CAPÍTULO IV

Paquetes y Rutinas de Transmisión y Recepción del Autómata

En este capítulo hablaremos de los enlaces físicos de comunicación SCO y ACL siendo este último es que ocupará el Autómata en la conexión inalámbrica con la terminal. Se explicará de una forma mas detallada cuales son las partes principales del paquete así como los tipos de enlace que es para el proyecto. También se explicará la rutina de transmisión y recepción que se llevará a cabo.

4.1 Partes de un paquete

4.1.1 Código de Acceso

Cada paquete comienza con un código de acceso. Dicho código es de 72 bits si es seguido de un paquete de datos de cabecera que enviará la terminal al Autómata. El código de acceso identifica los paquetes que llegarán.

El código de accesos también es usado para procedimiento de pagying e inquiry. En este caso el código de acceso por si mismo, es utilizado como mensaje de señalamiento y ni un encabezado ni un payload esta presente.

4.1.2 Encabezado de paquete

El encabezado contiene información de control de enlace y consiste de 6 campos. Ver figura 4.1.

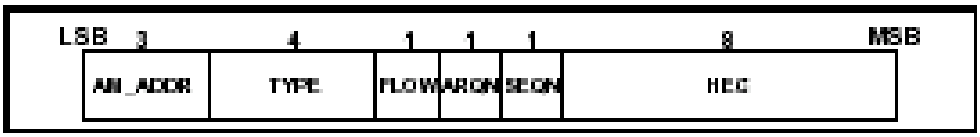


Figura 4.1 "Encabezado"

4.1.2.1 AM_ADDR

Representa una dirección de un miembro y es usado para distinguir entre los miembros activos que participan en la piconet. En un piconet uno o mas esclavos son conectados a un mismo maestro, para identificar cada esclavo por separado, a cada esclavo se le asigna temporalmente una dirección de 3 bits que se usa cuando esta activo. Los paquetes que se intercambian entre el maestro y el esclavo contienen el AM_DDR del esclavo, es decir, la dirección del esclavo se usa en ambos paquetes (maestro esclavo- esclavo maestro). Los esclavos cuando salen de la piconet dejan de tener esa dirección y cuando vuelven a entrar se les asigna otro AM_DDR

4.1.2.2 TYPE

Se pueden distinguir entre 16 tipos de paquetes. El código Type de 4bits especifica que tipo de paquete es usado, es importante notar que la interpretación del código type depende del lazo físico asociado con el paquete. Primero se debe determinar si el paquete se envía en un enlace SCO o ACL, posteriormente se puede determinar que tipo de paquete SCO o ACL se recibió. El código de type también nos indica cuantas ranuras del paquete actual se van a ocupar.

4.1.2.3 Flow

Este bit es usado para el control de flujo de paquetes sobre el enlace ACL. Cuando el buffer RX para el enlace ACL está lleno y no ha sido vaciado, una indicación de paro (Flow = 0) se regresa para detener la transmisión de datos temporalmente. Se debe notar que la señal de paro solo le concierne a paquetes ACL.

Los paquetes que incluyen información de control de enlace o paquetes SCO todavía pueden ser recibidos. Cuando el buffer RX se vacíe una indicación de continuar se regresa (flow =1). Cuando no se recibe ningún paquete o el encabezado recibido es un error, una señal de continuar se asume implícitamente. En este caso el esclavo puede recibir un paquete nuevo con CRC aunque su buffer de RX todavía no se vacíe. El esclavo debe regresar un NAK en respuesta a este paquete, aun cuando el paquete el CRC pasó.

4.1.2.4 ARQN

EL indicador ARQN de reconocimiento de un bit se usa para informar a la fuente de una transferencia exitosa de datos payload con CRC, y puede ser reconocimiento positivo ACK o negativo NAK. Si la recepción fue exitosa ACK (ARQN =1) si fue fallida un NAK se regresa (ARQN = 0). Cuando no se recibe un reconocimiento del mensaje de regreso, el NAK se asume implícitamente. NAK también es la información de regreso por default.

