



**Análisis de Prestaciones de Sistemas de
Comunicaciones Inalámbricas en Aplicaciones de
Robótica: Sistema Hardware**

Proyecto Final de Carrera

AUTOR: José Lao Albadalejo

DIRECTOR: Joan Oliver Malagelada

Ingeniería Técnica de Telecomunicaciones
Esp. Sistemas Electrónicos

Bellaterra 16 de Junio de 2009

ÍNDICE

1.	Introducción.....	5
1.1	Contenidos.....	6
2.	Planteamiento general.....	7
2.1	Descripción funcional.....	7
2.1.1.	Estación base	7
2.1.2.	Periféricos.....	8
3.	Planificación del proyecto	10
3.1	Tecnologías inalámbricas	10
3.2	Microcontrolador ATmega8.....	10
3.3	Control del servomotor y el sonar	11
3.4	Interfaz gráfica.....	11
3.5	Implementación Bluetooth	11
3.6	Implementación RF	11
4.	Elección de dispositivos	12
5.	Control de los periféricos	14
5.1	Estudio del ATmega8	14
5.1.1.	Hardware del microcontrolador.....	14
5.1.2.	Software del microcontrolador	15
5.1.3.	Realizar el control del servomotor y el sonar	17
5.1.3.1.	Servomotor	17
5.1.3.2.	Sonar MaxSonar-EZ4	18
5.1.4.	Control mediante el ATmega8 de los dispositivos.....	18
5.1.4.1.	Control del servomotor.....	18
5.1.4.2.	Control del sonar	20
5.1.4.3.	Comunicación con el microcontrolador	21
5.2	Interfaz gráfica.....	22
5.2.1.	Microsoft Visual Basic	22
5.3	Comunicación vía Bluetooth	22
5.3.1.	Estudio del módulo Bluemore200	22
5.3.2.	Análisis de la comunicación	24
5.3.3.	Conexionado.....	26
5.3.4.	Funcionamiento	27

5.3.5.	Resultados.....	28
5.3.6.	Trabajo futuro	28
5.4	Comunicación vía Radiofrecuencia.....	29
5.4.1.	Estudio del módulo RTF-DATA-SAW.....	29
5.4.2.	Análisis de la comunicación.....	30
5.4.3.	Conexionado.....	32
5.4.4.	Funcionamiento	32
5.4.5.	Resultados.....	34
5.4.6.	Trabajo futuro	35
6.	Conclusiones.....	36
7.	Bibliografía.....	38

ÍNDICE DE FIGURAS

Figura 1.	Esquema general	7
Figura 2.	Esquema de la estructura de la Estación Base	8
Figura 3.	Esquema de la estructura de los Periféricos.....	9
Figura 4.	Planificación del proyecto.....	10
Figura 5.	Arquitectura del ATmega8.....	14
Figura 6.	Características del ATmega8	15
Figura 7.	Entorno de trabajo del AVR Studio	16
Figura 8.	Entorno de trabajo del PonyProg	17
Figura 9.	Montaje del servomotor y el sonar.....	17
Figura 10.	Conexionado del servomotor	18
Figura 11.	Conexionado del sonar.....	18
Figura 12.	Etapas del programa en ensamblador del servomotor.....	19
Figura 13.	Pulso de control del servomotor.....	19
Figura 14.	Etapas del programa en ensamblador del sonar	20
Figura 15.	Salida PW del sonar	21
Figura 16.	Icono MSComm.....	22
Figura 17.	Esquema general del conexionado del Bluemore200	23
Figura 18.	Entorno de la aplicación del <i>USB Dongle</i>	24
Figura 19.	Características de la conexión punto a punto.....	25
Figura 20.	Placa de configuración del Bluemore200	25

Figura 21. Software de configuración del Bluemore200.....	26
Figura 22. Conexionado del sistema Bluetooth.....	26
Figura 23. Prototipo del sistema Bluetooth	27
Figura 24. Aplicación Bluetooth	28
Figura 25. Esquema general del conexionado del RTF-DATA-SAW	29
Figura 26. Protocolo RS232	30
Figura 27. Conector DB9	31
Figura 28. MAX232	31
Figura 29. Conexionado del sistema RF.....	32
Figura 30. Prototipo del sistema de RF	33
Figura 31. Aplicación de RF.....	34

1. Introducción

Hoy en día, cada vez es más común el uso de robots, tanto en la industria como en otros ámbitos de la vida cotidiana. La robótica se ha convertido en un elemento de uso esencial para muchas personas y empresas, debido a que el empleo de todo tipo de robots minimiza esfuerzos y facilita la ejecución de multitud de tareas que pudieran resultar peligrosas o repetitivas para el hombre. El auge de la robótica está intrínsecamente relacionado con el desarrollo de la tecnología, con el abaratamiento de costes y con la reducción del tamaño de los dispositivos microelectrónicos.

Una característica importante de la robótica es la comunicación entre sistema base y robot que puede establecerse de forma remota. Ello representa la base del proyecto que se describe a continuación, el cual se descompone de dos partes, una por cada miembro del proyecto: Sistema software y sistema hardware.

En el sistema software analizaremos las diferentes tecnologías inalámbricas (características, funcionamiento, seguridad, etc.), se realizará una comparativa de los diferentes módulos de comunicación y finalmente decidiremos aquellos que nos interesa para la implementación en Radiofrecuencia (RF) y Bluetooth. En este sistema también estudiaremos la interfaz gráfica que se utilizará, así como los programas creados en este entorno para realizar las implementaciones.

En el sistema hardware trataremos de realizar el control de dos periféricos de forma independiente, un servomotor y un sonar, que nos servirán como ejemplo para analizar una posible comunicación entre varios robots y un PC. Por lo tanto, en este apartado analizaremos a fondo los diferentes componentes que harán posible tanto la comunicación, vía RF y Bluetooth, como el control de los diferentes dispositivos.

Por lo tanto, los objetivos del proyecto son:

- Utilizar la comunicación inalámbrica como elemento de autonomía para un robot.
- Analizar los sistemas inalámbricos para aplicarlos a la robótica, profundizando en el sistema Bluetooth y en el sistema de Radiofrecuencia.
- Aplicación a diferentes robots en un mismo instante (comunicación multiterminal): Crear una red de control sobre diferentes dispositivos y a la vez, permitir la comunicación entre ellos.

Cabe destacar que para nosotros la profundización en ambas tecnologías tiene su base en las telecomunicaciones, aunque como se verá en posteriores apartados intervienen diferentes disciplinas, como teoría de control, lenguajes de programación, etc.

1.1 Contenidos

La exposición del trabajo la estructuramos en los siguientes capítulos. En primer lugar, en el capítulo Planteamiento General, se describe el sistema de comunicación de forma global, en el cual se presentan los diferentes bloques funcionales que lo componen: Estación base y periféricos. A continuación, se plasma a través de Microsoft Project la planificación del proyecto y se realiza una breve explicación de cada una de las tareas.

Posteriormente, remarcamos los criterios de selección que nos han llevado a elegir las dos tecnologías de implementación que se trabajaran en nuestro proyecto, así como el hardware seleccionado. Seguidamente, hablaremos de la interfaz gráfica que nos va a permitir realizar el control de los periféricos.

A continuación se expone en detalle el sistema Bluetooth y el de Radiofrecuencia, donde analizaremos los módulos de comunicación, la red de comunicación que se va a establecer para gestionar los periféricos, el conexionado del sistema, el funcionamiento y los resultados obtenidos. Se añaden diferentes especificaciones para trabajos futuros. Por último, veremos las conclusiones que se pueden extraer de nuestro proyecto.

Añadir que al ser un proyecto realizado por dos personas, en los contenidos de ambas memorias, existen distintos puntos en común como son: Introducción, planificación y conclusiones.

2. Planteamiento general

El objetivo de este planteamiento es proporcionar una visión general, a modo de bloques funcionales, de los sistemas que se van a implementar.

Nos hemos planteado desarrollar un prototipo de comunicaciones mediante Bluetooth y Radiofrecuencia, mediante el cual a través de una interfaz gráfica, seamos capaces de controlar de forma remota un servomotor y un sonar. Señalar que ambos periféricos se tratan como ejemplo del método utilizado para establecer la comunicación host y robot. Por lo tanto, aunque ambas implementaciones tengan sus características, las cuales se analizarán posteriormente, a nivel de bloques ambos sistemas son equivalentes. Estos subsistemas aparecen reflejados en la Figura 1 y son los siguientes:

- Estación Base: Constituye la interface con el usuario, y está implementada mediante un conjunto teclado-ratón y un enlace vía Bluetooth o Radiofrecuencia con el robot.
- Periféricos: Constituido por el microcontrolador, un módulo Bluetooth o de Radiofrecuencia y los elementos sobre los cuales queremos actuar, un servomotor y un sonar.

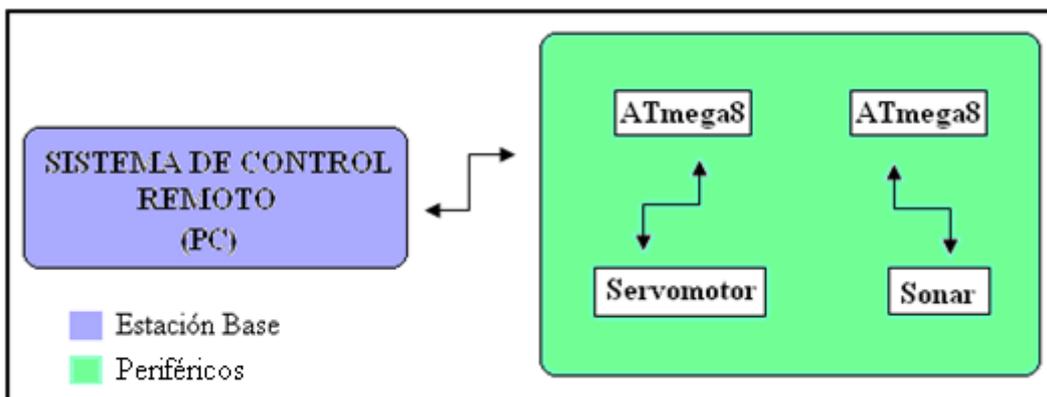


Figura 1. Esquema general

2.1 Descripción funcional

Pasamos ahora a describir a nivel de bloques cada uno de estos subsistemas.

2.1.1. Estación base

Formada por el PC, donde una aplicación software desarrollada en Microsoft Visual Basic es la encargada de gobernar un servomotor y un sonar, y a la vez recibir información de éstos. Dicho control se realiza a través de Bluetooth o de

Radiofrecuencia, a través del puerto serie del ordenador. En la Figura 2 vemos el esquema de la estación base.

