TX
    btfss   TXSTA, TRMT ; Se espera hasta completar el envío
    goto    wait_TX
    return
```

5.1.3. Rutina de tratamiento de interrupción

La rutina de interrupción es usada en el programa para desacoplar la parte de toma de medidas de los sónar de la parte de recepción de comandos de gestión, ambas en

el código del PIC. Así, los comandos recibidos son tratados de manera asíncrona. El uso de interrupciones permite realizar varias tareas y no perder tiempo en muestrear el puerto serie en busca de nuevos comandos. Una constante espera activa resultaría poco eficiente pues típicamente los comandos de configuración llegan al principio de la ejecución y después se envían en mucha menor medida.

En nuestro código cuando el PC envía datos al microcontrolador y éste los recibe, se genera la interrupción. El flujo de ejecución salta a la rutina de tratamiento que realiza las siguientes tareas:

- En primer lugar, se salvaguarda el estado en el que se encontraba la ejecución del programa antes de producirse la interrupción. Se guarda el registro trabajo y el de estado (W y STATUS), el contador de programa (PCLATH) y el registro para direccionar memoria (FSR).
- En segundo lugar, se comprueba si realmente saltó la interrupción esperada y si es así se llama a la subrutina de recepción RS232 que se encarga de extraer el comando enviado por el PC. Después se llama a la subrutina *escribir comando* para almacenar en el *buffer* el comando recibido para que posteriormente sea tratado. La razón por la que no se trata en la propia rutina de tratamiento de interrupción el comando, es tratar de minimizar el tiempo que se permanece en la propia rutina, ya que mientras se está dentro no se pueden recibir otros comandos nuevos.
- Por último, antes de salir de la rutina se restaura el estado en el que se encontraba ejecutando el programa y se retorna el flujo de ejecución.

Los comandos de gestión del subsistema sónar están formados por varias letras. Cada letra recibida genera una interrupción. La frecuencia de reloj del microcontrolador (4 MHz) es mucho más rápida que la velocidad de comunicación seleccionada para el puerto serie (9600 baudios). Esta diferencia asegura que durante el almacenamiento de una letra no llegará la siguiente en un tiempo razonable y habrá dado tiempo de sobra a la rutina de tratamiento de interrupción a terminar su labor.

5.1.4. Interfaz de comandos

La interfaz de comandos permite la interacción entre el PIC que gestiona el sistema sónar y el PC, ya sea a través de un programa de control o de la interacción de un usuario. El medio usado para la comunicación entre PIC y el PC es el puerto serie. La posibilidad de poder interactuar con la configuración de la placa redundante en una mayor flexibilidad de este subsistema.

El interfaz de comandos sirve para el envío de mensajes de configuración desde el PC al PIC y para la transmisión de datos del PIC al PC. La funcionalidad que ofrece esta interfaz es muy sencilla, los comandos que envía el PC al PIC permiten:

1. Activar/desactivar el sistema de sensores por completo (ON/OFF general).
2. Activar/desactivar un determinado sensor del sistema, indicando el número de sensor en el mensaje.

3. Pedir información de la configuración al PIC (frecuencia del reloj y número de sensores en la placa) para saber cuáles pueden activarse/desactivarse y para poder traducir a distancia la medidas temporales enviadas desde el PIC. El microcontrolador únicamente mide los *tics* de reloj que tarda la onda en rebotar. Estos *tics* serán mayores o menores en función de la velocidad del oscilador empleado. El PC deberá calcular el tiempo real de vuelo conociendo la frecuencia del reloj empleado por el microcontrolador y el valor enviado en el mensaje. Una vez calculado dicho tiempo deberá inferir la distancia al objeto.

Los datos que viajan en sentido PIC -> PC pueden ser de dos tipos: De configuración y de datos. Los datos de configuración son la respuesta al comando de *configuración* por parte del PC y contienen la respuesta a los datos solicitados en la petición. El otro tipo de mensaje (paquete de datos) contiene la información con los datos sensoriales de algún sensor.

5.1.5. Especificación del protocolo

Una vez especificados los comandos que intervienen en el protocolo descritos en el capítulo 4, hay que decidir una forma de implementarlos. Una primera aproximación fue que cada comando se identificara con una sola letra. De este modo si se quería activar el sensor 1 el comando sería la *A*. Si por contra se quisiera activar el sensor 2 el comando sería una *B*, y así sucesivamente con todos los comandos. El problema de esta implementación es que no es escalable pues hay que reservar tantos comandos como permutaciones entre sensores y órdenes haya.

La implementación realizada aborda de una manera más flexible y lógica la tarea de implementar el protocolo de comandos. Las órdenes se identifican con una letra o nemotécnico independiente del sensor involucrado. El número de dicho sensor se acompañará a continuación del nemotécnico de la orden. Al final de todo comando se enviará el carácter ';' para delimitar claramente su final y simplificar la rutina de tratamiento de mensajes. A continuación se muestra en detalle el protocolo implementado:

<i>Sintaxis</i>	<i>Función</i>
N;	Comando de ON , activa el sistema de sensores para que envíen medidas al PC vía PIC, en el caso que haya algún sensor activo.
F;	Comando OFF , desactiva el envío de medidas de sensores del PIC al PC
A,id;	Comando para A ctivar un sensor de ultrasonido concreto, el <i>id</i> indica el número de sensor a activar.
D,id;	Comando para D esactivar un sensor de ultrasonidos concreto, el <i>id</i> indica el número de sensor a desactivar.
R;	Comando de R eset que desactiva todos los sensores de ultrasonidos.
C;	Comando de C onfiguración, sirve para que el PC solicite la configuración del sistema de sensores de ultrasonido (frecuencia de reloj y <i>slots</i> de la placa con sensores conectados).

Tabla 5.1: Comandos de configuración

<i>Sintaxis</i>	<i>Función</i>
D,id,dato;	Envío de una medida de un sensor de ultrasonidos del PIC al PC. El campo <i>id</i> indica a qué sensor corresponde la medida enviada, mientras que el campo <i>dato</i> corresponde al valor de la medición.
C,reloj,mascara;	Comando que sirve para que el PC reciba del PIC la frecuencia de funcionamiento de su reloj interno y qué <i>slots</i> de la placa tienen un sensor conectado.

Tabla 5.2: Comandos de datos

5.1.6. Implementación usando un *buffer* circular

Para implementar esta interfaz de comandos para la comunicación entre el PIC y el PC hay que tener en cuenta que hasta que no se recibe un comando completamente (carácter ;), éste no es tratado. El tratamiento se realiza al terminar el ciclo actual de medida y sólo si existe algún comando en espera. La solución escogida para guardar las órdenes es usar un *buffer* circular manejado con dos punteros o índices. Un puntero marcará la posición de inicio del primer comando a tratar (PTRLECT) y el otro indicará la posición donde se almacenará la siguiente letra (perteneciente a un comando) recibida. Esta implementación no es trivial cuando se realiza en un microcontrolador y para realizarla hay que tener en cuenta las siguientes consideraciones:

- El *buffer* circular en sí no es más que un conjunto de posiciones de memoria consecutivas. Como se comentó en la fase de inicialización, las constantes BUFFER_LOW y BUFFER_HIGH establecen los límites inferior y superior del *buffer*, estableciendo indirectamente su tamaño.
- En una fase previa se inicializa el *buffer* circular. Concretamente esta inicialización consiste en poner los índices apuntando al comienzo del *buffer*, indicar en una variable que quedan 0 comandos por tratar y poner a 0 la variable que guarda el número de posiciones de memoria ocupadas.
- Para manejar el *buffer* es necesario usar direccionamiento indirecto a memoria a través del registro FSR. En éste se indica la dirección de la posición a leer/escribir y en el registro INDF se escribe el dato a guardar en caso de escritura o será el lugar donde se recibirá el dato en caso de lectura.

El ejemplo de la figura 5.7 muestra el efecto que tendría la instrucción:

```
movfw INDF
```

El valor de la posición de memoria #0x191 se copiaría en el registro W a través de INDF.

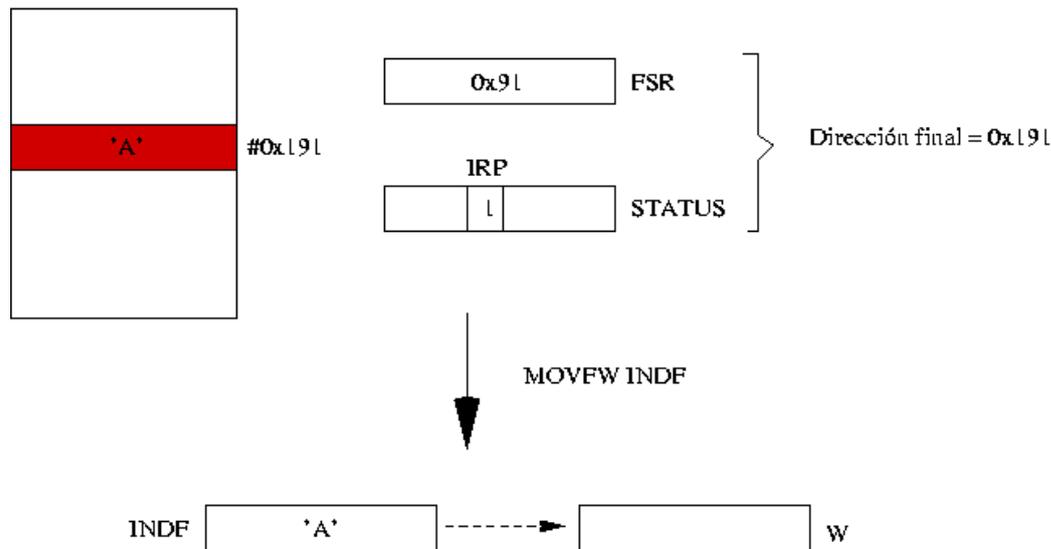


Figura 5.7: Ejemplo de lectura usando direccionamiento indirecto

En el siguiente fragmento de código real se muestra la lectura de una posición de memoria de las que componen el *buffer*. Primero se guarda en el registro de direccionamiento indirecto FSR la dirección de memoria a leer que está contenida en PTRLEC. A continuación se indica en el registro IRP que se va a usar direccionamiento indirecto sobre la segunda mitad de los bancos de memoria (2 y 3). Por último se mueve el contenido de INDF (el dato que estaba en la dirección contenida en PTRLEC) al registro W y de aquí se guarda en la variable de programa READ.

```

BANKSEL PTRLEC
movfw  PTRLEC
BANKSEL FSR
movwf  FSR
BANKSEL STATUS
bsf    STATUS,IRP
BANKSEL INDF
movfw  INDF
BANKSEL READ
movwf  READ

```

- Otra consideración a tener en cuenta es que una vez que se ha llenado el *buffer* no se aceptan más comandos y éstos se descartan. Cada vez que se inserta en el *buffer* un nuevo comando se comprueba si hay suficiente espacio libre para almacenarlo, chequeando la variable que mantiene el estado de ocupación del *buffer* (*BUFFER_LOAD*).

Una vez definido el *buffer* circular y el protocolo de comandos, resta describir cómo se ha realizado su manipulación. El manejo se realiza a través de subrutinas encargadas de funciones de manejo concretas: Incrementar los índices del *buffer*, escritura de un nuevo comando, lectura letra a letra de un comando y eliminación de órdenes incorrectas.

- La subrutina que incrementa los índices se lanza cuando se ha leído o escrito en el *buffer*. Si la operación realizada ha sido de lectura se incrementa la variable PTRLEC. Por contra, si la operación es de escritura se incrementa PTRESC (el índice de escritura en el *buffer*).
- La subrutina para la escritura de comandos, inicialmente comprueba si el *buffer* está lleno y si lo está, lo borra completamente. Si aún hay huecos libres comprueba primero si es correcto el comando recibido, es decir, mira la letra inicial y los parámetros. Si la información es correcta escribe en el *buffer* los caracteres que componen el comando, incrementando cada vez la variable que guarda el número de posiciones ocupadas y el índice que apunta a posiciones libres en el *buffer*. También incrementa la variable que almacena el número de comandos pendientes por atender, cada vez que se escribe un carácter de terminación de comando ';'.