4.1.2.5 SEQN

El bit SEQN provee un esquema de numeración secuencial para ordenar la cadena de paquetes de datos. Para cada nuevo paquete transmitido que contienen datos CRC, el bit SEQN se invierte. Esto se requiere para filtrar las retransmisiones. Si una retransmisión ocurre debido a una falla ACK el destino recibe el paquete dos veces.

4.1.2.6 HEC

Cada encabezado tiene una revisión de error de encabezado, para comprobar la integridad del encabezado. El hec consiste de una palabra de 8 bits. Antes de generar el hec el generador se inicializa con un valor de 8 bits. Para paquetes FHS enviados en estado de respuesta de page, se usa el esclavo de la parte alta de la dirección. Para paquetes FHS iniciados en respuesta inquiry, se utiliza la inicialización de chequeo por default. En todos los demás casos el UAP del dispositivo maestro es usado.

4.1.3 Formato Payload

En el payload dos campos se distinguen, el campo de voz y el campo de datos (para el Autómata solo se ocupará el de datos). Los paquetes ACL solamente tienen campos de datos y los paquetes SCO solamente tienen campo de voz.

Campo de Datos: Consiste de tres segmentos. Un encabezado payload, un cuerpo payload y posiblemente un código CRC.

4.1.3.1 Payload en la Transmisión de Datos al Autómata

Solamente los campos de datos como los que se van a utilizar en la conexión entre el Autómata y la terminal tiene encabezado. Dicho encabezado solamente es uno o dos bytes de longitud. Especifica el canal lógico, controla el flujo de canales y tienen un indicador de longitud del payload, aunque para esta aplicación es muy pequeño el Payload por que las rutas del autómata son muy cortas y no soportan muchas aplicaciones. El cuerpo del Payload incluye la información de la terminal y determina el uso efectivo del enlace. La longitud del cuerpo payload se indica en el encabezado.

4.2 Datos generados en la Terminal (Arreglo de Visual Basic)

La rutina de transmisión para cada enlace físico (ACL y SCO) se realiza por separado. Existe en la transmisión del enlace físico ACL solo un maestro (la terminal) para cada esclavo (el Autómata).

La primera parte trata de la modificación del programa realizado en la tesis pasada, en la cual los datos que iban a ser transmitidos se mandaban al puerto serial y posteriormente se mandaban al autómata por medio de la modulación en FM.

El software de trayectorias virtuales en lenguaje de alto nivel se encargará de los datos al autómata según la trayectoria planteada por medio de una interfaz gráfica al usuario.

El programa que fue realizado en dicho lenguaje fue el que se tuvo que modificar ya que los datos que se iban generando los mandaba al puerto serial para que posteriormente fueran transmitidos al Autómata vía Bluetooth. Se hizo un

arreglo en las instrucciones del programa de tal forma que se hiciera una archivo de texto con los datos que se generaban en Visual Basic. Una vez teniendo esto solo se transmite el archivo anteriormente mencionado de dos formas; encriptado o sin encriptar.

En las siguientes figuras se muestra como se dibuja la trayectoria de lazo abierto que el autómata va a seguir, junto con las coordenadas cartesianas de los puntos seleccionados y las coordenadas angulares.

En estas ventanas se puede observar datos tales como: Vectores dinámicos de control, parámetros de calibración, bytes de información que se enviará al autómata , puerto de comunicación que se va a ocupar así como el tipo de trayectoria y distancia que recorre entre punto y punto (velocidad del motor derecho, velocidad del motor izquierdo, tiempo de encendido,).

Se puede ver que el documento generado contiene los mismos datos que están en la interfaz gráfica de Visual Basic, los cuales serán transmitidos.

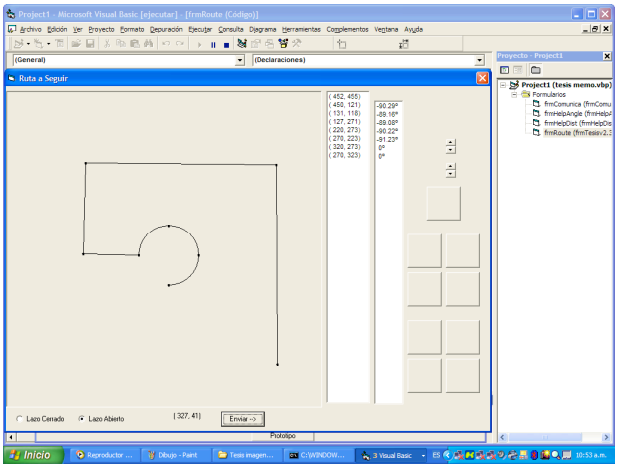


Figura 4.2 "Trazo de la ruta a seguir por el Autómata"