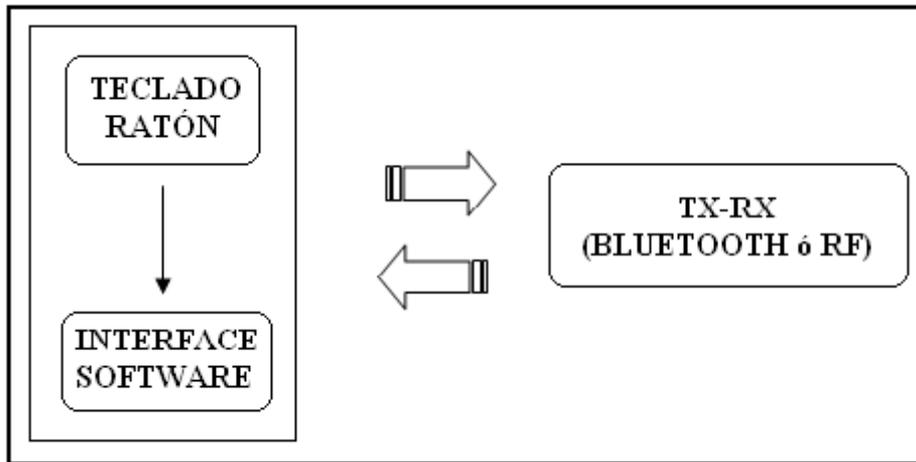


Figura 2. Esquema de la estructura de la Estación Base

2.1.2. Periféricos

Llamamos periféricos al conjunto independiente de cada uno de los microcontroladores con el servomotor y el sonar. Esta plataforma, la cual se aparece en la Figura 3, está compuesta por varios sistemas:

- Alimentación: Basado en un regulador de tensión (7805), donde se obtiene una tensión de salida de 5V.
- Módulos de comunicación: Módulos Bluemore200 en el caso de comunicación Bluetooth y módulos RTF-DATA-SAW en el caso de comunicación por Radiofrecuencia. Permiten realizar una comunicación asíncrona, TX-RX, con la estación base.
- Microcontrolador ATmega8: Dispositivo encargado de interpretar las órdenes de la estación base y realizar un control sobre los diferentes dispositivos a gobernar. Sobre el robot podría ser la unidad central de proceso o bien un módulo autónomo encargado de las comunicaciones.
- Servomotor y Sonar: Conjunto formado por un servomotor, realizará un recorrido de 180°, y un sonar, el cual detectará cualquier obstáculo a una distancia de 2 metros.

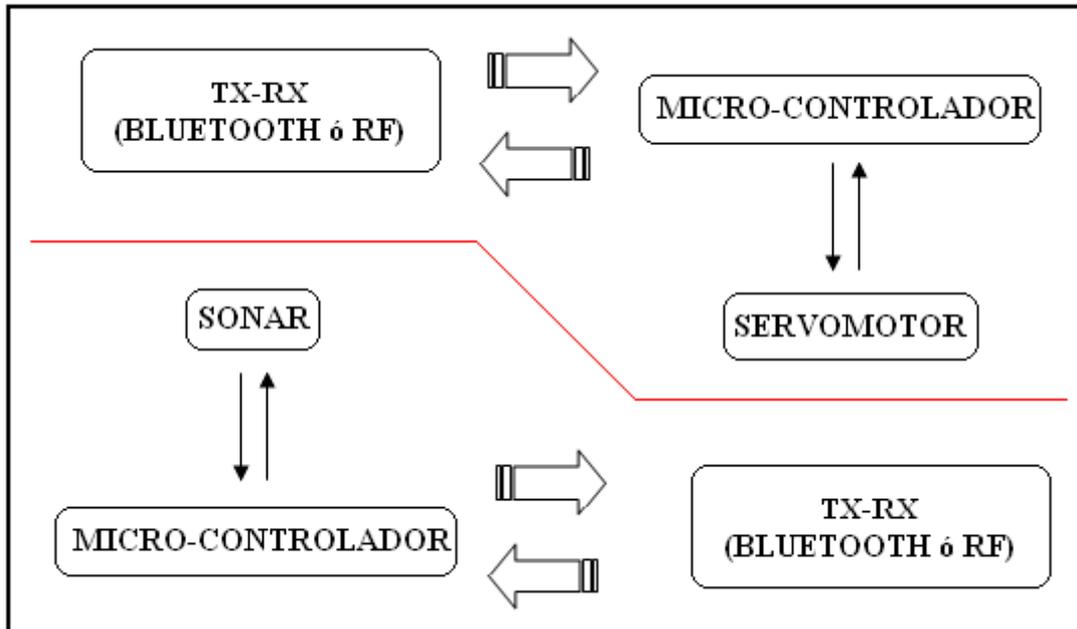


Figura 3. Esquema de la estructura de los Periféricos

3. Planificación del proyecto

En la planificación de nuestro proyecto hemos destacado las etapas que se describen a continuación y que se detallan en orden cronológico en la Figura 4.

	Nombre de tarea	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	☐ Proyecto Final de Carrera	lun 03/11/08	vie 05/06/09		
2	☐ Tecnologías Inalámbricas	lun 03/11/08	vie 19/12/08		
3	Información General	lun 03/11/08	vie 21/11/08		Elisabeth García Mancebo
4	Hardware	lun 24/11/08	vie 12/12/08	3	Elisabeth García Mancebo
5	Elección de Dispositivos	lun 15/12/08	vie 19/12/08	4	Elisabeth García Mancebo; José Lao Albadalejo
6	☐ Micro-controlador ATmega8	lun 03/11/08	vie 21/11/08		
7	Hardware	lun 03/11/08	vie 14/11/08		José Lao Albadalejo
8	Software	lun 17/11/08	vie 21/11/08	7	José Lao Albadalejo
9	☐ Control servomotor y sonar	lun 24/11/08	vie 16/01/09		
10	ATmega8-Periféricos	lun 24/11/08	vie 16/01/09	8	José Lao Albadalejo
11	☐ Interfaz Gráfica	mié 07/01/09	vie 16/01/09		
12	Microsoft Visual Basic	mié 07/01/09	vie 16/01/09	5	Elisabeth García Mancebo
13	☐ Implementación Bluetooth	lun 16/02/09	vie 17/04/09		
14	Estudio del Bluemore200	lun 16/02/09	vie 20/02/09	10	Elisabeth García Mancebo; José Lao Albadalejo
15	Analizar Comunicación	lun 02/03/09	vie 13/03/09	14	José Lao Albadalejo
16	Funcionamiento	lun 16/03/09	vie 03/04/09	15	José Lao Albadalejo
17	Extraer Resultados	mar 14/04/09	vie 17/04/09	16	Elisabeth García Mancebo; José Lao Albadalejo
18	☐ Implementación RF	lun 23/02/09	vie 05/06/09		
19	Estudio del RTF-DATA-SAV	lun 23/02/09	vie 27/02/09	10	Elisabeth García Mancebo; José Lao Albadalejo
20	Analizar Comunicación	lun 20/04/09	vie 01/05/09	19	José Lao Albadalejo
21	Funcionamiento	lun 04/05/09	vie 29/05/09	20	José Lao Albadalejo
22	Extraer Resultados	lun 01/06/09	vie 05/06/09	21	Elisabeth García Mancebo; José Lao Albadalejo

Figura 4. Planificación del proyecto

3.1 Tecnologías inalámbricas

Este es el punto de partida del proyecto. Analizamos las diferentes tecnologías inalámbricas existentes (características, funcionalidad, seguridad, etc.), y realizamos un estudio de mercado de los componentes asociados a cada una de las tecnologías para poder establecer una comparativa teniendo en cuenta factores como: tasa de transmisión, consumo, precio, etc.

3.2 Microcontrolador ATmega8

Podríamos decir que el microcontrolador ATmega8 es una de las piezas claves en todo el prototipo. Es por ello, que merece una etapa en la planificación del proyecto, en la cual se analiza el hardware del ATmega8, así como el software asociado a éste y tratamos pequeños programas a modo de ejemplo para ir familiarizándonos con el microcontrolador.

3.3 Control del servomotor y el sonar

Dos de los periféricos que se utilizan, a modo de ejemplo para analizar los módulos de comunicación host y robot, son un servomotor y un sonar. Es en esta parte de la planificación donde estudiamos las diferentes técnicas para tener control sobre ambos periféricos y se realiza el programa óptimo en el ATmega8 para gobernarlos.

3.4 Interfaz gráfica

El PC, nuestra estación base, requiere de un software encargado de realizar el control sobre los diferentes periféricos. En este apartado se analiza el software utilizado para establecer la comunicación y el componente que nos permitirá una conexión a través del puerto serie con los módulos inalámbricos.

3.5 Implementación Bluetooth

Es en esta etapa donde entramos a fondo en el prototipo Bluetooth. Se empieza analizando el módulo encargado de establecer la comunicación (Bluemore200), el software asociado, así como las posibilidades que nos ofrece para establecer la red de comunicación. Seguidamente se realizan diferentes pruebas punto a punto, para acabar diseñando la placa que permita gobernar todo el sistema. Como en todo diseño electrónico, se acaban extrayendo resultados y diferentes especificaciones para futuros trabajos.

3.6 Implementación RF

Es una etapa muy similar a la anterior, en lo que ha estudio se refiere, pero muy diferente a la hora de implementación. Se han de trabajar con niveles RS232, lo que conlleva el estudio del chip MAX232 y las diferentes pruebas para ver que los datos se reciben correctamente. En este caso, el módulo de comunicación es el RTF-DATA-SAW. Igual que en la etapa anterior, se analizan las posibilidades de la comunicación y se diseña la placa del prototipo para gobernar todo el sistema, se concluye extrayendo resultados y especificaciones para futuros trabajos.

4. Elección de dispositivos

De las diferentes tecnologías explicadas con detalle en el sistema software, el sistema de control se realizará mediante tecnología Bluetooth y vía Radiofrecuencia, debido a la mayor versatilidad de ambas con respecto a las otras tecnologías inalámbricas y la gran variedad de productos que presentan, frente a la escasez de productos de ZigBee y Wi-Fi adaptados a los objetivos del proyecto.

La elección de estas tecnologías es debido a la complejidad de implementación de Wi-Fi y ZigBee. En primer lugar, Wi-Fi necesita puntos de acceso directo a red en aquella zona donde se quiera establecer la comunicación, y esto restringe la funcionalidad del proyecto; ZigBee sólo nos permite la comunicación punto a punto entre dos dispositivos, además de ser una tecnología que varía mucho dependiendo de las características de los componentes utilizados en la comunicación.

En cuanto a la elección de los componentes de las tecnologías seleccionadas hemos impuesto una serie de criterios que tiene que cumplir nuestra red de comunicación y a partir de ahí seleccionado aquellos componentes que más se acoplan a nuestras perspectivas.