- La subrutina encargada de leer del *buffer* solo lee un carácter cada vez. De manera se deben hacer sucesivas llamadas para leer un comando completo. Con esto se persigue construir rutinas muy específicas y lo mayor desacopladas posibles.
- Para eliminar comandos incorrectos del *buffer* existe una subrutina que se encarga de este trabajo. A partir del índice de lectura, en el momento en que se llama a la subrutina, hasta el carácter de fin de comando incluido se eliminan todos los caracteres.

Por último, existe una subrutina que es la encargada de tratar los comandos recibidos del PC, una vez leídos del *buffer* circular. Esta subrutina averigua en primer lugar qué tipo de comando es (ON, OFF, activar un sensor, etc.) y en función de éste ejecuta el código correspondiente. De este modo las distintas posibilidades en función del comando son:

- Si el comando es de ON se pone la variable POWER a 1, indicando que el sistema está activo, como vimos en el capítulo 4. Esta variable es la que se comprueba periódicamente al comienzo de cada iteración de medidas sónar.
- Si el comando es de OFF se pone la variable POWER a 0, para indicar que no hay que hacer más ciclos de medida.
- Si el comando es A se lee el siguiente carácter del comando, que es un número de sensor. Se pone a 1 la posición correspondiente al número leído en la máscara de sensores activos (SENSOR_MASK), para indicar que éste está activo y se deben tomar medidas con él.
- Si el comando es D se lee el siguiente carácter, que es un número de sensor. Se pone a 0 la posición correspondiente en la máscara de sensores activos, para indicar que éste está desactivado y no se deben tomar medidas con él.
- Si la letra inicial del comando es R se borra la máscara de sensores activos, indicando así que no hay sensores disponibles para tomar medidas.
- Si la letra es C quiere decir que el PC quiere recibir información de configuración sobre la frecuencia de reloj del PIC y los sensores conectados a la placa. De este modo se envían 32 bits (en 4 llamadas) con el valor de la frecuencia de reloj y 8 bits con la máscara que indica los *slots* de la placa con sensores conectados.

5.2. Programa Monitor

En la anterior sección se ha detallado la programación del *hardware* necesario para adquirir los datos provenientes de los sensores de ultrasonido. Asimismo, también se dispone de una biblioteca *software* para recoger las medidas del sensor láser, ocultando los detalles de bajo nivel y ofreciendo los datos sensoriales por el puerto serie. Se ha programado una aplicación gráfica llamada Monitor que permite visualizar las medidas sensoriales de una manera clara e intuitiva y además, acepta parámetros para modificar el comportamiento de los sensores. Sus objetivos principales son la depuración y supervisión del sistema perceptivo. La aplicación desarrollada tiene el aspecto que muestra la figura 5.8. A continuación, describiremos la estructura *software* del programa y detallaremos cómo hemos realizado dicha aplicación.

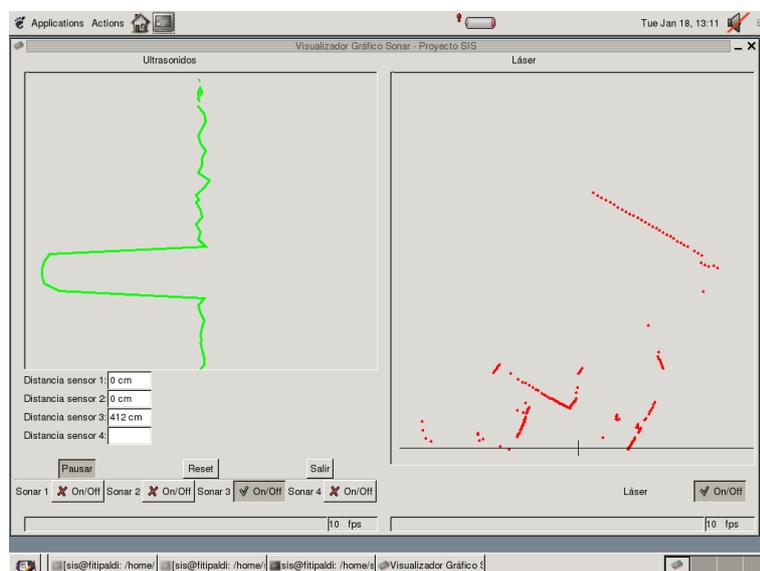


Figura 5.8: Aspecto general de la aplicación Monitor

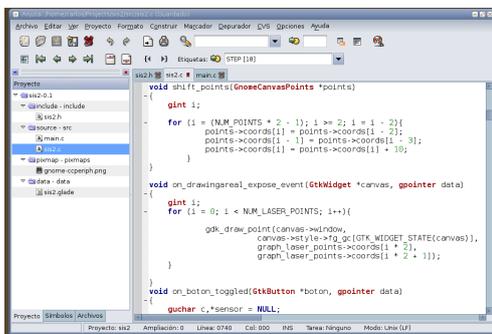
Antes de pasar a comentar los detalles relevantes de la aplicación hay que destacar que para la comunicación con la placa s3nar se ha utilizado una biblioteca auxiliar. Esta biblioteca proporciona primitivas para configurar la transmisi3n RS232 a la velocidad adecuada y funciones para enviar y recibir datos. Del mismo modo, se ha utilizado otra biblioteca para la comunicaci3n con el l3ser que permite configurar el dispositivo y solicitar la recepci3n de medidas. Estas bibliotecas se denominan *serial* y *laserdriver*.

5.2.1. Toolkit gr3fico: GTK+

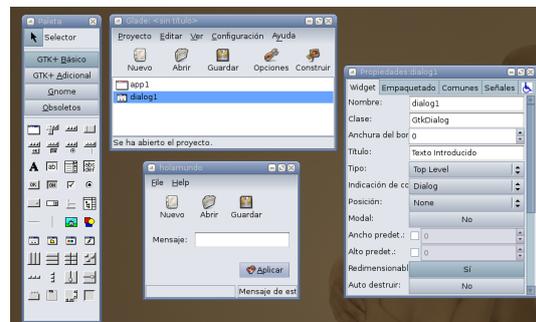
La primera decisi3n que hay que tomar antes de programar cualquier aplicaci3n gr3fica es seleccionar el *toolkit* gr3fico que se va a emplear. Un *toolkit* es una biblioteca que ofrece una gran cantidad de elementos gr3ficos (*widgets*), un API de programaci3n para utilizarlos y estructuras de datos necesarias para manejar aplicaciones visuales. Habitualmente se incorporan tambi3n herramientas para modelar el aspecto final del programa de manera c3moda y gr3fica. Para el sistema operativo que nos ocupa (GNU/Linux) hay varios *toolkits* disponibles entre los que podr3amos destacar Qt [Qt, 2004], Gtk+ [Moya, 2002], Xforms [T. C. Zhao, 1997], Tkinter [Lundh, 1999], etc.

Nuestra aplicación gráfica no va a formar parte del producto final, es decir, es únicamente un elemento de utilidad para depuración y de cara a los clientes, que podrán visualizar los datos sensoriales. El producto final es un sistema instalado en un camión, cuyo funcionamiento es totalmente transparente al conductor, sin pantallas, sin mensajes, etc. Sin embargo, en nuestro prototipo es muy importante disponer de toda la información sensorial a la vista para corregir errores y ajustar su funcionamiento, antes de escribir el programa que toma las decisiones. En las máquinas que hemos utilizado para desarrollar este proyecto utilizamos Gnome [Foundation, 2004] como escritorio, por lo que decidimos emplear Gtk+ como *toolkit* gráfico para nuestro programa de visualización, logrando así la máxima integración con el resto de aplicaciones.

El proyecto Gnome está formado por una comunidad muy activa de desarrolladores y Gtk+ está en constante desarrollo y evolución, disponiendo de gran cantidad de documentación. Éste fue otro de los motivos por lo que la elegimos, pues debíamos aprender desde cero una tecnología nueva para nosotros y la documentación es un punto de partida muy importante. Además, destacan dos herramientas asociadas que nos motivaron más aún a escoger Gtk+: Glade [Giles, 2003] y Anjuta [Naba Kumar, 2002] (figuras 5.9(b) y 5.9(a)). La primera de ellas permite diseñar visualmente la interfaz gráfica del programa. Glade genera un fichero *.xml* con toda la información del GUI pero sin ninguna funcionalidad, que será posteriormente cargado desde el código fuente. De esta manera conseguimos independizar al máximo el código y el GUI de Monitor, pudiendo incluso cambiar la interfaz de usuario sin necesidad de recompilar el código fuente. Anjuta es un entorno integrado de desarrollo que facilita mucho la labor de programación. Está diseñado para escribir código en C y C++ sobre GNU/Linux y construir aplicaciones GTK/GNOME.



(a)



(b)

Figura 5.9: Pantallazos de los entornos Anjuta a) y Glade b)

Gtk es un toolkit orientado a eventos. Al llamar a la función `gtk_main()`, el programa entra en un bucle infinito y permanece a la espera de eventos. Los eventos pueden ser de muchos tipos: La pulsación de un botón por el usuario, la entrada de texto, la destrucción de un *widget*, etc. Cada *widget* tiene sus propios eventos o señales (*signals*). Prácticamente todos los *widgets* generan alguna señal.

Para que el programa haga algo, será necesario asociar los eventos a *callbacks* (funciones que realizarán algo en función del evento que se haya producido). Todos los *widgets* tienen el método *signal_connect* que nos permitirá asociar un evento a un *callback*.

5.2.2. GUI de Monitor

Como se ha mencionado anteriormente se ha utilizado Glade para el diseño de los *widgets* (botones, cuadros de diálogo, etc) que componen la aplicación. Se ha dividido la ventana principal en dos mitades correspondiendo cada una de ellas a un tipo de sensorización. Así, la parte izquierda está dedicada a los sensores de ultrasonido, mientras que la parte derecha se destina al sensor láser. La figura 5.8 muestra la apariencia de *Monitor*.

La zona de ultrasonidos usa una ventana para representar las medidas. Para que resulte más intuitivo, los datos se dibujan usando una línea o estela que dependiendo de la distancia al objeto estará más cerca o lejos del centro de referencia. El objetivo de esta estela es mostrar el perfil de los objetos que tenemos delante de los sensores. Además, esta línea desliza con el tiempo hacia la parte inferior de la ventana, para mantener una visión más duradera de las medidas anteriores (el eje *X* de la ventana muestra la distancia al objeto y el eje *Y* representa el paso del tiempo). Cada sensor se identifica con una línea de color distinta para independizar todas las medidas y obtener una visualización más clara. En la figura 5.10 se puede ver con detalle esta zona de la aplicación. Se observa que hay tres sensores conectados (rojo, azul y verde). Las estelas roja y azul prácticamente están solapadas, lo que indica que los sensores están físicamente juntos y dirigidos al mismo objeto. Además, aparecen muy desplazados a la izquierda de la ventana, por lo que podemos inferir que la distancia al objeto es pequeña. La información que nos remite la estela verde es que los obstáculos están a mayor distancia y que los objetos son móviles puesto que la estela presenta picos y oscilaciones.