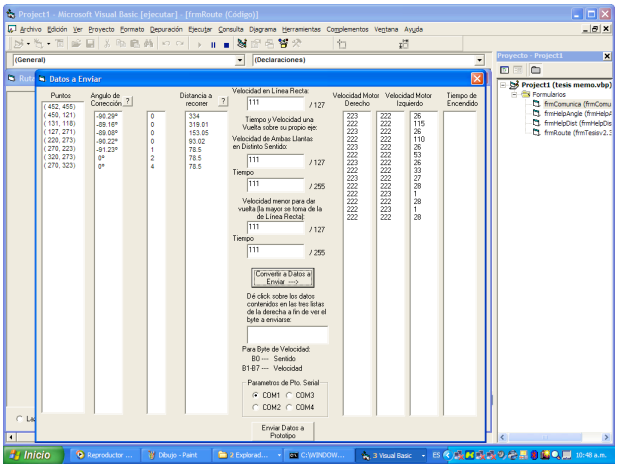


Figura 4.3 "Conversión de ruta en datos de Visual Basic"

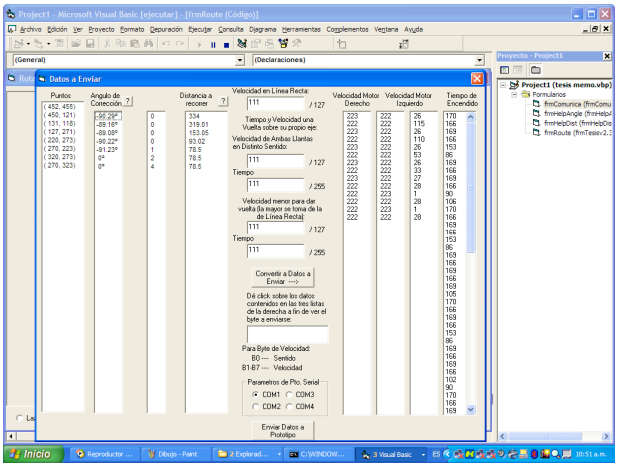


Figura 4.4 "Datos generados por Visual Basic para enviar"

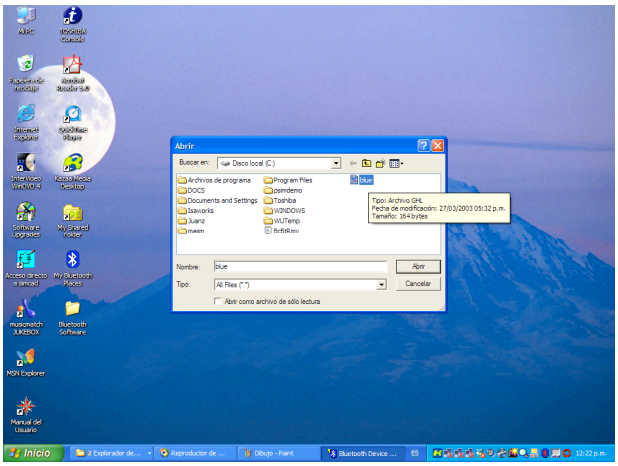


Figura 4.5 "Abriendo el archivo generado"

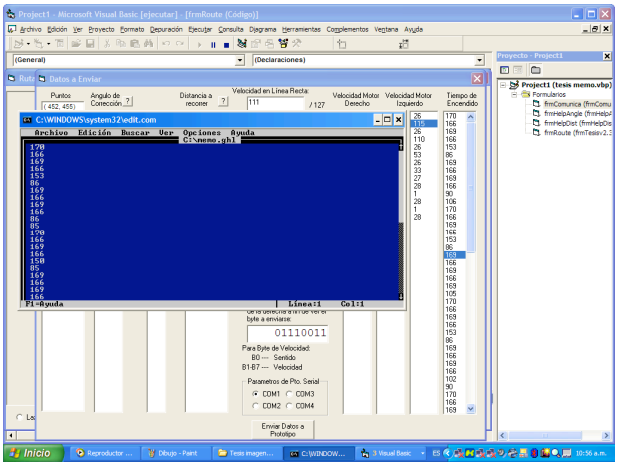


Figura 4.6 "Comparación de los datos de Visual Basic y del Archivo"

4.3 Archivo Transmitido al Autómata

Una vez instalado el software de Bluetooth se selecciona el archivo generado para enviar y posteriormente se manda al autómata por medio de Bluetooth.