En primer lugar, la comunicación debe abarcar unos 100 metros aproximadamente, es por eso que buscamos componentes que soporten distancias medias; En segundo lugar, cada elemento de la comunicación debe ser capaz de transmitir y recibir datos, hecho que comporta que los componentes incorporen en un mismo chip TX/RX; En tercer lugar, nos interesa una velocidad de transmisión de datos lo más alta posible: esto no quiere decir que la velocidad de transmisión/recepción sea elevada, ya que al influir otros aspectos como el económico (componentes baratos), la velocidad se ve muy reducida; Otro aspecto a tener en cuenta es la tensión a la que se alimentan los dispositivos. A la hora de implementar es más sencillo que ambas tecnologías se alimenten con la misma tensión para simplificar el circuito. Y por último el coste unidad del dispositivo (nos interesa el más económico, siempre que cumpla las características requeridas por la comunicación que se quiere establecer).

De acuerdo con los resultados obtenidos del estudio (sistema software) se han elegido los componentes:

Bluetooth: Módulo de comunicación Bluemore200 → Módulo emisor/receptor (transceptor). Permite la comunicación half-duplex vía Bluetooth entre dos estaciones con

una velocidad desde 1200 a 230400 baudios. Tensión de alimentación +3.3Vdc y +5Vdc, consumo de 65 mA el master y de 60 mA el slave. Dispone de módulo USART con salida RS232.

RF: Módulo de comunicación RTF-DATA-SAW → Módulo emisor/receptor (transceptor). Permite la comunicación half-duplex vía radio entre dos estaciones con una velocidad de 2400 baudios. Tensión de alimentación +5Vdc, consumo de 4.5mA el transmisor y de 2.5 mA el receptor, frecuencia 433.92 MHz. Conforme a la norma EN 300-220 con disponibilidad del correspondiente certificado.

5. Control de los periféricos

En este capítulo empezaremos hablando del microcontrolador ATmega8, hardware y software, y seguiremos con la interfaz gráfica utilizada en la estación base, para acabar concluyendo con las implementaciones Bluetooth y RF, de donde se extraerán diferentes resultados.

5.1 Estudio del ATmega8

En este punto se presenta el microcontrolador que nos permitirá ser capaces de tener un control remoto del servomotor y el sonar. Se detallan las características intrínsecas del procesador y el software asociado para realizar diferentes programas. Toda la información que se plasma en este documento aparece en detalle y con pequeños ejemplos de programas en el enlace bibliográfico [1].

5.1.1. Hardware del microcontrolador

La arquitectura de los procesadores ATmega corresponde a una arquitectura RISC, Figura 5, donde los principales módulos de los que se compone son:

- La unidad aritmética lógica.
- El banco de 32 registros.
- El contador de programas.
- El registro de instrucciones y el módulo decodificador de instrucción.
- El controlador de interrupciones.

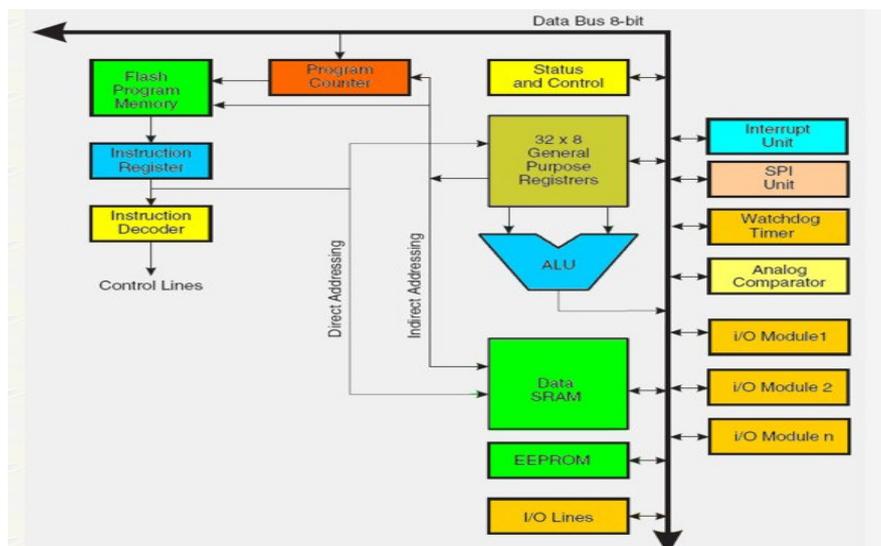


Figura 5. Arquitectura del ATmega8

Alrededor de la CPU y unidos por el bus de datos hay un conjunto de módulos que complementan la operatividad de la CPU:

- La memoria flash que es la encargada de contener el programa y memorias SRAM y EEPROM para almacenar datos y realizar el reinicio desde el microcontrolador.
- Timers, generadores de reloj interno, de sincronismo con reloj externo o *watchdogs*.
- Módulos de entrada/salida y contadores que se comunican con el exterior mediante puertos de entrada/salida.
- Unidades de comunicación basadas en protocolos SPI y USART.
- Y módulos analógicos, como ahora contadores y conversores analógico-digitales.

Las características del microcontrolador ATmega8 utilizado en nuestras implementaciones aparecen resumidas en la Figura 6.

	ATmega8
CPU	Microcontrolador AVR de 8 bits de arquitectura RISC
Memoria FLASH	8 Kbytes
Memoria EEPROM	512 bytes
Memoria SRAM	1 Kbyte
Puertos de E/S	23
Timer/counters	3
USART	1
Puertos SPI	1
Canales de E/S analógicos con ADC	6(8)

Figura 6. Características del ATmega8

5.1.2. Software del microcontrolador

En nuestro proyecto hemos utilizado la forma más simple y barata de trabajar con un microcontrolador ATmega8, la cual consiste en programarlo a través del puerto ISP mediante una placa simple con regulación de tensión, conector de programación con ISP y expansión de pines. Dicha programación se realiza a través del puerto paralelo

mediante el programa PonyProg, al cual se le carga el programa en C o en ensamblador del AVR Studio.

AVR Studio es el entorno de desarrollo integrado gratuito para escribir y depurar aplicaciones en el entorno Windows 9x/NT/200/XP que incluye un ensamblador y un simulador.

El WinAVR es una plataforma software abierta que contiene herramientas de desarrollo para los microcontroladores RISC de Atmel para plataforma Windows. Construida por Eric Weddington incluye el compilador GNU GCC para C y C++ para los AVR.

En la Figura 7 aparece el entorno de trabajo de un programa AVR Studio, donde tenemos en el centro de la imagen el código que vamos redactando, y a la derecha aparecen los diferentes módulos sobre los cuales podemos modificar sus registros o visualizar como varían al depurar el programa.

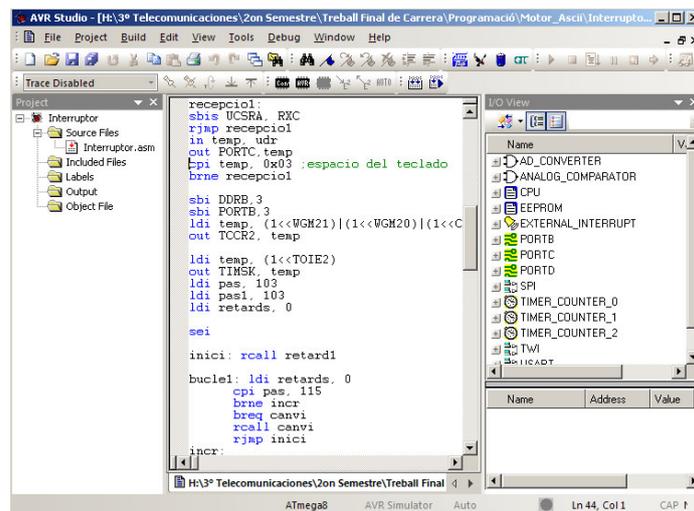


Figura 7. Entorno de trabajo del AVR Studio

Para la programación de los microcontroladores existen diferentes herramientas comercializadas tanto por Atmel como por otras empresas. Como se comentó anteriormente nosotros trabajaremos con el puerto ISP (MISO, MOSI y CLK) a través del puerto paralelo mediante el programa PonyProg, un software con entorno de programación fácil disponible para Windows 95/98/ME/NT/2000/XP y Intel Linux.

En la Figura 8 observamos el entorno de trabajo del PonyProg.

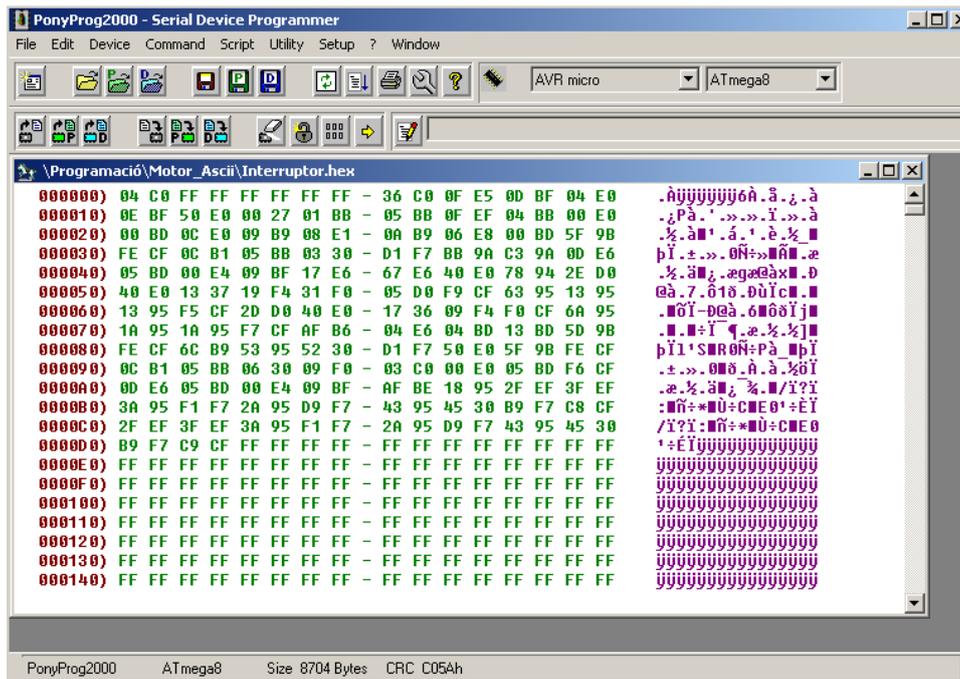


Figura 8. Entorno de trabajo del PonyProg

5.1.3. Realizar el control del servomotor y el sonar

El proyecto se fundamenta en el control remoto, vía Bluetooth o vía RF, de un servomotor y un sonar. Se creará una aplicación en Visual Basic, que se encargará de gestionar el arranque y la parada de los periféricos, así como visualizar la posición del servomotor y la distancia detectada por el sonar. En la Figura 9 podemos ver el montaje conjunto de ambos dispositivos.