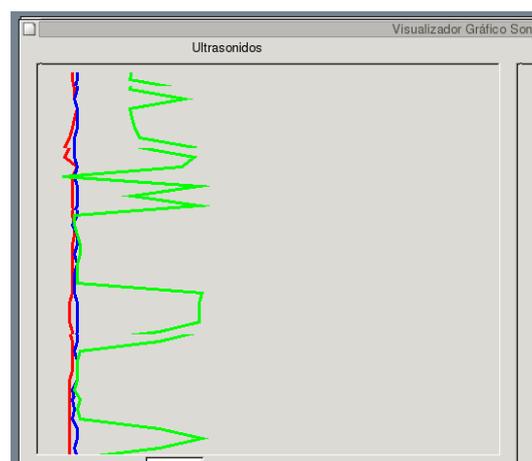


Figura 5.10: Detalle de la zona de visualización de ultrasonidos

Debajo de la ventana de visualización de las estelas, hemos situado unos cuadros de diálogo que muestran las medidas numéricas instantáneas de cada uno de los sensores activos en ese momento. En la parte inferior se encuentran los elementos de control, que permiten pausar todo el sistema de ultrasonidos, activar o detener sensores concretos y dejar todo el sistema en un estado inicial conocido (Reset). Estos botones están acoplados con los mensajes correspondientes del protocolo entre el PC y la placa sónar, según vimos en la tabla 5.1. En la esquina inferior derecha hay una pequeña ventana que muestra la frecuencia de medidas recibidas por segundo, a través de la cuál se comprueba la vivacidad de las capturas. El código que hay por debajo de esta aplicación se apoya en el diálogo a través de puerto serie y sigue el protocolo descrito en la tabla 5.1.

En el lado derecho de la aplicación (figura 5.11) hay una gran ventana donde se representan en dos dimensiones los datos recibidos por el láser. Esta representación es un visión cenital o de vista de pájaro de los datos en el plano del láser (que permanece constante y está definido por la altura a la que se sitúa el mismo). La función principal del láser es identificar los perfiles de los obstáculos y gracias al *Monitor* se comprobará el grado de acierto o de error en los mismos. También dispone de un botón para detener las medidas y borrar la visualización y un pequeño panel informativo con el número de barridos realizados por segundo.



Figura 5.11: Detalle de la zona de visualización láser

Por último se ha incluido un botón para salir de la aplicación correctamente, aunque también es posible hacerlo usando el icono en forma de aspa del marco superior de la ventana.

5.2.3. Estructura de *Monitor*

Ahora llega el momento de codificar el propio programa para que lea los datos provenientes de los sensores y modifique los *widgets* apropiados para que los usuarios puedan ver toda la información. A través del entorno integrado de desarrollo Anjuta creamos toda la estructura de directorios de la aplicación para conseguir mayor orden en el código (figura 5.12).

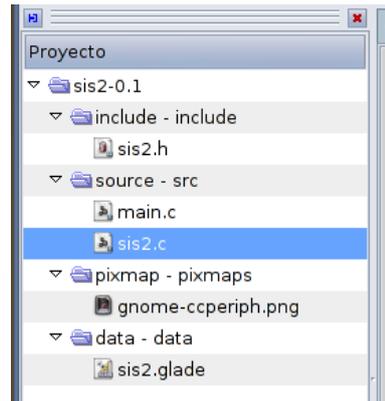


Figura 5.12: Árbol de directorios creados para Monitor

El árbol de directorios queda de la siguiente manera:

- **include:** Aquí se encuentran todos los ficheros de cabecera utilizados (*.h*).
- **src:** En este directorio encontramos todos los ficheros de código fuente (*.c*).
- **pixmaps:** Directorio que contiene todas las imágenes usadas en la aplicación (iconos, dibujos de botones, etc.).
- **data:** Este es el sitio para los ficheros *.XML* correspondientes al interfaz.

Una vez estructurado el código fuente y con ayuda de la herramienta *Makefile* se elaboraron los ficheros con las interdependencias del código fuente. A través de esta herramienta la fase de compilación se limita a introducir el comando *make*, ya que toda la información de dependencias y búsqueda de librerías está contenida en ficheros de configuración de la herramienta.

El fichero *sia.c* contiene el código funcional de la aplicación Monitor. Como muestra la figura 5.13 *Monitor* consta de varias partes que se describirán a continuación. Al comienzo del programa la primera tarea a realizar es una fase de inicializaciones. Tanto el puerto serie de los sensores de ultrasonido como el sensor láser se inicializan y ajustan a las velocidades seleccionadas por defecto (9600 baudios para los sonda y 38400 baudios para el láser). Seguidamente se activan todas las señales asociadas a los eventos que contiene la aplicación (pulsación de botones, finalización de programa, etc.). El siguiente paso es obtener todas las referencias de los *widgets* que se necesitan en el código a partir del fichero *.XML* que contiene el interfaz gráfico. Por supuesto, todas estas tareas las llevamos a cabo utilizando el API de programación ofrecido por GTK+.

Después de otra fase de inicialización de estructuras de datos para las estelas, comienza una parte importante del código que es la creación de **canales**. Un canal es el medio utilizado en GTK+ para el tratamiento de ficheros y canales de entrada/salida. Existe un tipo de datos llamado *GIOChannel*, que permite encapsular ficheros, *sockets* y *pipes*. Esta abstracción no sólo posibilita el acceso uniforme a todos estos recursos, sino que también asegura portabilidad e integración al bucle de eventos. De esta manera, podremos especificar que se produzca un evento cuando lleguen datos por

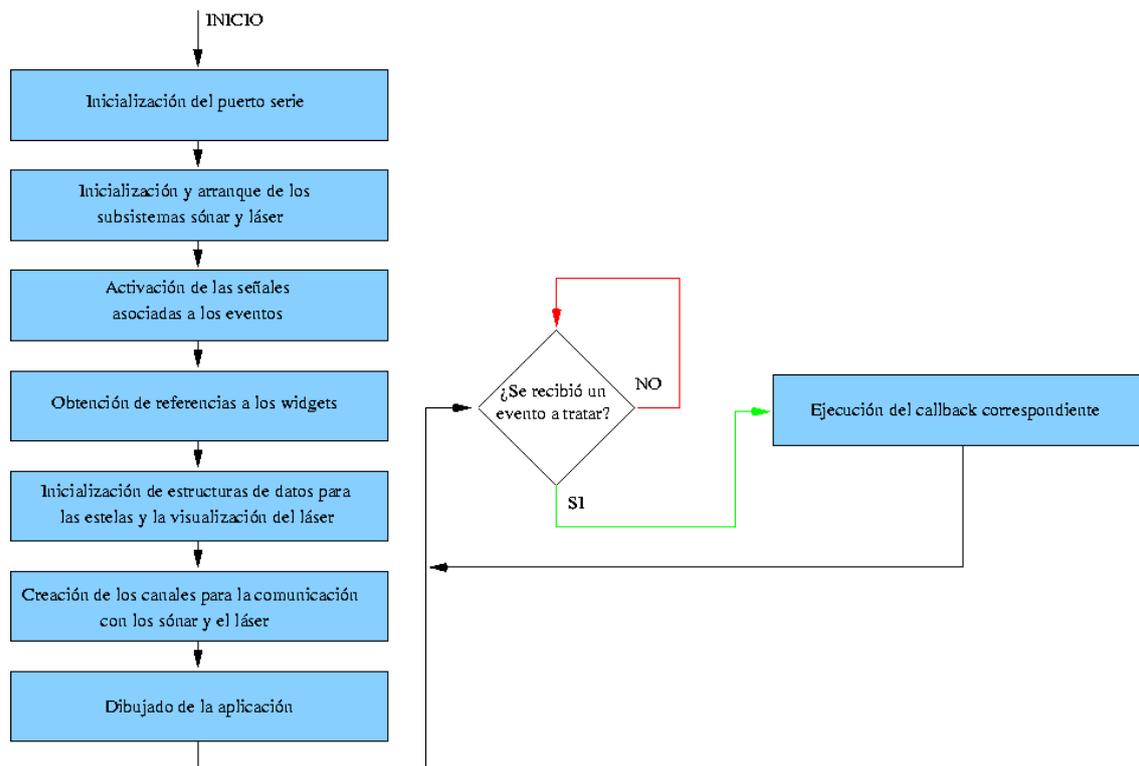


Figura 5.13: Diagrama de flujo de Monitor

un determinado canal creado. En concreto, se crean dos canales asociados a los dos puertos serie conectados (el de conexión con la placa de ultrasonidos y el de conexión con el láser) y se añade un evento a cada canal cuando se reciban datos por cualquiera de ellos.

Finalmente se terminan de inicializar las estructuras de datos para los contadores de medidas por segundo y se ordena el dibujado de la aplicación. Justo a continuación el programa entra en un bucle infinito en el que sólo se atienden eventos. En este punto el control de la aplicación pasa a GTK+. Todo el resto de código desarrollado se destina a atender determinados eventos, que serán tratados por el *callback* o rutina de tratamiento del evento definida por nosotros para cada caso. Los eventos necesarios que hemos identificado para la aplicación Monitor son cinco. Para todos ellos hemos programado y conectado las rutinas pertinentes de servicio.

1.- Evento: Recepción de datos por el canal de ultrasonidos

Lo primero que se realiza en esta función es extraer los datos del canal. El dato recibido corresponde a una medida de tiempo interna del microcontrolador que controla los sónar y no es directamente una medida de distancia. Por tanto, se calcula la distancia real al objeto correspondiente a partir de ese valor y disponiendo de la frecuencia de trabajo del microcontrolador. Después, se modifica la estructura de datos de la estela adecuada con la nueva medida y se redibuja. Por último se refresca el cuadro de diálogo con la medida exacta recibida.

2.- Evento: Recepción de datos por el canal del láser

En el momento que se recibe la señal de que hay datos procedentes del láser, éstos son recogidos por medio de una función. A continuación, se actualiza la estructura de datos con la representación 2D del láser para que el siguiente repintado refresque la ventana con la nueva información.

3.- Evento periódico de refresco del identificador de medidas recibidas (*fps*)

Cada segundo hay definido un evento para que emita una señal y el *callback* correspondiente refresque los contadores de medidas recibidas por segundo en cada ventana. Por supuesto, cada vez que se reciben datos, las funciones incrementan las variables correspondientes.

4.- Evento para el envío de comandos de control hacia la placa sónar

En nuestro caso, la comunicación por los canales es bidireccional y también podemos modificar el comportamiento de alguno de los sensores, detenerlos, etc. Cuando se recibe la pulsación de un determinado botón, se asocia el comando de control a enviar, según el identificador del botón que ha generado el evento. Después se envía la orden a la placa de ultrasonidos a través de su canal y respetando el protocolo interno.

5.- Evento para salir de la aplicación

Cuando se pulsa el botón de salida de la aplicación o bien el aspa de la ventana, se produce el evento que desencadena la finalización del programa. Antes de salir se elimina la memoria dinámica reservada para las estelas, se elimina del bucle de eventos periódicos la función que refresca los *fps* recibidos y se detienen los subsistemas sensoriales. Después se llama a la función *gtk_main_quit()* que abandona el bucle principal y cierra la ventana del *Monitor* liberando todos los recursos.

5.3. Aplicación de control

Una vez construido y programado un subsistema perceptivo sólido, tenemos que programar la aplicación que digiere todos los datos ofrecidos por los sensores y decide en consecuencia. Estos datos llegan con interferencias (en el caso de los sónares) y con más información de la necesaria (en el caso del láser y debido a la gran amplitud angular de la que se dispone). La aplicación de control analiza los datos y extrae la información relevante, decidiendo finalmente el estado de la aplicación. Existen tres posibles estados: Búsqueda de obstáculo frontal, detección de obstáculo lateral y detección de fin de obstáculo. La figura 5.14 ilustra el autómata de estados que utiliza la aplicación de control.

El primer estado se basa en el análisis de los datos tomados por el láser. Su objetivo es identificar un vehículo que potencialmente será adelantado. El segundo estado, se inicia cuando se ha detectado el automóvil y se ha iniciado la maniobra de adelantamiento. Ahora, los sónares están bajo aviso y se analizan sus medidas. Cuando se percibe que el vehículo está a la altura del sensor se produce un nuevo cambio de estado. La última transición nos traslada a la detección del fin del adelantamiento,



Figura 5.14: Diagrama de flujo del autómata empleado en la aplicación de control

analizando de nuevo las medidas proporcionadas por los ultrasonidos. Cuando el vehículo es rebasado, el autómata retorna al punto de partida inicial y comienza de nuevo el ciclo. Durante los estados dos y tres se activa el limitador que permite mayor velocidad, mientras que en el estado uno, el limitador permanece en su posición más restrictiva.