Existen dos opciones para mandar este archivo. Una es seleccionar el software de Bluetooth para el envío y después buscar el archivo en la PC (fuera del programa de Visual Basic). La otra opción es seleccionar el archivo que se quiere mandar y darle click con el botón derecho para que despliegue el menú de opciones y posteriormente se selecciona el icono de enviar en donde aparecerá el logotipo de Bluetooth.

En el caso de que la computadora no detecte el hardware o no este bien instalado se desplegara un mensaje de error como en la figura.

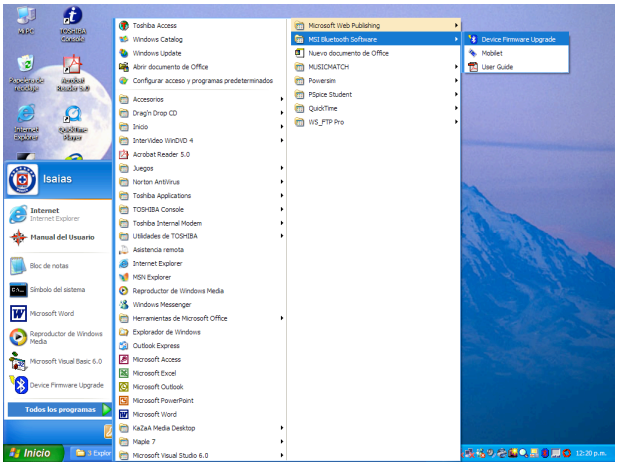


Figura 4.7 "Selección de Software de Bluetooth para el transmisión"

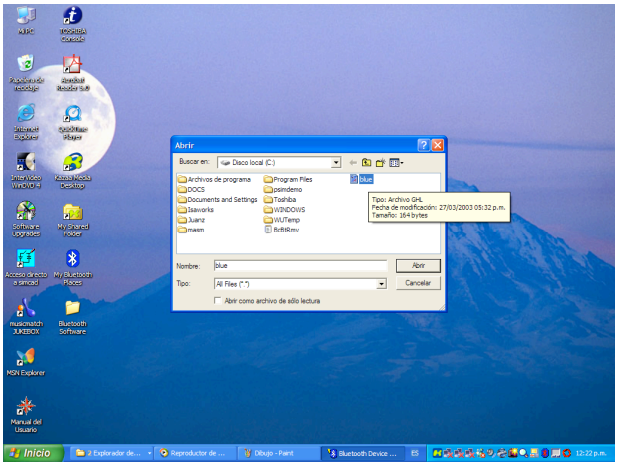


Figura 4.8 "Seleccionando el archivo generado para enviar"

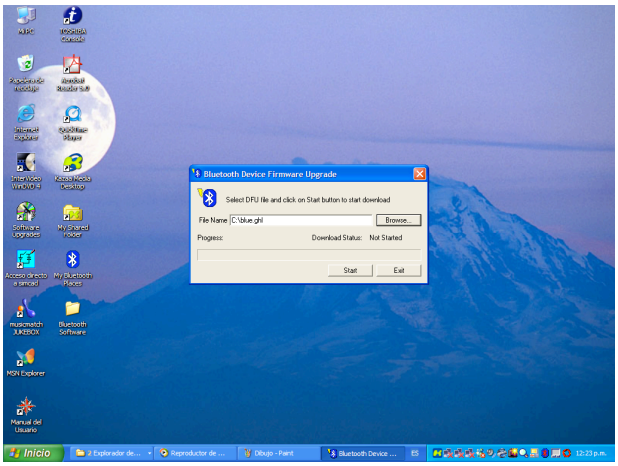


Figura 4.9 "Envío del archivo por Bluetooth"