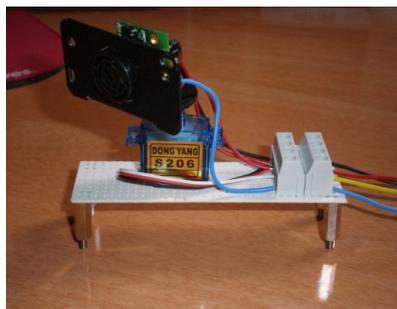


Figura 9. Montaje del servomotor y el sonar

5.1.3.1. Servomotor

El servomotor necesita de una alimentación de +5V y tierra, realizándose su control a través de un único cable que recibe un pulso de amplitud variable a una frecuencia de 50 Hz. El pulso se realiza a través de la función Output Compare del ATmega8 y va desde 1 ms (eje del motor a un lado) hasta 2ms (eje del motor en el otro lado), siendo el punto medio (con un pulso de 1,5 ms) el punto en el que el eje del motor queda en posición intermedia. En la Figura 10 se observa el conexionado del servomotor con el ATmega8.



Figura 10. Conexión del servomotor

5.1.3.2. Sonar MaxSonar-EZ4

Por otro lado, tenemos el sonar MaxSonar-EZ4, el cual ofrece diferentes salidas para analizar los diferentes obstáculos que encontramos. En nuestro caso, utilizamos la salida PW, donde una señal pulso varía su amplitud en función de la distancia a la que se encuentra el obstáculo. En la Figura 11 se observa el conexionado del sonar con el ATmega8.

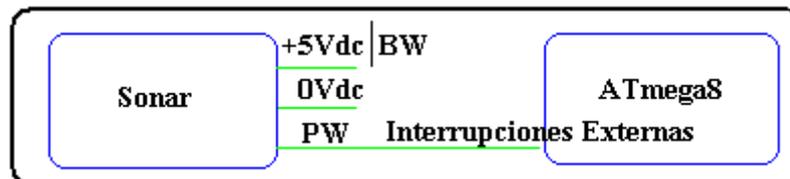


Figura 11. Conexión del sonar

5.1.4. Control mediante el ATmega8 de los dispositivos

Este es uno de los puntos más interesante e importante de todo el montaje, ya que es la base para realizar el control remoto de nuestro prototipo. Empezaremos hablando de cómo es posible con el microcontrolador ATmega8 gobernar el servomotor y el sonar, y cómo éste envía las diferentes señales hacia los módulos Bluetooth y RF a través del módulo USART.

5.1.4.1. Control del servomotor

Como hemos comentado anteriormente el servomotor se controla a través de pulsos con diferente duración, lo cual nos permite posicionar el servomotor. La forma de generar estos pulsos mediante el ATmega8 es a través de la función Output Compare, en la Figura 12 se observa la cronología de etapas del programa en ensamblador.

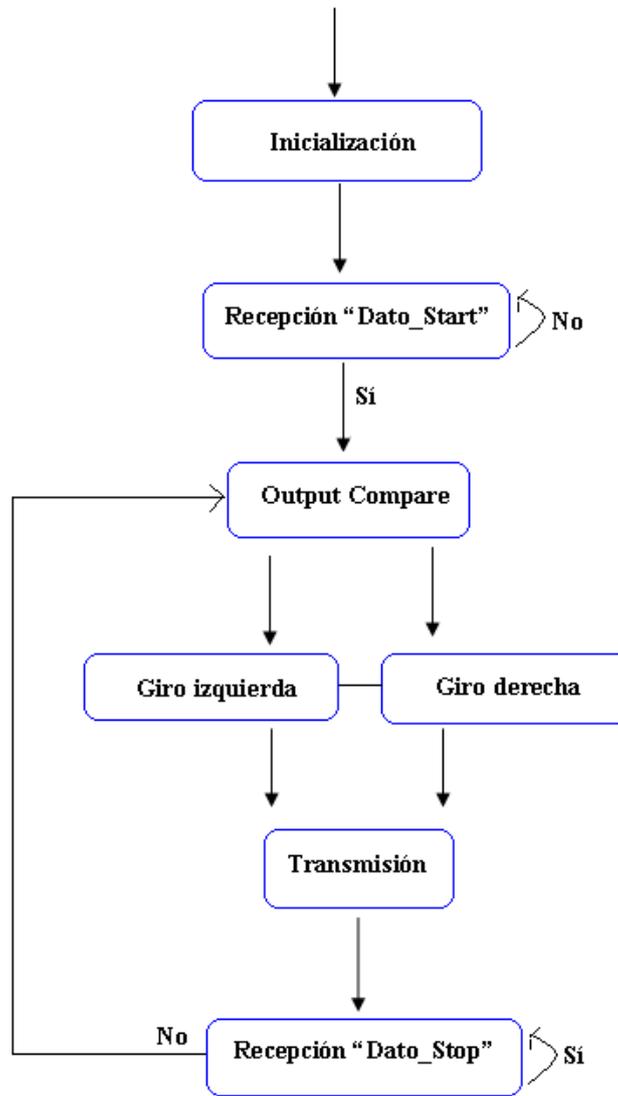


Figura 12. Etapas del programa en ensamblador del servomotor

En la Figura 13 aparece reflejado el pulso que generamos en el ATmega8.

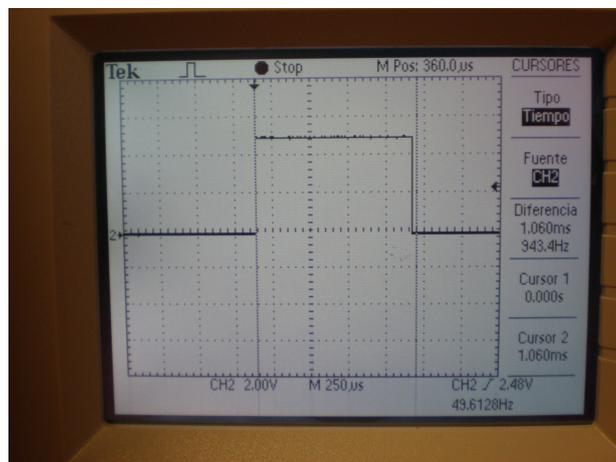


Figura 13. Pulso de control del servomotor

5.1.4.2. Control del sonar

Otro de los dispositivos es el sonar MaxSonar-EZ4, donde hemos comentado que utilizamos la salida de la señal pulso y que en función de la amplitud de ésta obtenemos la distancia del obstáculo. Para ello, hacemos servir las dos interrupciones externas del ATmega8, donde una controla el comienzo del pulso y la otra interrupción, la bajada de éste. Entonces, simplemente activamos un *timer*, donde el valor obtenido lo relacionamos con la distancia del objeto. En la Figura 14 vemos la cronología de etapas del programa en ensamblador.

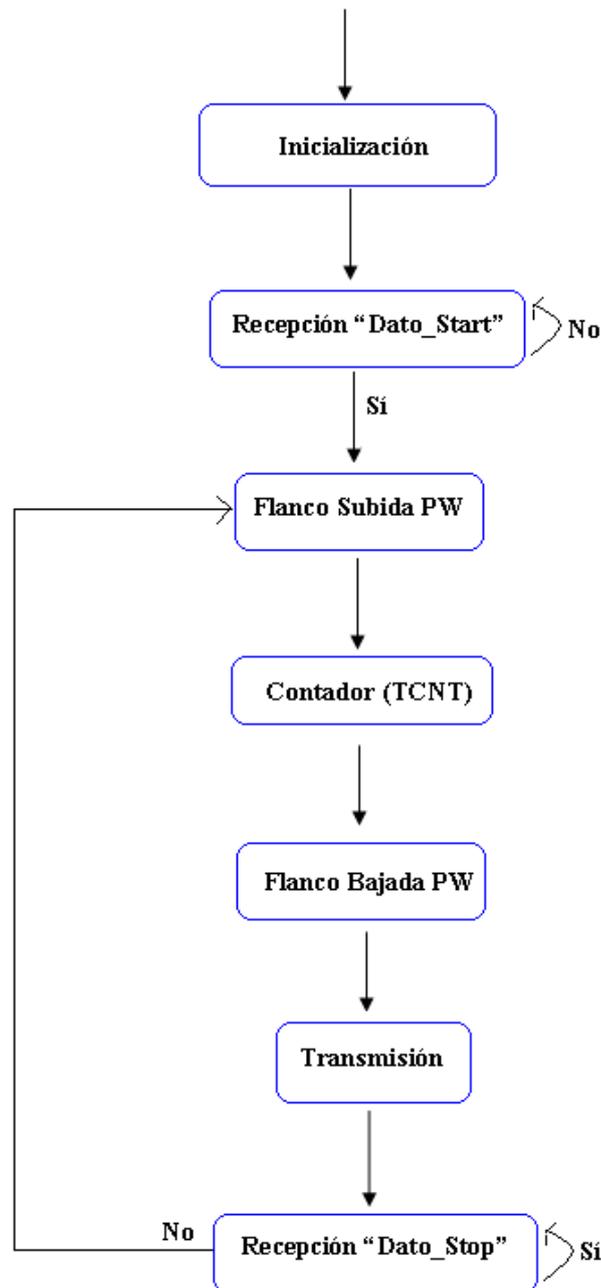


Figura 14. Etapas del programa en ensamblador del sonar

En la Figura 15 se observa la señal que ofrece el sonar a su salida.



5.1.4.3. Comunicación con el microcontrolador

Finalmente, falta por analizar cómo se realizará la comunicación serie del microcontrolador ATmega8 con los diferentes módulos de comunicación. Para ello utilizaremos el módulo USART del ATmega8, el cual permite la transmisión/recepción serie síncrona/asíncrona con características como:

- Full dúplex: Tiene registros de transmisión y recepción independientes que le permiten enviar y recibir al mismo tiempo.
- Comunicación con múltiples baudrate's y posibilidad de comunicación a doble velocidad.
- Soporta diferentes formatos de protocolo: De 5 hasta 9 bits, con paridad o no y con 1 o 2 bits de stop.
- Detección de errores.
- Posibilidad de utilizar la interrupción como medio para mejorar la transmisión/recepción.

Antes de realizar la comunicación serie primero se ha de inicializar la USART, donde se establece el baud rate, el número de bits, paridad, bits de stop y habilitar la transmisión y recepción. Luego simplemente, se actúa sobre el PUERTOD.0 para la recepción y el PUERTOD.1 para la transmisión, donde a través de un bucle se observa el estado del buffer y se recibe el dato o se envía.