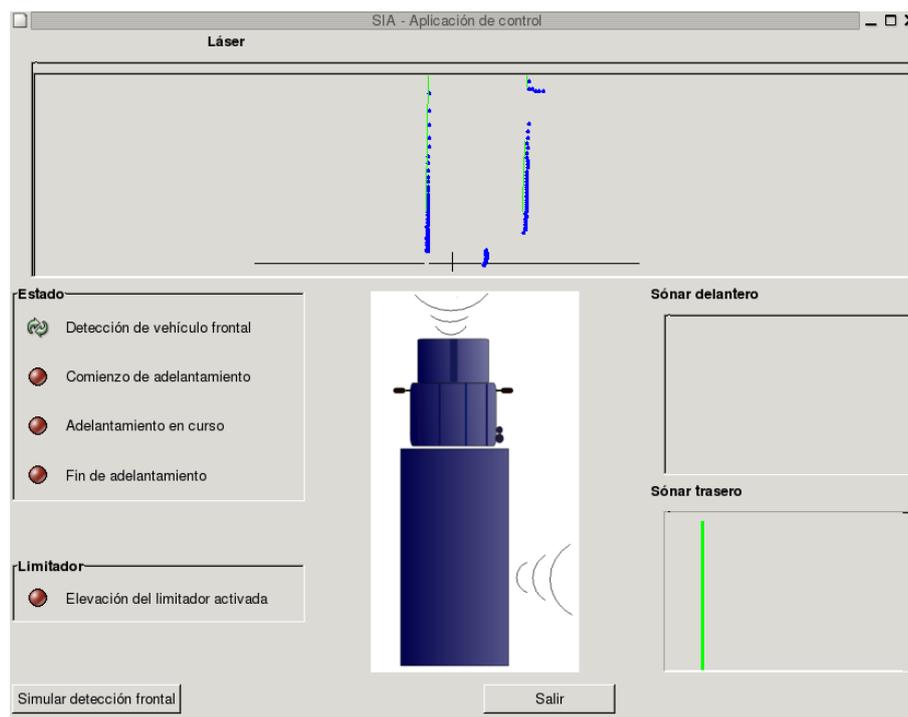


Figura 5.15: Aspecto de la aplicación de control desarrollada para el proyecto SIA

La figura 5.15 muestra el aspecto de la aplicación de control. En la parte izquierda se aprecian una serie de iconos que indican el estado en el que se encuentra la aplicación. En la figura aparecen cuatro estados en lugar de los tres mencionados anteriormente porque esa aplicación maneja dos sensores de ultrasonido y con esa configuración se obtiene un estado adicional. En la esquina inferior izquierda se sitúa otro icono que muestra si el limitador de velocidad está activo o no. En la parte derecha y superior se dispone de ventanas de visualización para tener una idea de las medidas obtenidas

si no se quiere lanzar la aplicación Monitor. Por último, existe un botón en la parte inferior izquierda de la ventana, utilizado para simular la detección frontal con el fin de depurar la aplicación.

Las transiciones de un estado a otro están dirigidas por rutinas *software* que estudian los datos convenientemente. A continuación exponemos cómo se ha realizado la detección frontal y lateral del vehículo (el fin de adelantamiento es prácticamente análogo a la detección lateral).

5.3.1. Detección frontal del vehículo

La información sensorial proporcionada por el sensor láser consiste en 180 medidas de distancia correspondientes a la colisión o no de los haces de luz emitidos con distintos objetos. En la figura 5.16 se pueden observar en rojo los puntos correspondientes a las medidas láser.

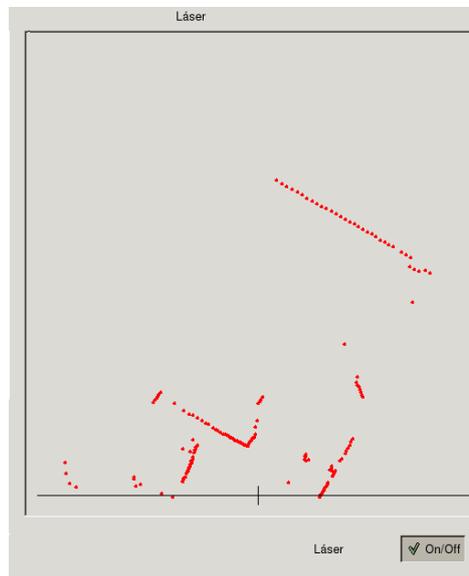


Figura 5.16: Medidas obtenidas por el sensor láser

Esta información hay que procesarla para la detección del vehículo precedente al realizar la maniobra de adelantamiento. Las 180 medidas que se obtienen son muy difíciles de interpretar directamente para concluir que hay o no un vehículo delante. Esta información hay que transformarla para que sea más sencillo extraer esta conclusión. Esta transformación se hace usando una función de segmentación que agrupa puntos en segmentos rectos si el láser se sitúa en frente del camión, centrado y mirando hacia delante. Con la información en forma de segmentos, un vehículo precedente se corresponderá a un segmento aproximadamente horizontal, de una determinada longitud y centrado en el foco del láser.

Para la implementación de una función que realiza esta segmentación hay que tener en cuenta que la solución deberá ser lo más robusta posible pero rápida a la vez, y que lleve poco tiempo de proceso para no ralentizar otras acciones. Teniendo en cuenta estas consideraciones se adaptó un algoritmo de segmentación implementado para el comportamiento autónomo sigue-pared para un robot EyeBot [Gómez, 2003].

Este algoritmo de segmentación se basa en el principio de hipotetizar sobre segmentos, se trata de un algoritmo que extrae los segmentos desde los puntos finales a los iniciales. Este algoritmo consiste en estimar la pendiente entre el primer (haz 0) y el último (haz 179) punto obtenidos por el láser, para posteriormente intentar verificar si todos los puntos están, con una determinada tolerancia, dentro de esta recta hipotetizada. De este modo se calcula la posición hipotética sobre el eje de coordenadas Y para cada punto x del interior, basándonos en la ecuación punto-pendiente de la recta mostrada a continuación. Con esta ecuación podemos calcular el valor de y ($y_{hipotetico}$) para un punto si conocemos el valor x para ese punto (x), otro punto de la recta (x_1, y_1) donde se encuentra el punto a analizar y el valor de la pendiente ($pendiente_recta$) para la recta.

$$y_{hipotetico} = pendiente_segmento * (x - x_1) + y_1$$

Una vez calculado este valor se compara con el valor real que tiene ese punto en el eje Y . Si se sale del umbral puede asumirse como punto inicial de un segmento de manera que se para ahí el algoritmo. El algoritmo empezará ahora a trabajar con un nuevo segmento hipotetizado que será el que tenía como punto inicial, al que su valor del eje Y quedaba fuera del umbral establecido, y como punto final el mismo que tenía el segmento descartado. Esta solución asume que un segmento es bueno, hace válida la hipótesis, cuando alrededor del segmento todos los puntos que pertenecen a éste al ser analizados están dentro del umbral para ese segmento. Además de que todos los puntos estén dentro del umbral tiene que cumplirse otra condición y es que la distancia entre puntos no sea mayor que un determinado valor. Si hay puntos entre los cuales la distancia es muy grande seguramente no pertenezcan al mismo segmento (dos coches en paralelo, etc.).

Cuando se valida una de estas hipótesis, se extrae un segmento, quiere decir que hemos encontrado algún segmento por la izquierda. Entonces el siguiente paso es pasar a analizar el resto de las medidas que queda sin explicar, así ahora el algoritmo trabajará con un segmento de punto inicial el que está más a la derecha (primeras medidas) y como punto final el punto anterior al comienzo del segmento validado. Estas operaciones se repetirán hasta que se hayan analizado los 180 puntos obtenidos por el láser.

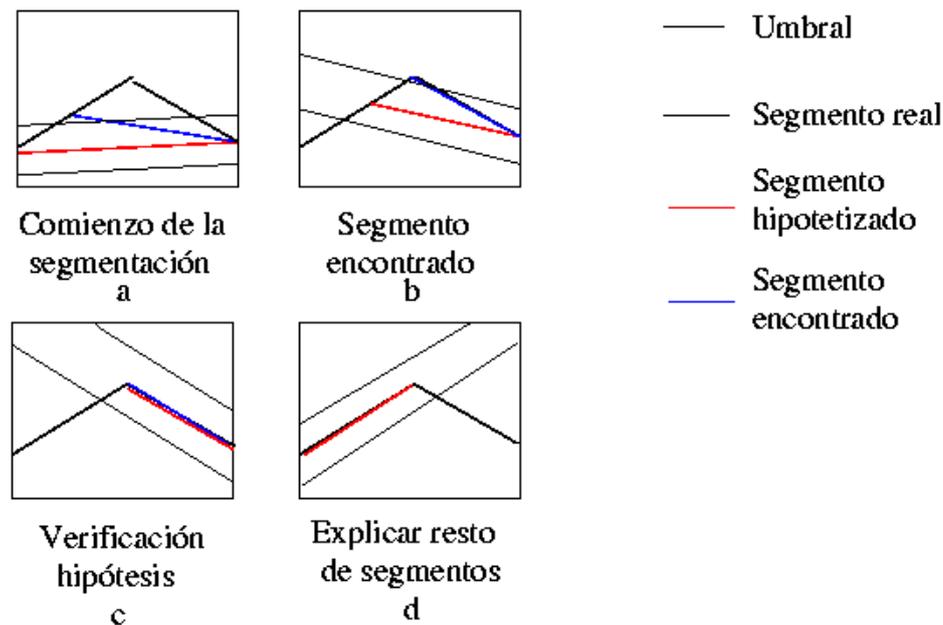


Figura 5.17: Proceso teórico de segmentación

En la figura 5.17 se muestra como el algoritmo va realizando la segmentación. En la imagen a) se puede ver como el primer segmento hipótesis (en color rojo) va del punto inicial al final de los puntos a segmentar. En esta figura a) también se puede observar como la hipótesis se rompe porque hay varios puntos de segmentos reales que están fuera del umbral de tolerancia. En ese momento el algoritmo crea un nuevo segmento hipotético (en color azul) que va desde el punto que rompió la hipótesis hasta el final. En la figura b) se observa como la última hipótesis (en rojo) tampoco era correcta y se realiza una nueva (en azul). En la imagen c) se valida el último segmento hipotetizado. Y por último, en la figura d) se empieza a explicar el resto de los puntos.

El proceso de segmentación de los puntos láser se puede ver en la imagen 5.18. En la figura a) se puede observar como inicialmente el segmento hipotetizado (en color rojo) es el que va de la medida del haz 0 al 179. En la figura a) también se puede observar que el primer punto analizado que está fuera del umbral (en color rojo en discontinua) es el que rompe la hipótesis (señalado por la flecha de color negro). Cuando se rompe la hipótesis se realiza otra nueva que en este caso va desde el punto donde se rompió hasta el punto final, como se puede observar en la imagen b). En esta imagen b) se puede observar también en qué punto se rompe la hipótesis (indicado por la flecha). En la figura c) el algoritmo ya ha descartado más hipótesis y descarta ésta también porque la distancia entre los puntos de la parte de arriba de la hipótesis es muy grande. En este caso se rompe la hipótesis en el punto (indicado por la flecha) en el que su distancia con el anterior era muy grande. Por último en la figura d) se puede ver como el algoritmo ha validado una hipótesis (el segmento de color verde) y se dispone a analizar el resto de las medidas.

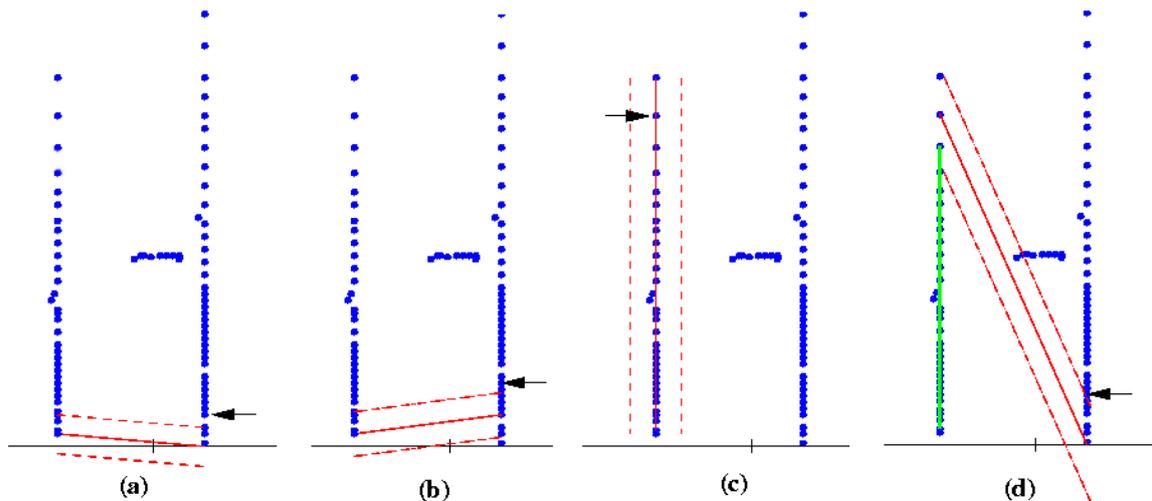


Figura 5.18: Proceso de segmentación de los puntos láser

Después de realizar esta segmentación se comprueba si se está en el caso en el que hay un vehículo precedente cercano. Esta comprobación se realiza analizando si hay algún segmento de los calculados que posea ciertas características. Estas características son:

1. El segmento debe estar centrado en el foco del láser, es decir, debe ser un segmento correspondiente a la medidas centrales de entre todas las obtenidas por el barrido del láser.
2. El segmento debe tener pendiente casi nula. De momento sólo estamos analizando el caso sencillo de adelantamiento. En este caso se supone que el adelantamiento se producirá en un tramo recto. Por este motivo el vehículo precedente se corresponde con una segmento casi horizontal.
3. La distancia entre el láser y el segmento debe ser menor a un cierto valor. Se debe respetar este umbral para que el vehículo precedente se pueda detectar desde unos 50 m. pero hasta que no estemos suficientemente cerca el adelantamiento no tiene sentido.
4. El segmento debe tener una cierta longitud para que se pueda concluir que es un vehículo.

5.3.2. Detección lateral del vehículo

Los sensores de ultrasonido situados en el flanco derecho del vehículo entran en juego cuando un adelantamiento está en curso. Deben ser capaces de detectar cuando el vehículo rebasado está a la altura del sensor y cuando deja de estarlo. Estos instantes se corresponden con los cambios de estado vistos en la figura 5.14. Los sensores de ultrasonido por sí solos no proporcionan suficiente información para detectar esta situación en todos los casos. El láser sí que es suficientemente preciso

y rico en información para poner sobre aviso a los sensores laterales cuando la maniobra comienza. En este momento, los sensores funcionan como una barrera virtual detectando cuando son interrumpidos (detección de vehículo) y cuando vuelven a dejar de estarlo (fin del adelantamiento).

Hemos seleccionado una ventana dinámica y configurable que analiza y calcula la media de los valores capturados durante un espacio de tiempo (por ejemplo 2 seg.). Cada medida capturada por los ultrasonidos es comparada con la media de valores recibidos en la última ventana temporal. Si la medida varía un cierto umbral, esta medida se considera como un posible obstáculo. Si la siguiente medida también queda por debajo del umbral, se pasa a una fase de verificación del obstáculo. La media de valores obtenida hasta este punto se almacena temporalmente. Si hay un obstáculo delante del sensor de ultrasonidos, la media de valores de la ventana dinámica se reducirá paulatinamente. En el momento que esta media difiera el umbral marcado respecto a la media anterior almacenada, se concluye que se ha detectado un obstáculo. Esta técnica es robusta ante ruido durante la fase de verificación del obstáculo.

Sin embargo, una medida ruidosa al comienzo del ciclo podría generar una falsa alarma y tomar la medición como posible obstáculo. Si era simplemente ruido, la siguiente medida, previsiblemente no superará el umbral y se iniciará de nuevo el ciclo, descartando la medida anterior al considerarse ruido.

La detección del fin de obstáculo es simétrica a la detección de obstáculo. En este caso, las medidas deberán superar un umbral por encima de la medida actual, que indicará que el sensor detecta mayor distancia al objeto, por lo que presumiblemente el automóvil habrá sido rebasado.

Pruebas Experimentales

Una de las etapas del ciclo de vida previa a la entrega de un proyecto informático para su explotación es la fase de pruebas. Nuestro sistema no se trata de un programa informático aislado, sino que consta de una parte *hardware* y varios programas *software* que deben ser probados. Esta fase de pruebas analiza conjuntamente *hardware* y *software* como un único sistema. Las pruebas se centran inicialmente en comprobar la validez del sistema sensorial (ultrasonidos y láser) tanto en interiores como en exteriores. Una vez validado éste, se analiza la primera versión completa del sistema, es decir, el subsistema sensorial y el subsistema de control en conjunto. Esta última fase no pretende comprobar la eficacia total de la aplicación, sino únicamente corroborar que todo funciona adecuadamente y que la aplicación responde correctamente ante una situación controlada y en un entorno de interior. No ha sido posible probar el sistema completo en un escenario real instalado en un camión, debido a que se precisa un montaje industrial para fijar todos los elementos al vehículo y que permita realizar las pruebas con total garantía de seguridad. La comprobación del funcionamiento de la aplicación en entornos reales se efectuará en líneas futuras.

Por tanto, las pruebas se dividen en dos grupos: Validación del sistema sensorial en interiores/exteriores y validación de la versión tentativa de todo el sistema completo en interiores.

6.1. Pruebas del sistema sensorial en laboratorio

Estas pruebas se realizan en un laboratorio, entorno muy controlado y donde resulta más fácil poner en marcha la aplicación. El fin de estas pruebas es aislar el sistema perceptivo en un entorno carente de peligro y de elementos desconocidos como sucedería en un escenario real.

En el capítulo 3 se comentaron las características nominales de los sensores escogidos. Con el fin de caracterizar empíricamente el sensor utilizado se realizaron una serie de medidas y a continuación se exponen los resultados obtenidos.

1. Caracterización de las medidas del ultrasonido a diferentes distancias

En la tabla 6.1 se recoge el resultado de tomar medidas hasta un objeto situado delante de un sensor de ultrasonidos.



Figura 6.1: Elementos utilizados para el montaje en interiores

De estas medidas se desprende que para la frecuencia fijada en el oscilador del microcontrolador que gobierna el sensor, cada 560 tics del contador equivalen a 10 cm. de distancia al objeto. Como era de esperar esta proporción es lineal, por lo que continúa a medida que la distancia al objeto es mayor, como se puede observar en la gráfica 6.2. En cuanto al rango de medida del sensor, la distancia mínima es de 40 cm. y la máxima probada es de 7 m. Este rango de valores es suficiente para nuestra aplicación pues se manejarán distancias de entre 0.5 y 5m. La velocidad del reloj seleccionada para el microcontrolador también parece razonable puesto que no se desborda hasta pasados 10 m. y ofrece una buena resolución (menos de 1 cm.), más que suficiente para cumplir su objetivo.

<i>Distancia al objeto (cm.)</i>	<i>Tiempo en tics del contador (1 Mhz.)</i>
10	2500 - 2600
20	2500 - 2600
30	2500 - 2600
40	2610 - 2620
50	3170 - 3190
60	3740 - 3750
80	4815 - 4820
100	5985 - 5995
120	7165 - 7175
140	8330 - 8340
160	9474 - 9480
180	10640 - 10647
200	11820 - 11825
220	12985 - 12990
240	14128 - 14132
260	15320 - 15326
280	16463 - 16470
300	17627 - 17635
330	19380 - 19390
360	21140 - 21190
390	22890 - 22910

Tabla 6.1: Medida Frontal

También cabe destacar que durante los primeros 40 cm., el valor obtenido por el contador es prácticamente el mismo. Esto es debido a que el sensor necesita un intervalo de espera, para evitar que su propia señal de inicio genere un ECHO. Hemos comprobado este hecho de manera empírica y, aunque proporciona una cierta ambigüedad a muy cortas distancias (< 40 cm.), no es problema para la aplicación pues la distancia mínima manejada será de 50 cm.

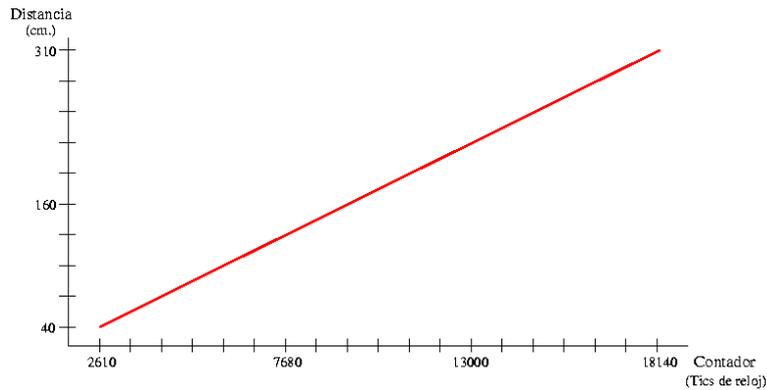


Figura 6.2: Gráfica que muestra que el tiempo de vuelo de la onda ultrasónica es lineal a medida que aumenta la distancia al obstáculo

2. Caracterización de la apertura angular del ultrasonido

La siguiente prueba se realizó para estimar el grado de apertura angular de la onda de ultrasonidos. El objetivo de esta prueba es comprobar si se mantiene constante este ángulo o por el contrario varía con la distancia. Los resultados se pueden observar en la tabla 6.2 que aparece a continuación. Como se puede ver en la tabla, los valores de apertura angular se mantienen aproximadamente en 8 grados y bastante constantes.

<i>Distancia frontal (cm.)</i>	<i>Distancia lateral (cm.)</i>	<i>Apertura del haz (grados)</i>
73	11	7.69
173	27	8.2
273	35	7.3

Tabla 6.2: Apertura del haz

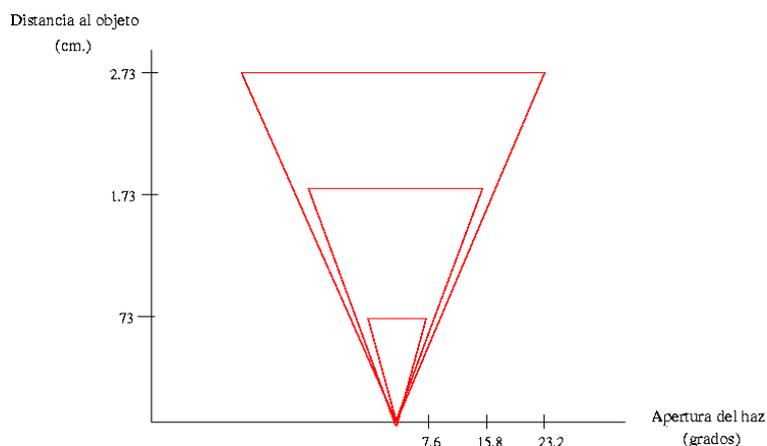


Figura 6.3: Gráfica que muestra la apertura del ángulo de la onda ultrasónica a medida que el tiempo de vuelo es mayor

3. Periodicidad de los disparos de ultrasonido

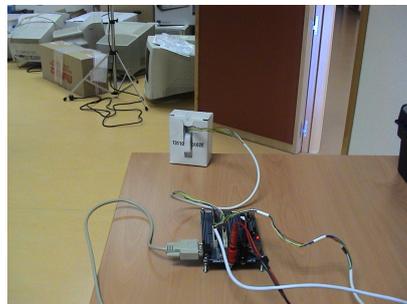
Una vez comprobadas las características nominales de los sensores de ultrasonidos que componen el subsistema s3nar, lo siguiente a comprobar es con qu3 periodicidad deben lanzarse para obtener medidas precisas. En este caso el par3metro determinante es el tiempo que se espera desde que se lanza una medida (se activa el *INIT*) hasta que se dispara la siguiente. Si este tiempo es menor al tiempo que tarda en retornar la onda ultras3nica emitida previamente (se3al *ECHO*), cuando 3sta se reciba interferir3 en la siguiente medida. Por otro lado, si se espera demasiado se corre el riesgo de que los sensores perciban los objetos muy lentamente, resultando poco vivaz para la aplicaci3n. Este intervalo de espera debe ser lo suficiente ajustado para evitar las interferencias y maximizar el n3mero de disparos s3nar por segundo.