5.2 Interfaz gráfica

La interfaz gráfica seleccionada para interactuar con el usuario ha sido Microsoft Visual Basic, un lenguaje de programación orientado a objetos, el cual permite crear un entorno visual muy sencillo de interpretar por el usuario y ofrece la posibilidad de comunicación serie, la cual se detalla a continuación. (En el siguiente vínculo [2], aparece una aplicación Java que nos hace una demostración de la comunicación serie a través de Microsoft Visual Basic).

5.2.1. Microsoft Visual Basic

El objeto que nos permite la comunicación a través del puerto serie del ordenador es el MSComm, Figura 16. Dicho objeto posee diferentes características como la configuración del bit rate, los bits de paridad y los bits de stop, pero cabe destacar las dos funciones principales que utilizaremos a lo largo de toda la aplicación: *Output* e *Input*. La función *Output* permite enviar un dato de 8 bits y la función *Input* lee un dato de 8 bits del buffer.

En el sistema software, veremos como se hace servir el módulo MSComm y en que consiste la aplicación en Visual Basic para el control remoto de los periféricos.



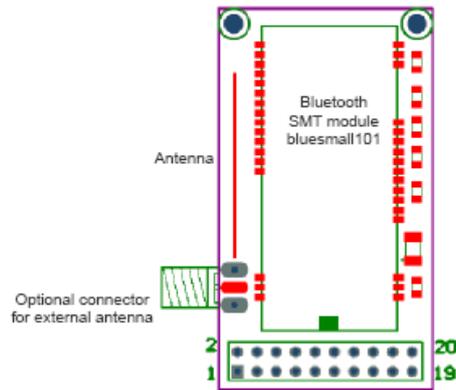
Figura 16. Icono MSComm

5.3 Comunicación vía Bluetooth

A continuación explicaremos la implementación del sistema Bluetooth, comenzando por el módulo de comunicación y analizando las posibilidades que nos ofrece para crear la red del prototipo. Continuaremos con el conexionado del prototipo, el funcionamiento del sistema, así como los resultados extraídos y ampliación del proyecto.

5.3.1. Estudio del módulo Bluemore200

En este apartado vemos en la Figura 17 la distribución de los pines del módulo de comunicación Bluetooth y el pinout de éste.



El pinout del módulo es:

Pin	Name	In Out	Description
1	GND	In	Ground
2	UART_CTS	In	UART CTS +3.3V or +5V TTL
3	+5V	In	+5V (do not use if powered by Pin 5)
4	UART_TX	Out	Output UART TX +3.3V
5	+3.3V	In	+3.3V (or regulated output if there's power supply on Pin 3)
6	UART_RTS	Out	Output UART RTS +3.3V
7	MISO	Out	SPI programming MISO signal
8	UART_RX	In	Input UART RX +3.3V or 5V TTL
9	CSB	In	SPI programming CSB signal
10	UART_RTS_5	Out	Output UART RTS 5V TTL
11	CLK	In	SPI programming CLK signal
12	UART_TX_5	Out	Output UART TX 5V TTL
13	MOSI	In	SPI programming MOSI signal
14	PIO 0	In/Out	Digital I/O #0 (Firmware defined)
15	PIO 5	In/Out	Digital I/O #5 (Firmware defined)
16	PIO 7	In/Out	Digital I/O #7 (Firmware defined) *
17	RS232_RTS	Out	Output RS232 RTS
18	RS232_TX	Out	Output RS232 TX
19	RS232_CTS	In	Input RS232 CTS
20	RS232_RX	In	Input RS232 RX

Figura 17. Esquema general del conexionado del Bluemore200

Es interesante destacar las siguientes características:

- Posible alimentación a +3.3Vdc y a +5Vdc.
- Permite trabajar con niveles de alimentación TTL, como con niveles RS232.
- El pin 16 si se conecta a un LED, éste parpadeará hasta que el módulo establezca comunicación con algún dispositivo.
- Uniendo los pines 17 y 19 eliminamos el protocolo *handshake* (interrupción por encuesta).
- Por último, señalar que el RS232_TX especifica que es del tipo Out, pero en nuestro caso se utiliza como In, lo cual se ha de tener en cuenta para el

conexionado. Lo mismo ocurre con el RS232_RX, del cual se especifica del tipo In, pero en nuestro caso es Out.

5.3.2. Análisis de la comunicación

El módulo Bluetooth Bluemore200 tiene dos modos de funcionamiento:

- Master: El módulo realiza una conexión punto a punto con otro módulo que actúa como Slave.
- Slave: Espera a que un Master acceda a él y se cree una conexión punto a punto. En este caso el Master puede ser otro Bluemore200 o bien el propio ordenador.

En nuestro prototipo utilizamos el segundo modo de funcionamiento. Como se ha comentado anteriormente, la idea es controlar de forma remota el servomotor y el sonar, donde cada uno de ellos está gobernado por un microcontrolador ATmega8. Por lo tanto, el sistema está formado por un módulo Bluemore200 unido a cada uno de los dispositivos, donde dichos módulos actúan como Slaves, siendo el propio ordenador el que actúa como Master.

La forma en la que el ordenador detecta cada uno de los módulos Bluemore200 con los que se quiere realizar una conexión punto a punto, se realiza a través de un *USB Dongle*, donde el fabricante facilita un software que permite detectar los dispositivos Bluetooth que nos rodean y acceder a cada uno de ellos, creando un Puerto Serie Virtual. En la Figura 18 se observa el entorno de dicha aplicación y en la Figura 19 vemos las características de una de las conexiones punto a punto establecidas.

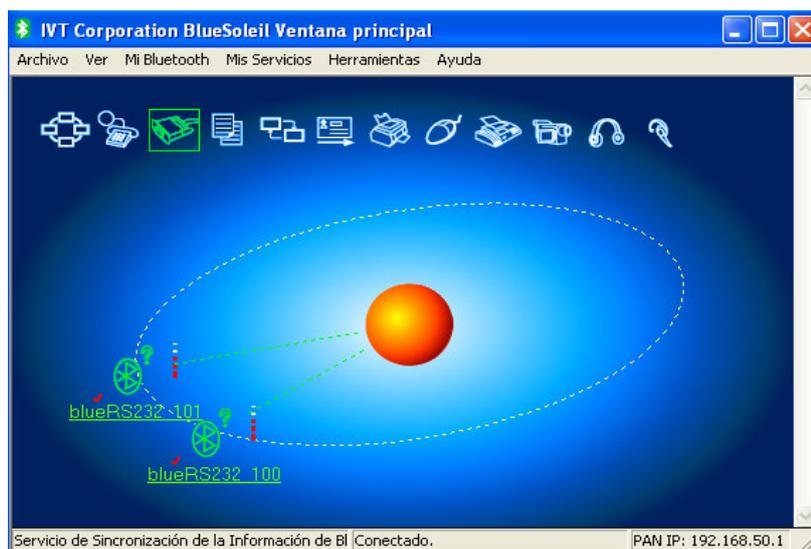


Figura 18. Entorno de la aplicación del *USB Dongle*



Figura 19. Características de la conexión punto a punto

Cabe destacar, que en la página oficial de Eikon, donde se encuentra toda la documentación [3], podemos descargar el software de configuración de los módulos Bluemore200. Dicho software nos permite configurar el modo de trabajo del módulo, la velocidad de transmisión, los bits de paridad, los bits de stop e incluso introducir una contraseña para establecer una conexión con él. Para poder realizar dicha configuración, se utiliza la placa de la Figura 20, donde a través de un cable serie hembra-hembra cruzado, se conecta a uno de los puertos serie del ordenador y establecemos una conexión. En la Figura 21 vemos los diferentes parámetros que podemos configurar del Bluemore200.



Figura 20. Placa de configuración del Bluemore200

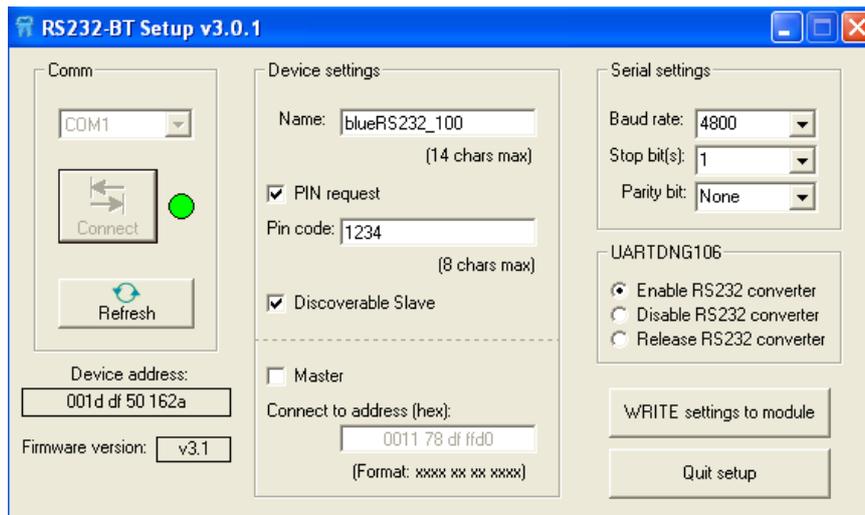


Figura 21. Software de configuración del Bluemore200

5.3.3. Conexionado

En la Figura 22 se observa a modo de bloques el conexionado del sistema Bluetooth.

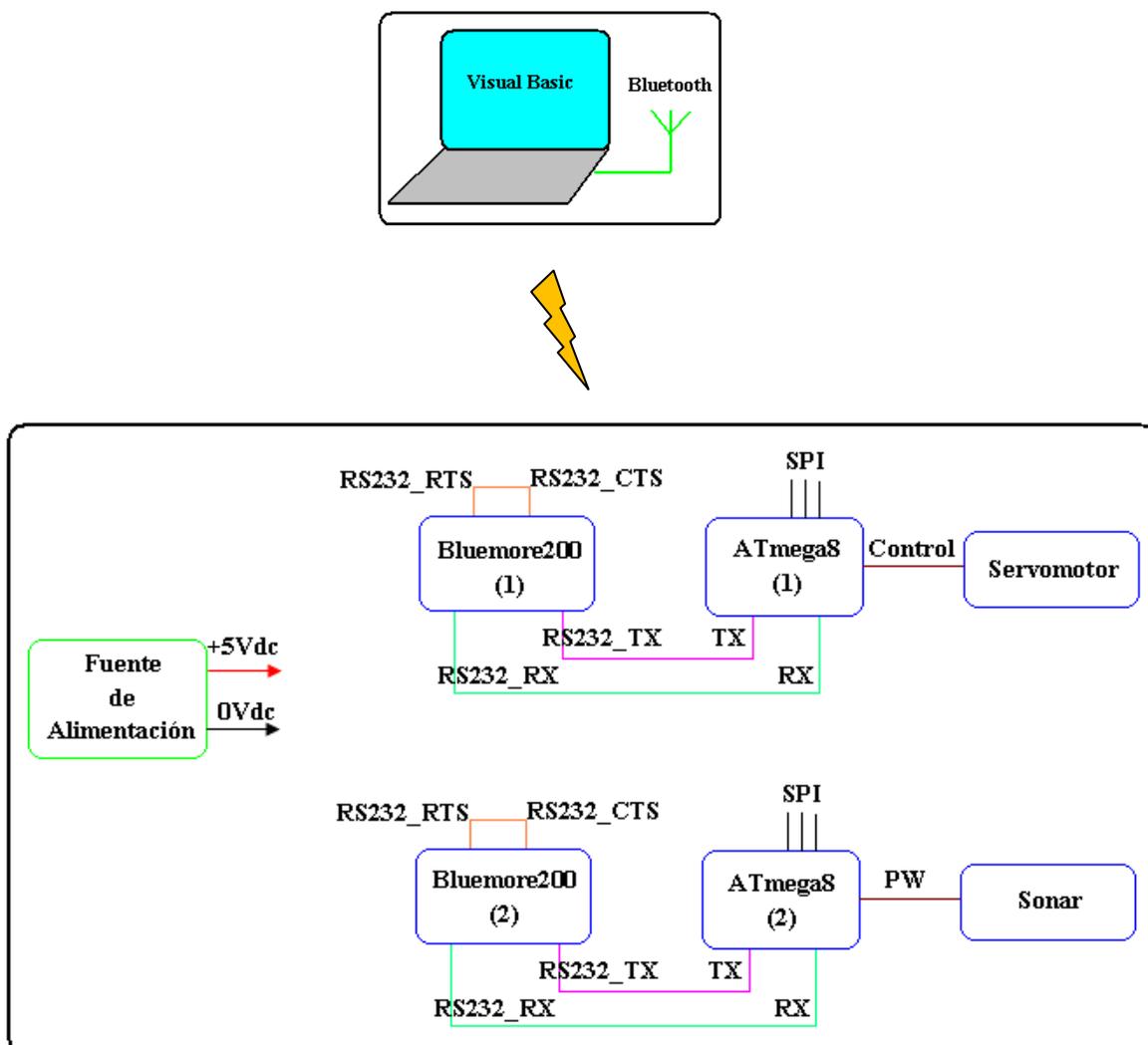


Figura 22. Conexionado del sistema Bluetooth

5.3.4. Funcionamiento

En la Figura 18 se observa la placa del prototipo del sistema Bluetooth. En ella se observa dos módulos Bluemore200 conectados a su respectivo microcontrolador ATmega8 y estos al servomotor y al sonar.

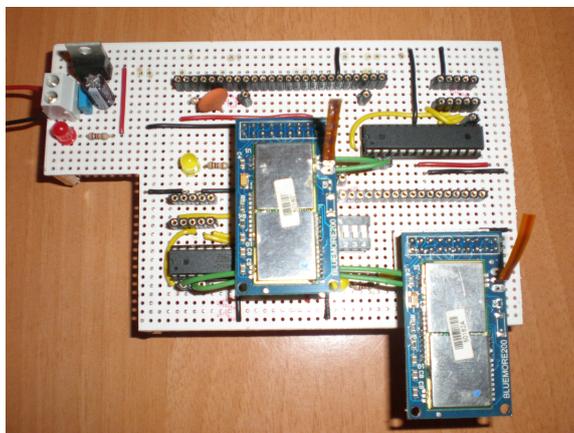


Figura 23. Prototipo del sistema Bluetooth

Entonces el funcionamiento es el siguiente:

- La comunicación del módulo Bluemore200 con el ATmega8 se realiza a través del módulo de comunicación USART de ambos.
- En los módulos Bluetooth se elimina el protocolo *handshake*, ya que no lo utilizamos en el PC.
- Se carga el programa que controla el servomotor y el sonar en los microcontroladores, pero se añade la particularidad de la comunicación serie, de tal forma que dicho código no se ejecutará hasta que no reciba un determinado dato (valor hexadecimal del 'espacio' del teclado = 0x03), y lo mismo para detenerlo (valor hexadecimal del '0' del teclado = 0x06). Los valores hexadecimales de los códigos de esta implementación, como los de RF, no se han elegido por ningún motivo en particular, ya que podrían haber servido otros valores. En la siguiente dirección [5] vemos la correspondencia de símbolos con su valor hexadecimal.
- Previamente hemos detectado los dos módulos Bluemore200 con el *USB Dongle* y hemos creado los puertos series virtuales.
- El resto consiste en programar en Visual Basic con el componente MSComm y darle un formato a los datos recibidos. El resultado es el que se observa en la Figura 24.

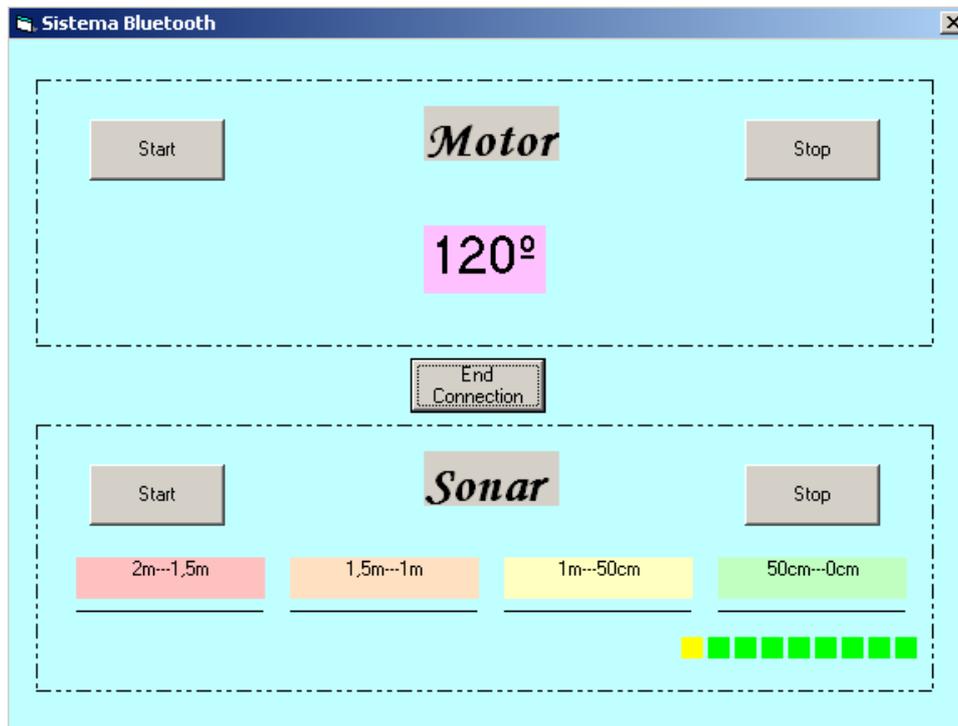


Figura 24. Aplicación Bluetooth

5.3.5. Resultados

En este apartado detallaremos diferentes aspectos o conclusiones que se extraen del sistema Bluetooth:

- La comunicación se realiza con una tasa de 4800 bits por segundo, debido a que a una velocidad mayor por ejemplo de 9600 bits por segundo, el microcontrolador introduce mucho error por bit transmitido. Por lo tanto, es el ATmega8 el que nos marca la barrera de la tasa de transmisión.
- Debido a aspectos de la programación en Visual Basic, como es el hecho de utilizar unos Timers para el envío constante de los datos de Start del programa en el ATmega8 y los datos de Stop, se pierde precisión a la hora de recibir las posiciones del servomotor, como las distancias detectadas por el sonar.
- Destacar que la red creada es del tipo punto a punto (comunicación directa entre los periféricos y el PC), no siendo posible la comunicación entre los periféricos, ya que el módulo Bluemore200 no permite esta función, como se explicó en el apartado de análisis de la comunicación.

5.3.6. Trabajo futuro

Una posible ampliación para esta implementación es utilizar módulos Bluetooth que permitiesen crear una piconet, de este modo existiría una comunicación entre los

periféricos-robots. Por lo tanto, el sistema a crear sería parecido al establecido en nuestro prototipo, pero en este caso todos los módulos actuarían como masters, lo que conllevaría incluir en la aplicación tantos componentes MSComm como dispositivos de comunicación hubiese, y a la vez a través del mismo ATmega8 gobernar la comunicación entre los distintos periféricos.

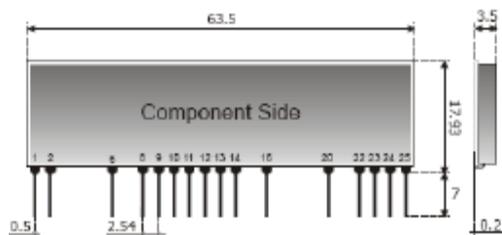
Es decir, la idea es ser capaz de activar y desactivar el servomotor, por ejemplo, recibir la posición de éste, pero a la vez, si el motor llega a un determinado ángulo que fuese éste el que activase el sonar. A partir de este punto, no existe límite de expansión del proyecto, simplemente pensar en diferentes aplicaciones.

5.4 Comunicación vía Radiofrecuencia

Una vez concluida la implementación en Bluetooth, comenzamos con el sistema de Radiofrecuencia, donde ahora el módulo a analizar es distinto, como las redes de comunicación que podemos crear. Posteriormente hablaremos del prototipo que se ha creado y veremos que resultados se extraen, así como una posible ampliación.

5.4.1. Estudio del módulo RTF-DATA-SAW

En este apartado vemos en la Figura 25 la distribución de los pines del módulo de comunicación RF y el pinout de éste.



El pinout del módulo es:

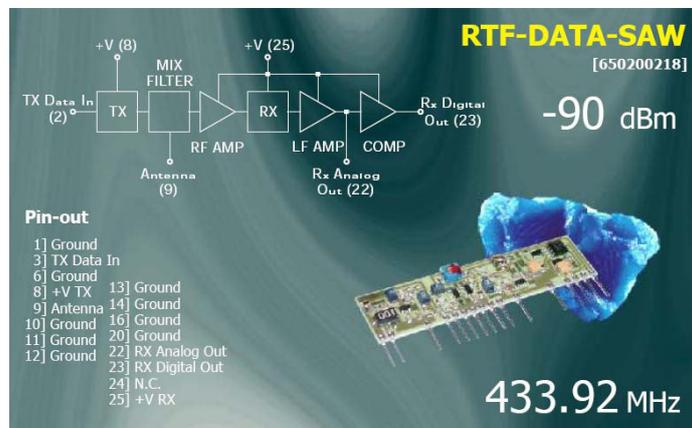


Figura 25. Esquema general del conexionado del RTF-DATA-SAW

Es interesante destacar las siguientes características:

- Alimentación de +5Vdc, en el pin 8 para permitir la transmisión y en el pin 25 para la recepción.
- Antena de aproximadamente 17 cm, conectada en el pin 9.
- El dato a enviar se transmite por el pin 2, donde colocamos una resistencia de 1 K Ω en serie con un condensador de 10 nF para eliminar interferencias, y por el pin 23 recibimos la señal. Cuidado porque si estamos transmitiendo, estaremos recibiendo lo que transmitimos, por lo tanto se han de establecer unos tiempos de actuación para el transmisor y el receptor.