Para estimar correctamente este tiempo de espera entre medidas hay que tener en cuenta cu3l es el n3mero de s3nares disponibles. 3sto es debido a que si se van lanzando alternativamente, el tiempo de espera desde que se lanza un sonar hasta que le toca de nuevo a 3ste, es suficiente para que no haya interferencias. Tras varias pruebas del subsistema s3nar con varias frecuencias de medidas, se concluy3 que la frecuencia de disparo m3s adecuada era la de 10 medidas por segundo (independientemente del n3mero de sensores conectados), informaci3n m3s que suficiente para nuestra aplicaci3n. Con esta periodicidad no se producen demasiadas interferencias y las medidas son fiables.

Estas pruebas del sistema s3nar en el laboratorio se completaron comprobando c3mo se comporta ante determinados casos:

4. Comportamiento del ultrasonido ante objetos est3ticos

El objetivo de esta prueba es comprobar que el sensor de ultrasonidos mide correctamente distancias a objetos est3ticos. En este caso se comprob3 qu3 medidas se obtenían al colocar objetos fijos a diferentes distancias ante el sensor. En las figuras se muestra como el sensor apunta a una puerta 6.4(a) y en la figura 6.4(b) se muestra la medida obtenida. Entre las conclusiones obtenidas podríamos destacar la precisi3n de las medidas captadas y la poca oscilaci3n de las mismas.



(a)



(b)

Figura 6.4: Imagen del objeto medido (a) y medidas obtenidas por el sonar (b)

5. Comportamiento del ultrasonido ante objetos móviles

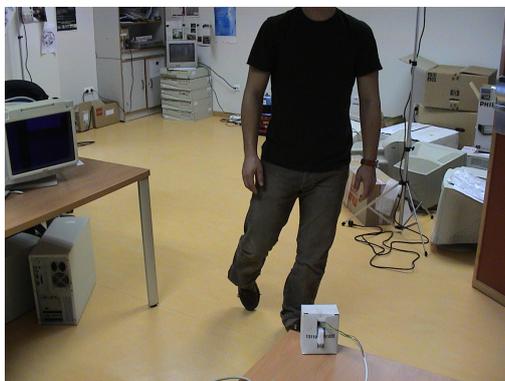
En esta prueba comprobamos si el s3n3nar responde bien ante objetos movi3ndose delante suya. Una persona se coloca delante del sensor de ultrasonidos y se acerca paulatinamente a 3ste. En las im3genes a,b y c se muestra la secuencia real de acercamiento y en la figura 6.5(d) observamos las medidas obtenidas. En la gr3fica se puede observar la curva lineal que representa el acercamiento al sensor. Las medidas obtenidas reflejan correctamente el movimiento efectuado por la persona. En la interfaz gr3fica se ve c3mo la l3nea que representa las medidas se va cerrando a medida que la persona se acerca al sensor. Las nuevas medidas entran por la parte superior de la ventana y el tiempo var3a hacia la parte inferior de la misma.



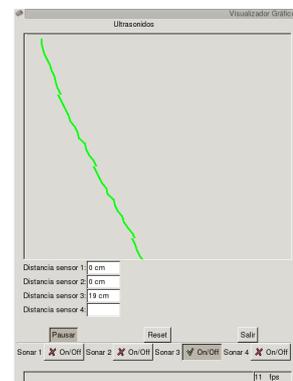
(a)



(b)



(c)



(d)

Figura 6.5: Secuencia de im3genes de una persona acerc3ndose al s3n3nar a distancia larga (a), media (b), corta (c) y las medidas obtenidas por el sensor (d)

6. Comportamiento del ultrasonido ante el cruce de un objeto

El objetivo de esta prueba es que el sensor sea capaz de detectar un objeto móvil (una persona) cruzándose delante de él. Aunque la persona permanece poco tiempo delante del sensor, éste debe ser capaz de percibirla. Esta situación es muy parecida al uso real que se le dará a los sónares para detectar al vehículo que se está rebasando, pero realizada en un entorno de interiores. En la figura 6.6(a) vemos a una persona que cruza por delante del sensor de ultrasonido en dirección a una puerta. En la imagen 6.6(b) se observan las medidas obtenidas por el sónar. En la estela se aprecia un claro y brusco descenso que coincide con el momento en que la persona interrumpe la onda de ultrasonido. Al poco tiempo, se observa que la medida de distancia vuelve a ser mucho mayor, momento que coincide con la salida de la persona del campo visual del sensor. Las medidas obtenidas reflejan que el sensor es capaz de percibir claramente el obstáculo, como lo demuestra la curva en la línea de medidas del sensor.



(a)



(b)

Figura 6.6: Imagen de la persona pasando (a) y medidas obtenidas por el sonar (b)

7. Comportamiento del ultrasonido ante el cruce de múltiples objetos

El objetivo es comprobar que el sensor de ultrasonidos es capaz de percibir varios objetos móviles (en este caso personas) cruzando secuencialmente delante de él. En las figuras 6.7(a), 6.7(b) y 6.7(c) se observa la secuencia de imágenes reales mientras las personas atraviesan la puerta. En la imagen 6.7(d) se muestra la gráfica de medidas captadas por el ultrasonido, donde se puede observar que las medidas son muy precisas y se distinguen claramente los instantes que corresponden al cruce de las personas. En la imagen de la estela aparece una pequeña distorsión ocasionada por la captura de pantalla y que no ocurre en el caso real.



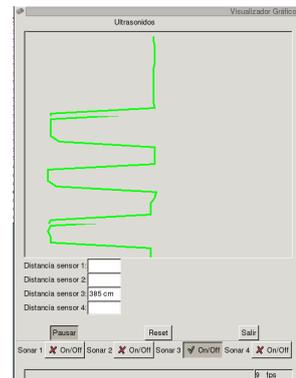
(a)



(b)



(c)



(d)

Figura 6.7: Secuencia de imágenes de tres personas pasando delante del sonar, primera persona (a), segunda persona (b), tercera (c) y las medidas obtenidas por el sensor (d)

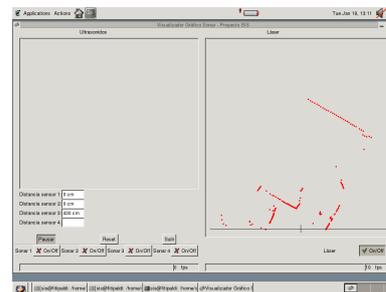
8. Detección de objetos con láser

Una vez probado el sistema s3nar, a continuaci3n comprobamos el funcionamiento del sistema l3ser. Este es un sensor mucho m3s conocido que el anterior y las pruebas han sido mas bien rutinarias para comprobar la velocidad de las capturas y que su funcionamiento es el adecuado.

El fin buscado en la prueba es comprobar que el sensor l3ser mide correctamente los objetos que se encuentran delante de su haz de 180 grados y c3mo al girarlo a su derecha sigue midiendo correctamente sin apenas distorsi3n. Para comprobar la precisi3n del l3ser cuando hay objetos ante 3l, en esta prueba se colocaron una serie de elementos geom3tricos como cajas, mesas, etc. delante del sensor. En la primera imagen 6.8(a) se puede observar el l3ser sobre una silla y su posici3n respecto a las cajas y la mesa de la derecha. En la figura 6.8(b) comprobamos las medidas obtenidas por 3l y como se aprecian varios segmentos correspondientes a las cajas, el borde de la mesa derecha, las mesas de la izquierda y la pared frontal (el segmento m3s largo). En la segunda imagen 6.8(c) se aprecia que se ha girado ligeramente el l3ser a su derecha y como 3ste ha refrescado las medidas y ha variado la representaci3n del entorno 6.8(d). Como era de esperar comprobamos que este sensor funciona con gran precisi3n y nunca se han detectado fallos en su funcionamiento.



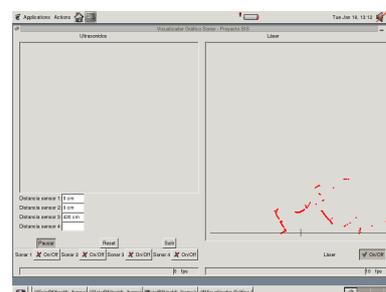
(a)



(b)



(c)



(d)

Figura 6.8: Imagen del l3ser midiendo objetos (a) y medidas obtenidas por el l3ser (b). Imagen del l3ser girado hacia su derecha y midiendo los objetos (c) y figura con las medidas obtenidas (d)

6.2. Pruebas del sistema sensorial en exteriores

Una vez comprobado el correcto funcionamiento del sistema en el laboratorio, hay que probar la validez del sistema sensorial en exteriores. Concretamente, nos desplazamos a una carretera y probamos la aplicación bajo condiciones de tráfico real. Estas pruebas de exteriores sólo se realizaron para el sensor lateral (sónar). El láser es muy pesado y muy difícil de fijar en un turismo convencional. El montaje del sistema puede observarse en la figura 6.9, donde se aprecia la placa controladora de los sónares en una caja, un sensor de ultrasonido en otra caja de menores dimensiones y el ordenador portátil.



Figura 6.9: Montaje del sistema sensorial para exteriores.

9. Percepción de vehículo estáticos a baja velocidad

Buscamos comprobar que el sensor de ultrasonidos ofrece medidas diferentes ante obstáculos estáticos y en ausencia de ellos. También se desea comprobar la calidad de las medidas recibidas ante condiciones de velocidad relativa media, puesto que los vehículos permanecen estáticos. Esta prueba se realiza en un tramo de aparcamiento en el que los coches están estacionados en batería. Dicho tramo es recorrido por nuestro vehículo mientras se toman medidas desde el lateral derecho del mismo utilizando el subsistema sónar.

Como se observa en la figura 6.11, se aprecia claramente cuando el sensor percibe un vehículo y cuando un hueco. Los datos muestran una línea en forma de U que se repite periódicamente y que corresponde a la secuencia hueco-vehículo-hueco. A la vista de los datos, podemos concluir que éstos varían lo suficiente como para poder ser detectados como objetos distintos, tarea de la que se encarga la aplicación de control. También observamos que la frecuencia de captura de medidas sónar es suficiente para condiciones de velocidad relativa media como este caso.

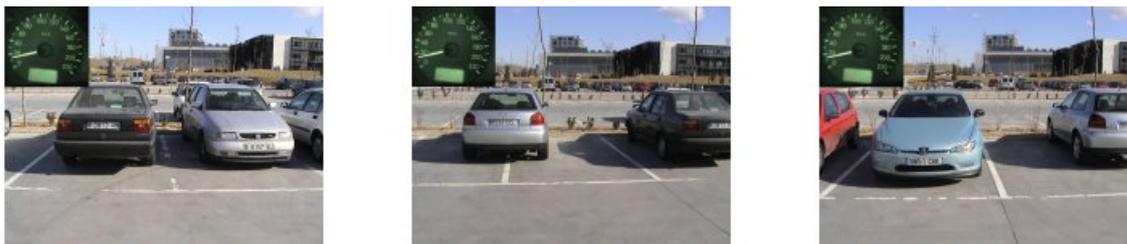


Figura 6.10: Secuencia de fotografías del *parking* real

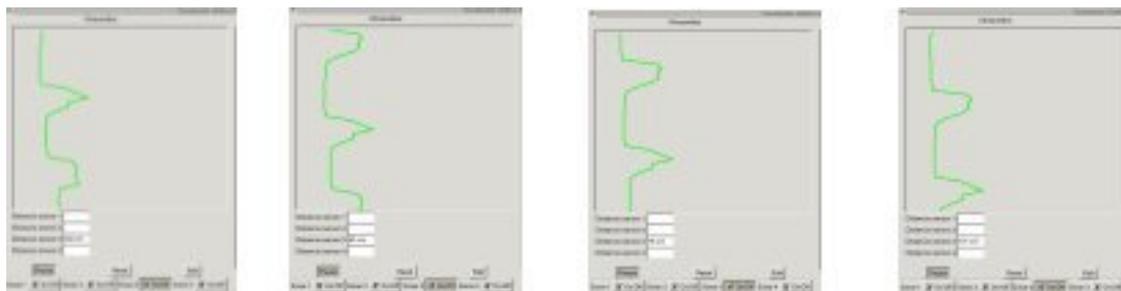


Figura 6.11: Datos obtenidos por el subsistema s3n3nar

10. Percepci3n de un veh3culo ligero a velocidad media

La finalidad de esta prueba es constatar la calidad de las medidas recibidas por el subsistema s3n3nar adelantando a un veh3culo utilitario a 80 Km/h. Para el desarrollo de esta prueba nos dirigimos a una carretera convencional y realizamos un adelantamiento a 80 Km/h. En estas condiciones, la velocidad relativa es media-baja puesto que el veh3culo adelantado circulaba a una velocidad ligeramente inferior que la nuestra. En la figura 6.12 se puede observar la secuencia de im3genes de la maniobra de adelantamiento, durante el inicio, transcurso y fin del mismo. La figura 6.13 muestra, a su vez, los datos obtenidos por el subsistema s3n3nar en dichos instantes. Analizando los datos se desprende que el comienzo del adelantamiento genera un ligero descenso de las medidas de aproximadamente un metro. Este descenso, a pesar de tener peque1as oscilaciones, permanece dentro de unos l3mites razonables y vuelve a incrementarse cuando el veh3culo es rebasado. Las mediciones obtenidas nos parecen suficientes para que la aplicaci3n que analiza los datos sea capaz de distinguir el comienzo y fin del adelantamiento, por lo que consideramos aceptables las medidas ofrecidas por los s3n3ares.

11. Percepci3n de un veh3culo pesado a velocidad alta

Con esta prueba se pretende comprobar el funcionamiento de los s3n3ares adelantando a un cami3n a la m3xima velocidad permitida (120 Km/h). Este test ha sido realizado en una autopista y acelerando nuestro veh3culo hasta alcanzar 120 Km/h. Nuestro objetivo es adelantar al cami3n que tenemos inmediatamente delante de nuestro veh3culo. La secuencia de im3genes de la figura 6.14 ilustra el transcurso de la maniobra. En esta ocasi3n la duraci3n del adelantamiento es mayor que el caso anterior (adelantamiento a un coche), ya que la longitud del cami3n es sensiblemente mayor que la del coche. La figura 6.15 representa



Figura 6.12: Secuencia de fotografías de la maniobra de adelantamiento

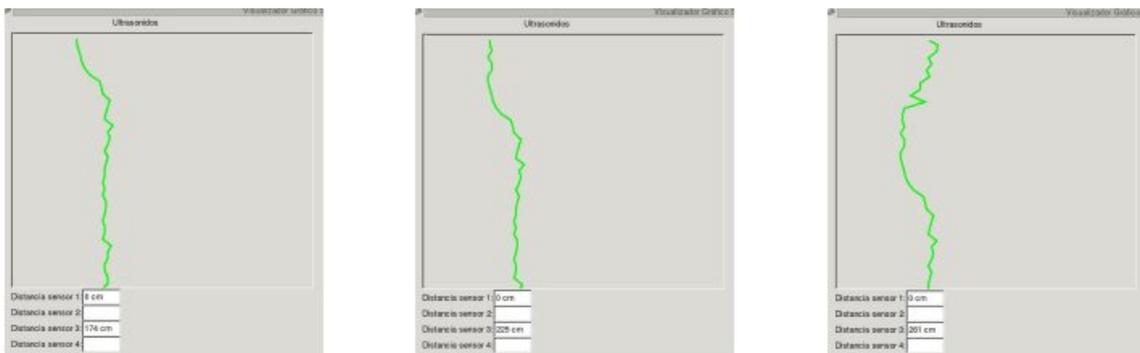


Figura 6.13: Datos obtenidos durante el inicio, transcurso y fin del adelantamiento

los datos obtenidos por los sensores y se aprecia la mayor longitud del vehículo rebasado.

Los datos muestran un descenso bastante claro de la distancia en el momento del inicio del adelantamiento. Este descenso se mantiene relativamente constante hasta el final del mismo, donde se percibe de nuevo un incremento en la distancia. La conclusión principal que obtenemos de esta prueba es que podemos percibir con bastante seguridad el inicio y fin del adelantamiento. Esta labor es vital porque es precisamente la tarea primordial del subsistema s3nar. Adem3s, se observa que a la m3xima velocidad permitida las interferencias aparecen, pero no resultan cr3ticas para la detecci3n del comienzo y fin de la maniobra.



Figura 6.14: Maniobra de adelantamiento real a un camión a 120 Km/h.



Figura 6.15: Medidas obtenidas durante la maniobra de adelantamiento

6.3. Pruebas de la aplicación

Después de verificar el sistema sensorial se realizaron las pruebas al sistema final en conjunto. Como hemos mencionado al comienzo de este capítulo, este proyecto pretende construir una primera versión o prototipo real que funcione adecuadamente en interiores y que lo haga razonablemente bien en exteriores. Está fuera del alcance de esta versión el hacer un sistema definitivo y comercial. Por tanto, a continuación planteamos un caso de prueba consistente en simular el adelantamiento de un vehículo en interiores.

Hemos equipado un robot Pioneer [ActivMedia, 2002] con nuestro sistema sensorial y nuestra aplicación ejecutándose en un portátil que viaja sobre el robot. A través del ordenador portátil, enviamos comandos de movimiento al robot, de manera que le teleoperamos a nuestro gusto a través de una conexión inalámbrica desde otro PC. Dirigimos el robot a un pasillo, donde colocamos una caja simulando el automóvil a adelantar. Activamos nuestro sistema y seguimos la evolución de la aplicación según avanza la maniobra.



Figura 6.16: Secuencia de simulación de adelantamiento usando un robot

La figura 6.16 muestra los instantes significativos de la prueba: Detección frontal, detección lateral y detección del fin de adelantamiento. La figura 6.17 ilustra el estado del sistema en dichos momentos. La imagen superior izquierda ilustra el estado de la aplicación antes de la detección del obstáculo frontal. Se puede ver que no está activado el limitador y que la aplicación está buscando obstáculos frontales. En la imagen superior derecha podemos ver el momento exacto en el que el obstáculo es detectado gracias a la técnica de segmentado del láser

y su posterior análisis. Se observa también que la aplicación ha cambiado su estado (iconos de la izquierda de la imagen) y el limitador se ha activado, ya que en breve comenzará el adelantamiento. Dirigimos el robot a la parte izquierda del pasillo y continuamos la marcha para simular su adelantamiento. Cuando estamos a la altura de la caja, los sensores de ultrasonido detectan el obstáculo (imagen inferior izquierda) y la aplicación transita al último estado: Detección del fin de adelantamiento. Continuamos con la maniobra y cuando sorteamos definitivamente la caja, los sensores de ultrasonido informan a la aplicación y el limitador retorna a su posición original (imagen inferior derecha), dando por terminado el ciclo de adelantamiento. A pesar de que el robot ha sido teleoperado para la prueba, el sistema funciona de manera totalmente autónoma y sin ninguna intervención.

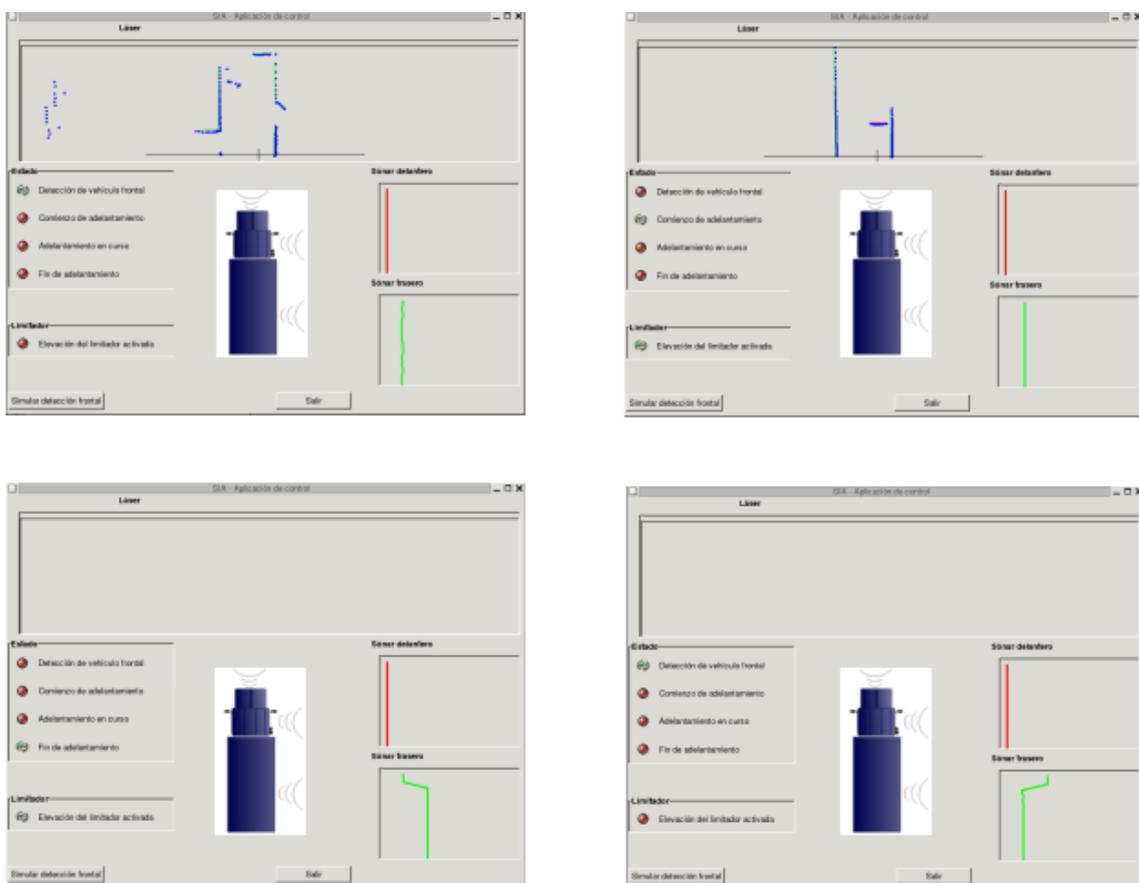


Figura 6.17: Capturas de pantalla obtenidas desde la aplicación de control

Podemos concluir que aunque no hemos probado el sistema completo en un entorno real, el sistema perceptivo y el de control funcionan adecuadamente. En posteriores etapas futuras, se tendrán en cuenta las situaciones del tráfico real y será necesario tener en cuenta multitud de factores adicionales, pero en el caso que nos ocupa, hemos demostrado la validez del prototipo final en un entorno controlado.

Conclusiones y mejoras

En este capítulo se resumen las conclusiones más relevantes que se han extraído durante el tiempo que duró el desarrollo del proyecto SIA (Sistema Inteligente de Adelantamiento), así como las posibles mejoras que se pudieran introducir.