Toda la información de este módulo, así como información sobre el protocolo RS232, la podemos encontrar en los enlaces bibliográficos [5] y [6].

5.4.2. Análisis de la comunicación

En nuestro prototipo requerimos de 3 módulos RTF-DATA-SAW. Dos de los módulos irán conectados a los Atmega8 que controlan el servomotor y el sonar. Dichos componentes se encargarán de establecer una comunicación punto a punto, con el RTF-DATA-SAW que irá conectado al puerto serie de nuestro ordenador. Comentar que hemos optado por una comunicación punto a punto, debido a que los módulos se basan en el protocolo de comunicación RS232, donde tenemos un bit de start ('0'), 8 bits de datos y un bits de stop ('1') como se observa en la Figura 26. Entonces la idea para realizar la comunicación es jugar con estos 8 bits de datos, de tal forma que el módulo del ordenador sea capaz de saber en que momento mandamos o recibimos señal del servomotor, y lo mismo ocurre para el sonar.

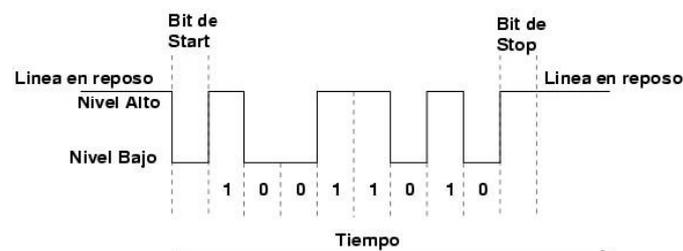


Figura 26. Protocolo RS232

Otro aspecto a tener en cuenta, es la comunicación entre el microcontrolador-RTF-DATA-SAW con el puerto serie del ordenador. Empezaremos hablando del cable serie cruzado hembra-hembra, el cual dispone de un conector DB9, el cual se muestra la disposición de los pines y su información en la Figura 27.

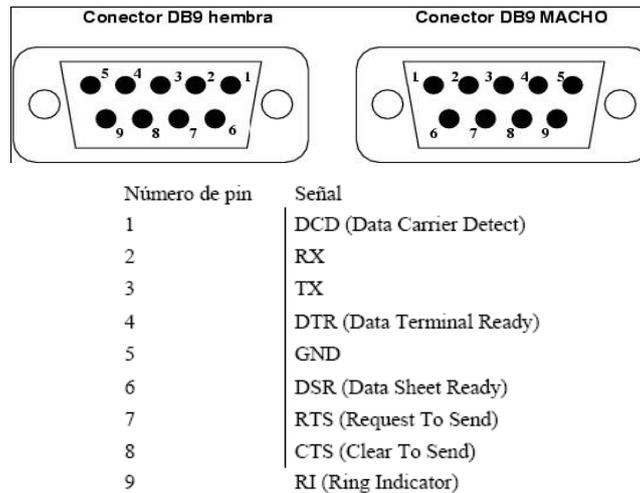


Figura 27. Conector DB9

En nuestro caso simplemente utilizaremos el bit 2 para recibir una señal, el bit 3 para transmitir y el bit 5 que se conecta a masa. No haremos servir los bits 7 y 8 debido a que no establecemos ningún protocolo de *handshake*.

Ahora nos queda resolver el problema de adaptación de niveles de señal del ordenador con el microcontrolador. El microcontrolador reconoce niveles TTL, sin embargo el ordenador necesita niveles RS232, de ± 12 Vdc. En el mercado hay muchos circuitos integrados que permiten la conversión entre niveles TTL y niveles RS232, Entre ellos destaca el Transceptor MAX232, fabricado por Dallas Semiconductor-MAXIM. El MAX232 convierte los niveles RS232 (cerca de +12 y -12 V) a voltajes TTL (0 a +5 V) y viceversa sin requerir nada más que una fuente de +5V. El chip contiene dos drivers TTL \rightarrow RS232 y dos RS232 \rightarrow TTL. Necesita cuatro condensadores externos de unos pocos microfaradios para generar el voltaje RS232 internamente tal como se muestra en la Figura 28.

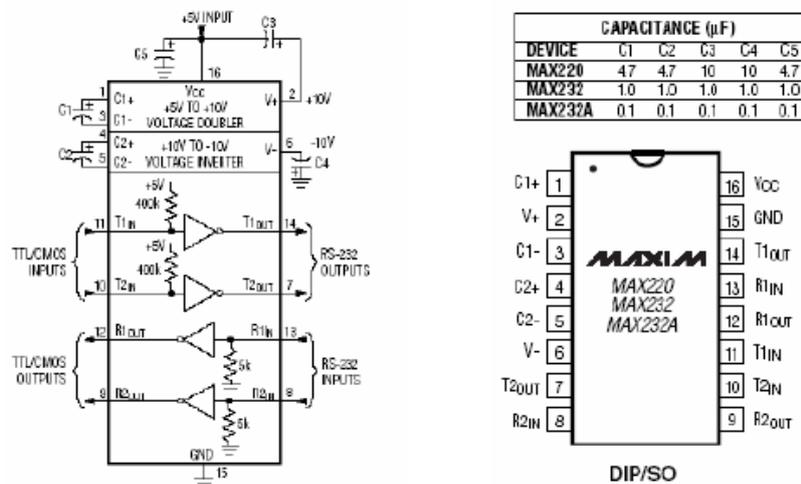


Figura 28. MAX232

5.4.3. Conexionado

En la Figura 29 se observa a modo de bloques el conexionado del sistema Bluetooth.

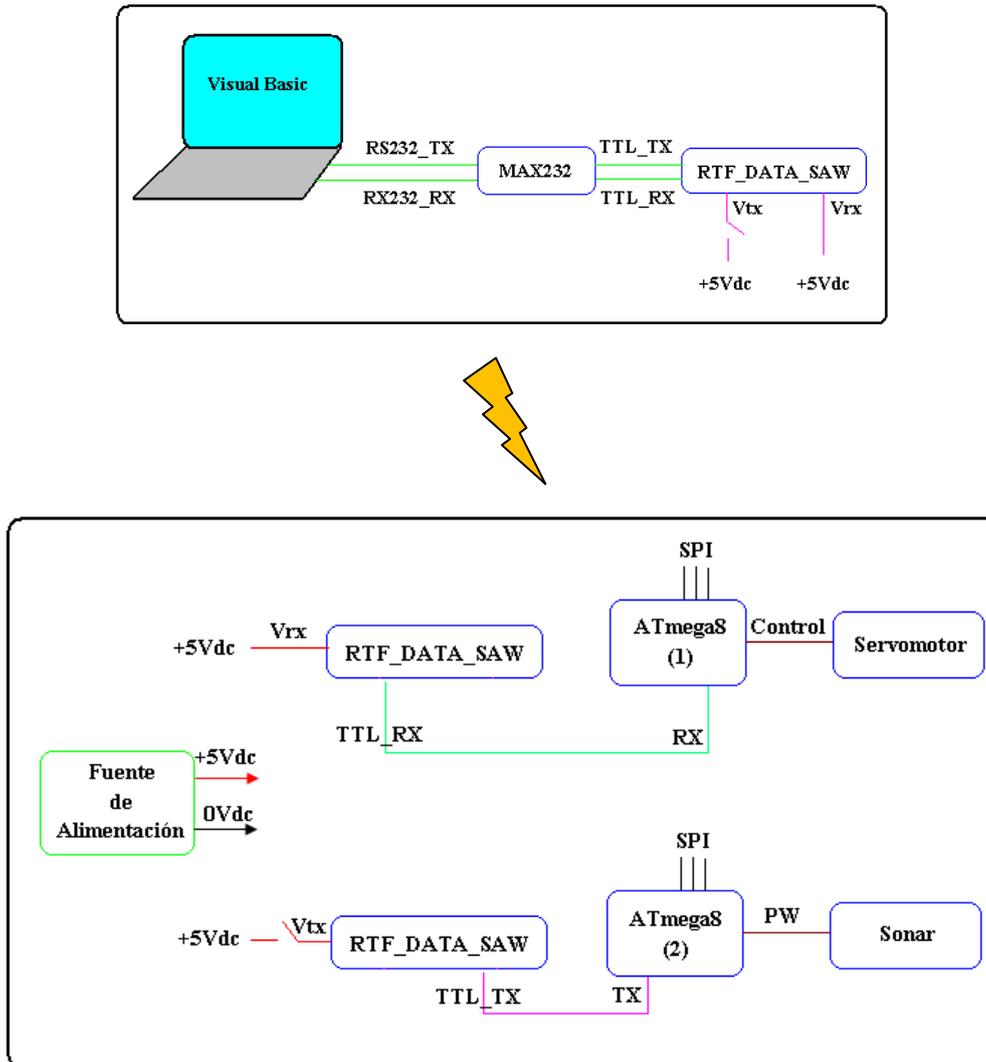


Figura 29. Conexionado del sistema RF

5.4.4. Funcionamiento

En la Figura 30 vemos las placas del prototipo del sistema de RF. Por un lado tenemos el módulo RTF-DATA-SAW que se conectará a la estación base, el cual permitirá una comunicación de transmisión y recepción de datos, y además se observa otra placa conectada con el cable serie, que es la encargada de generar los niveles TTL-RS232 y viceversa. El otro conjunto está formado por dos módulos RTF-DATA-SAW conectados a su respectivo microcontrolador ATmega8 y estos al servomotor y al sonar.

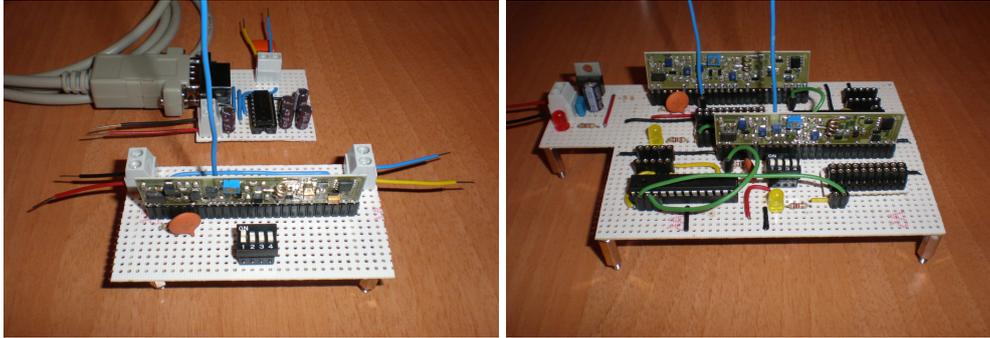


Figura 30. Prototipo del sistema de RF

Al haber dos partes claramente diferenciadas trataremos por separado su funcionamiento.