7.1. Conclusiones

En el capítulo 2 se comentó que el principal objetivo que perseguía el desarrollo de este proyecto era el diseño, programación y construcción de un prototipo que agilizará la maniobra de adelantamiento en vehículos pesados, limitados por ley a una velocidad de 90 km/h. El sistema debía detectar el inicio y fin de dicha maniobra para subir y bajar el límite de velocidad respectivamente. Una vez concluido el proyecto, la principal conclusión extraída es que ha sido posible el desarrollo del prototipo precomercial dentro de los plazos de entrega establecidos con los clientes y funcionando correctamente. Se ha desarrollado un sistema sensorial completo funcionando y una versión preliminar de la aplicación de control. El proyecto concluyó con una demostración en el INTA a la espera de una continuación, demostrando que el sistema es viable.

Además se ha desarrollado un proyecto real de ingeniería con clientes reales donde hay que encontrar solución a todos los problemas que puedan aparecer. Para estos proyectos el tiempo de desarrollo y el coste del material son muy importantes y en nuestro caso han sido factores muy tenidos en cuenta.

Además de estos objetivos principales el sistema ha cumplido los subobjetivos:

1. **Estudio sensorial:** Como se describió en el capítulo 3, en una primera fase se llevó a cabo la identificación de las necesidades sensoriales, así como un minucioso estudio de las alternativas sensoriales existentes en el mercado (ultrasonidos, láser puntuales, láser de barrido, láser de barrido multiplano, radares). Este estudio sirvió para decidir las tecnologías sensoriales más convenientes para el proyecto y la disposición de los sensores en el vehículo (tecnología láser para el sensor frontal y ultrasonidos para el sensor lateral). Estos sensores escogidos tras las pruebas experimentales demostraron ser válidos para la aplicación.
2. **Diseño eléctrico y lógico:** Esta fase del proyecto era clave para su desarrollo

y era muy importante realizar un buen diseño. En esta fase comentada en el capítulo 4, se decidieron qué componentes formarían el sistema: Sensores, placas de control, etc., sin olvidar también otros elementos como la alimentación de estos dispositivos o el tipo de interfaces entre ellos (entre la placa controladora de ultrasonidos y el PC, etc.). La conclusión extraída es que este objetivo se ha cumplido ya que el diseño final del sistema está modulado en bloques (subsistema sonda, subsistema láser, PC, alimentación, etc.). Se ha construido una placa controladora que se encarga de la gestión de los sensores de ultrasonido (dispara varios sensores, recibe comandos del PC, envía medidas de los sondas al PC, etc.). Además se ha resuelto el problema de la alimentación de los sensores y de la placa, que puede ser proporcionada por las baterías convencionales de un vehículo pesado.

3. **Programación de bajo nivel y recepción de los datos sensoriales:** Una vez resuelto el diseño del sistema se pasó a programar el microcontrolador de la placa que gestiona los ultrasonidos. El programa que contiene el microcontrolador de esta placa está escrito en lenguaje ensamblador y era necesario dominar interrupciones, envío/recepción a través del puerto serie, etc. En cuanto al láser, sus datos estaban disponibles de una manera más cercana al programador, por lo que únicamente se han utilizado librerías *software* ya existentes, escritas en lenguaje C, para recoger la distancia a los objetos desde el láser.
4. **Programación de alto nivel:** Después de diseñar la arquitectura general del sistema, había que identificar y caracterizar todos los casos que se podían dar en la maniobra de adelantamiento (en curva, adelantamiento simultáneo de varios vehículos, etc.), analizando los datos procedentes de los sensores. Una vez caracterizada la multitud de escenarios y entornos más frecuentes, había que pasar a programar el sistema para que se activen las señales encargadas de modular el regulador de velocidad. Esta casuística no se ha analizado completamente ya que inicialmente se ha obtenido un prototipo básico. La aplicación de control del prototipo sólo tiene en cuenta un caso sencillo de adelantamiento, probado en el laboratorio. Además, era recomendable programar una aplicación para visualizar y chequear el correcto funcionamiento del sistema, de tal manera que se permitiera comprobar visualmente las distancias a objetos y desde donde se pudieran ajustar algunos parámetros sensoriales. En total se han implementado dos aplicaciones de visualización: el Monitor que sirve de herramienta de depuración y visualización de las medidas obtenidas por el sistema sensorial y la interfaz de la aplicación de control.
5. **Pruebas del prototipo:** La fase de pruebas comentada en el capítulo 6, se realizó con el fin de comprobar la viabilidad del sistema y la validez de los sensores empleados. En esta fase se realizaron pruebas del sistema sensorial tanto en exteriores como en interiores y de la aplicación de control para el caso básico de adelantamiento en interiores. Estas pruebas del sistema sensorial se realizaron en el laboratorio y en exteriores, más concretamente en carretera con tráfico real. De esta fase de pruebas se puede concluir en primer lugar que el comportamiento sensorial en las pruebas realizadas es totalmente correcto, lo que hace válidos a estos sensores para el proyecto SIA. Las pruebas de la aplicación

de control se realizaron únicamente en el laboratorio ya que no disponíamos de un camión de pruebas y el sensor láser es muy pesado y complicado de anclar en un turismo convencional. El resultado de las pruebas de la aplicación de control sirvió para demostrar la viabilidad del sistema y de nuevo la validez de los sensores empleados.

Una vez revisados los objetivos perseguidos en la realización del proyecto SIA, éste también debía desarrollarse teniendo en cuenta los requisitos previos. A continuación se analiza el sistema en función de estos requisitos.

El sistema debía ser totalmente autónomo, es decir, no tenía que ser necesaria la intervención humana para el correcto funcionamiento de éste. Esto se ha conseguido gracias al sistema de control autónomo empleado. Aunque el sistema empleado sólo se encarga de detectar una situación concreta de adelantamiento todo lo hace de manera autónoma.

Otro requisito era que el sistema debía funcionar correctamente bajo condiciones ambientales adversas, haciendo uso de sensores apropiados. El estudio sensorial realizado tenía en cuenta este requisito. Los sensores escogidos para la implementación del sistema son capaces de trabajar bajo condiciones extremas. Tanto el sensor láser Sick empleado, que contiene un calefactor, como los ultrasonidos de Polaroid que son estancos, están indicados para funcionar bajo estas condiciones. Aunque estos sensores están capacitados para trabajar en exteriores, aún no se han probado bajo condiciones adversas reales (niebla, lluvia, nieve, etc.).

El sistema además debe funcionar correctamente ante cualquier situación de tráfico como curvas, rectas, carreteras de un solo sentido, etc. Ésto no se ha probado exhaustivamente, aunque este requisito se ha cumplido en parte con la versión preliminar de la aplicación, probada en interiores.

La arquitectura escogida para el sistema debía contener una serie de sensores, conectados a un ordenador convencional a través de un medio de conexión altamente extendido y de fácil adquisición. Esto se ha conseguido desarrollando un sistema de ultrasonidos que se comunica con el ordenador a través de interfaz serie.

Un requisito importante era que los datos fueran enviados al ordenador de control con la rapidez suficiente para ser procesados en tiempo real. Además, no solo se exigía que el envío fuera rápido, sino que fuera lo suficientemente rico en información como para decidir adecuadamente la tarea a realizar. Este requisito se cumple gracias al desarrollo del sistema sensorial que se encarga de controlar los sensores (láser y sónar), enviar la información con bastante celeridad (hasta 20 medidas sónar por segundo y 10 medidas láser por segundo).

El sistema debe funcionar en un vehículo pesado comercial como los que circulan por nuestras carreteras. El sistema debe cumplir una serie de restricciones como tamaño, peso y alimentación. Todas estas restricciones se han tenido en cuenta en el desarrollo del SIA. De este modo, el sistema se alimenta de la batería (12V./24V.) del vehículo sin ser necesaria la instalación de alimentación adicional. Este sistema de alimentación se ha probado en un turismo convencional con el que además se realizaron las pruebas de exteriores. En cuanto al tamaño, el sistema se ajusta a las medidas de los vehículos pesados y es relativamente ligero por lo que no hay problemas de peso.

Además se ha conseguido en parte que este sistema realizado tenga un coste asequible para que sea competitivo en el mercado, cara a la comercialización del sistema

final. El sensor láser encarece mucho el coste ya que su precio es de 4000 euros. En cuanto al resto, se ha encontrado un compromiso entre calidad y precio, garantizando la funcionalidad del sistema. Ésto se ha conseguido principalmente gracias al tipo de *hardware* empleado, como los sensores de ultrasonido Polaroid de 40 euros/und., la realización de la placa controladora de los sónares, que abarata mucho el coste en relación a si se hubiera empleado una tarjeta de adquisición comercial, etc.

7.2. Líneas Futuras

Como se comentó anteriormente sólo se ha terminado una versión preliminar del prototipo SIA, donde se ha desarrollado un sistema sensorial completo con una versión preliminar de la aplicación de control. La principal línea futura de trabajo es dotar a la aplicación de control de la capacidad de detectar más situaciones de adelantamiento (curvas, vías de doble sentido, vías de un solo sentido, etc.). En el proyecto sólo se ha tenido en cuenta el caso de adelantamiento en línea recta y a un único vehículo. Además quedan pendientes las pruebas del sistema completo en exteriores, ya que sólo se realizaron pruebas en exteriores para el sistema sensorial.

Otra línea de trabajo es la de prescindir del sensor láser, ya que éste encarece mucho el coste final del producto. El objetivo sería contar sólo con los sensores de ultrasonidos. Habría que estudiar la viabilidad de esta última alternativa debido a las incertidumbres asociadas a los sensores de ultrasonido.

Bibliografía

- [ActivMedia, 2002] ActivMedia. ARIA reference manual. Technical Report version 1.1.10, ActivMedia Robotics, November 2002.
- [Dickmanns, 1997] E. Dickmanns. Vehicles capable of dynamic vision. In *Proc. Int. Joint Conf. Artificial Intelligence*, pp. 1577–1592, 1997.
- [Foundation, 2004] Gnome Foundation. Gnome developer's home. <http://developer.gnome.org>, 2004.
- [Giles, 2003] Rikke D. Giles. graphical interface development with glade2. Technical report, http://www.kplub.org/glade_tutorial/, 2003.
- [Gómez, 2003] Víctor Manuel Gómez Gómez. Vision based schemas for an autonomous robotic soccer player. In *Proceedings of IV Workshop de Agentes Físicos WAF-2003*, pp 109-120, Universidad de Alicante (Spain), 2003.
- [iRobot, 2005] iRobot. Roomba vacuum cleaner robot. <http://www.irobot.com/>, 2005.
- [Lundh, 1999] Fredrik Lundh. An introduction to tkinter. <http://www.pythonware.com/library/tkinter/introduction/>, 1999.
- [Maxim, 2004] Maxim. Max232/233 manual. Technical report, <http://www.maxim-ic.com>, 2004.
- [Microchip, 2001] Microchip. Microchip pic 16f87x datasheet. Technical report, <http://www.microchip.com>, 2001.
- [Microchip, 2004] Microchip. Mplab ide quick start guide. Technical report, <http://www.microchip.com>, 2004.
- [Moya, 2002] Rodrigo Moya. Programación en el entorno gnome. <http://libros.es.gnome.org/librognome/librognome/librognome/>, 2002.
- [Naba Kumar, 2002] Biswapesh Chattopadhyay Naba Kumar, Andy Piper. Anjuta ide manual. Technical report, <http://anjuta.sourceforge.net>, 2002.
- [Noga, 1999] Markus L. Noga. legos documentation. Technical report, <http://www.noga.de/legOS/>, 1999.

- [Qt, 2004] Qt. Qt reference documentation. <http://doc.trolltech.com/3.3/>, 2004.
- [T. C. Zhao, 1997] Mark Overmars T. C. Zhao. A graphical user interface toolkit for x. <http://www.public.iastate.edu/~xforms/homepage.html/>, 1997.
- [Volvo, 2005] Volvo. Research & technology. <http://www.volvo.com/group/global/en-gb/Volvo+Group/researchandtechnology/t.update/>, 2005.
- [Wang y Thorpe, 2002] Chieh-Chih Wang y Chuck Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2918–2924, Washington DC, May 2002.