El funcionamiento de la estación base es el siguiente:

- Se encarga de mandar los datos hexadecimales de los caracteres B (0x42) y H (0x48) que le envía la aplicación software a través del puerto serie, y permite recibir los datos que nos transmite el sonar de la distancia de los diferentes objetos localizados. Es por ello, que exista la placa de conversión de voltajes (MAX232) debido a que existe una TX-RX en ambos sentidos.
- Remarcar que la transmisión y recepción de datos por parte de la estación base se gobierna con el interruptor negro que se observa en ella. En el apartado de resultados analizaremos el por qué de este interruptor.

El funcionamiento de los módulos conectados a los periféricos es el siguiente:

- Existe un módulo RFT-DATA-SAW por cada uno de los ATmega8 que realizan el control del servomotor y el sonar respectivamente.
- En los módulos RF se elimina el protocolo *handshake*, ya que no lo utilizamos en el PC.
- En este prototipo controlaremos la activación y desactivación del movimiento del servomotor, pero no recibiremos los datos que indican su posición. Por lo tanto, al microcontrolador que controla el servomotor se cargará el programa de gestión de éste con la particularidad de los datos hexadecimales que transmite la estación base, esto quiere decir que el módulo conectado a esta unión sólo actuará como receptor. Por otro lado, del sonar no controlaremos ni el arranque ni la parada, simplemente cargaremos en el microcontrolador un programa que transmita a través del módulo USART los datos de posición del sonar, ello

conllea que el módulo de comunicación asociado solo trabajará como transmisor. En este RTF-DATA-SAW existe otro interruptor que gobierna la alimentación de la transmisión de éste.

- El resto consiste en utilizar el código de la implementación en Bluetooth, cambiando los valores a transmitir, y el resultado es el que se observa en la Figura 31.

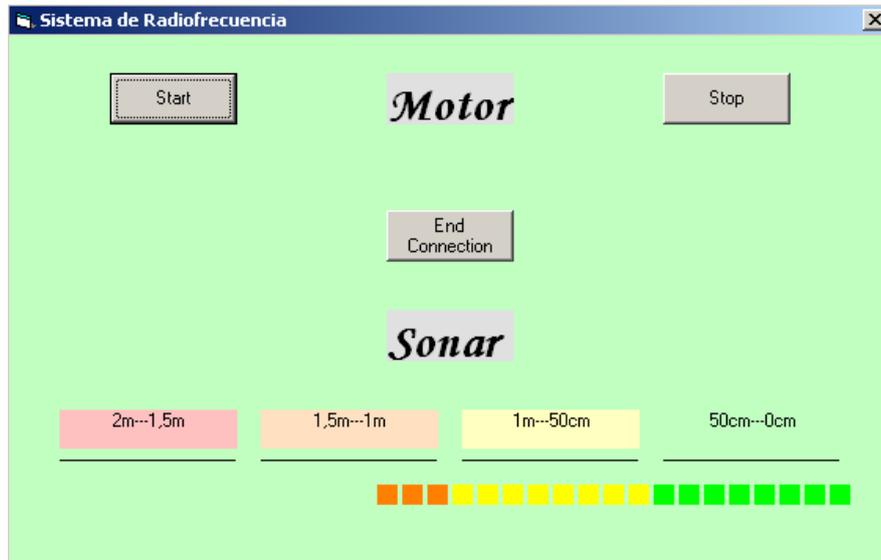


Figura 31. Aplicación de RF

5.4.5. Resultados

En este apartado detallaremos diferentes aspectos o conclusiones que se extraen del sistema de RF:

- La comunicación se realiza con una tasa de 2400 bits por segundo, debido a que los módulos RTF-DATA-SAW permiten como máximo que la transmisión se realice a esta velocidad.
- Se han realizado diferentes pruebas con los módulos de RF, y el resultado es que si utilizamos en el prototipo simplemente dos módulos de comunicación, ambos pueden trabajar en TX y RX. Por lo tanto, si hubiésemos querido, por ejemplo, controlar el movimiento del motor y recibir su posición no habría habido ningún problema. Pero al no ser esa la idea del proyecto, se optó por realizar un control sobre ambos periféricos, aunque ello conllevara perder algunas opciones. Es por ello, que existe un interruptor en la estación base y otro en el RTF-DATA-SAW del sonar. Si en el módulo de la estación base está alimentada la transmisión, el propio módulo deshabilita la recepción, al haber más de dos módulos, por lo

tanto, hemos de controlar este factor para poder controlar el servomotor, y a la vez, recibir los datos del sonar.

- Otro aspecto es que los módulos de comunicación funcionan basándose en el protocolo RS232, que se explicó en apartados anteriores, lo cual implica trabajar con 8 bits, donde habría que incluir dirección, datos y paridad. En nuestro caso, hacemos servir los 8 bits como datos para indicar el arranque y la parada, y el sonar utiliza los 8 bits como datos para indicar la distancia.
- En este prototipo, existe más exactitud en los datos que ofrece el sonar, porque constantemente se reciben los datos, no existen unos Timers como en el caso Bluetooth que nos producían un error en el sistema.

5.4.6. Trabajo futuro

La ampliación que se puede realizar utilizando los mismos módulos RTF-DATA-SAW, consiste en ver como solucionar los problemas ocasionados al utilizar tres módulos, y crear un protocolo con 8 bits, donde tendríamos unos bits para indicar el dispositivo (3 bits), otro de paridad (1 bit) y transmitir por un lado los 4 bits más significativo de los datos y por otro lado los 4 bits menos significativos.

Si por el contrario, nos olvidamos de módulo RTF-DATA-SAW y buscamos en el mercado cualquier otro que no se base en el protocolo RS232, disponemos de una gran cantidad de bits para crear nuestro protocolo, lo que nos permitiría en una misma trama localizar el bit de start, el dispositivo, los datos, la paridad y el bit de stop. A mayor cantidad de bits para identificar los dispositivos, mayor será el número de dispositivos sobre los que podremos actuar. De esta forma se creará una red donde todos se pueden comunicar entre ellos.

6. Conclusiones

El estudio de las diferentes tecnologías inalámbricas existentes en el mercado, nos permite realizar una comparativa de las diferentes prestaciones de cada una de ellas. De ahí que el proyecto se realice mediante tecnología Bluetooth y vía Radiofrecuencia, debido a la mayor versatilidad de ambas con respecto a las otras tecnologías inalámbricas y la gran variedad de productos que presentan.

Otro aspecto a destacar, es el hardware de cada uno de los medios inalámbricos. En el mercado existen multitud de alternativas en función de nuestros requerimientos, lo cual nos ha llevado a seleccionar dos componentes aptos para nuestros prototipos teniendo en cuenta: distancia, velocidad, precio, prestaciones, etc. De una banda tenemos el Bluemore200, un módulo Bluetooth económico para sus prestaciones y fácil de conectar y, el RTF-DATA-SAW, módulo de Radiofrecuencia con una fácil operatividad y muy económico. Y lo más importante nos permite desarrollar los prototipos de nuestro proyecto.

Por lo tanto, una vez analizadas las implementaciones Bluetooth y RF y observar los resultados obtenidos, las conclusiones que se extraen son las siguientes:

- El sistema Bluetooth presenta pocas interferencias, y las que aparecen son debidas al movimiento del servomotor. Los módulos se conectan perfectamente con el PC y en la comunicación Bluemore200-ATmega8 no existe ningún problema, siempre y cuando se tenga en cuenta que se ha de eliminar el protocolo *handshake*.
- Por el contrario, el sistema de Radiofrecuencia presenta muchas interferencias, ya que cualquier equipo electrónico conectado alrededor ya introduce errores en la implementación. La conversión de niveles TTL-RS232 y viceversa permite un entendimiento perfecto entre PC y microcontrolador.
- En ambos sistemas no ha sido posible establecer una red en la que todos se pudiesen comunicar entre ellos. En el caso Bluetooth, el objetivo era crear una red piconet, pero después de contactar con Eikon, se nos informó que dicho módulo sólo podía establecer una conexión punto a punto, por lo que se optó por utilizar un USB *Dongle* para controlar todos los periféricos con conexiones punto a punto. En el caso de RF, el problema es más de tipo electrónico, cuando el prototipo está formado por dos módulos la comunicación TX/RX funciona sin

problemas, pero al añadir otro módulo de comunicación falla la transmisión o bien la recepción del RTF-DATA-SAW de la estación base.

Por último, respecto a la aplicación del proyecto, remarcar que el abanico de posibilidades es inmenso, sólo depende de la finalidad del prototipo. Lo que nos permite concluir que a pesar de existir diferentes inconvenientes para realizar una comunicación entre todos los dispositivos, el prototipo Bluetooth, ya no el de RF debido a las interferencias, puede ser de gran utilidad en la vida diaria, por ejemplo, en el sistema de vigilancia remota de un establecimiento. Ello nos permitiría controlar dos periféricos, un servomotor para indicar el punto de visibilidad y una cámara para visualizar el entorno, dando al usuario diferentes opciones dependiendo de la aplicación software.

7. Bibliografía

- [1] Web Can Nic. “Característica, Funcionamiento y Programas a modo de Ejemplo del microcontrolador ATmega8”. Acceso on-line disponible en (Junio 2009): <http://www.cannic.uab.es/docencia/cannicframedocencia.htm>
- [2] Yahoo Geocities. “Aplicación Java de la Comunicación Serie a través de Microsoft Visual Basic”. Acceso on-line disponible en (Junio 2009): http://www.geocities.com/ejemplo_mscomm/index.html
- [3] Página oficial de Eikon. “Información y Software sobre el módulo de comunicación Bluetooth Bluemore200”. Acceso on-line disponible en (Junio 2009): <http://www.eikonsite.it/>
- [4] Códigos ASCII. “Tabla de códigos ASCII”. Acceso on-line disponible en (Junio 2009): <http://www.ascii.cl/es/>
- [5] Página oficial de Aurel. “Información sobre el módulo de comunicación de Radiofrecuencia RTF-DATA-SAW”. Acceso on-line disponible en (Junio 2009): <http://www.aurel.it/>
- [6] Abacom Technologies. “Datasheet del RTF-DATA-SAW”. Acceso on-line disponible en (Junio 2009): http://www.abacom-tech.com/data_sheets/RTF_L-DATA-SAWuser.pdf