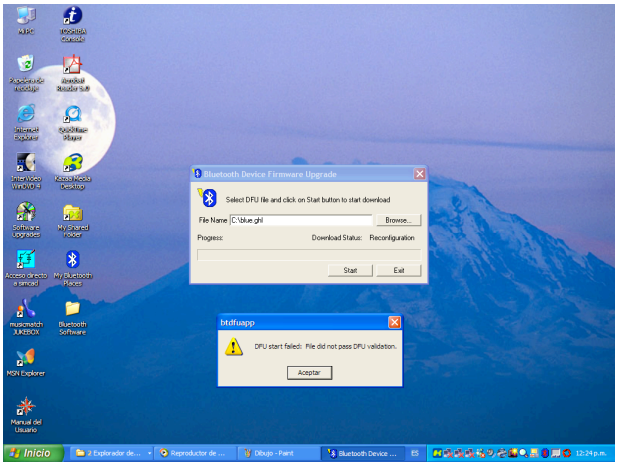
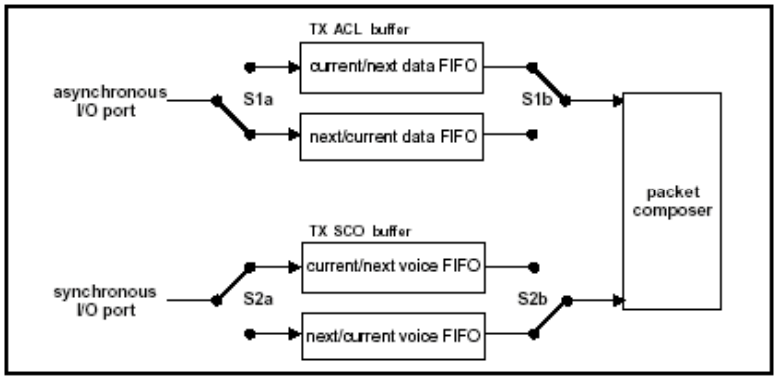


Figura 4.10 "Mensaje de error al no detectar el hardware instalado"

4.4 Rutina de Transmisión ACL de la terminal

Una vez que el usuario ha trazado la trayectoria que desea y que el programa genera el documento que se va a transmitir, el cual contiene todas las instrucciones para que el autómata las ejecute, la terminal va seguir la siguiente rutina para establecer conexión con el robot.

La rutina TX trabaja de la siguiente manera: el gestor de enlace Bluetooth (terminal) carga la información nueva en el registro donde esta señalando el switch S1a. Posteriormente el robot manda un comando flush al controlador de enlace Bluetooth, el cual forza al switch S1 a cambiar (ambos S1a y S1b están sincronizados). Cuando el payload necesita ser enviado, el editor de paquetes lee el registro actual y, dependiendo del tipo de paquete, construye un payload, el cual se añade al código de acceso del canal, al header y posteriormente son transmitidos.



En el paquete de respuesta del robot (el cual llega en el siguiente slot RX si se trata de una transmisión del maestro, o se puede transmitir después si se trata de una transmisión de algún esclavo) se reporta el resultado de la transmisión.

Mientras el gestor de enlace (la terminal) se mantenga cargando los registros con la nueva información, el controlador de enlace Bluetooth seguirá transmitiendo el paquete automáticamente. Si existiera un error en la transmisión el paquete se enviará nuevamente.

Si no se han cargado datos en el registro NEXT, durante la ultima transmisión, el editor de paquetes estará apuntando un registro vacío después de la última transmisión en la que se hizo un reconocimiento y el registro NEXT se vuelve el registro CURRENT.

El comando flush pude ser usado en caso de datos de tiempo limitado. En caso de un mal acoplamiento, la retransmisión será necesaria varias veces. En ciertas aplicaciones los datos son de tiempo limitado: si una carga útil se retransmite a toda la hora debido a errores del acoplamiento, puede llegar a ser obsoleta, y el sistema puede decidir entre continuar con datos mas recientes o seguir con los datos que no pueden ser transmitidos correctamente. Esto es logrado gracias al comando flush. Con dicho comando, el switch es forzado a cambiar y el

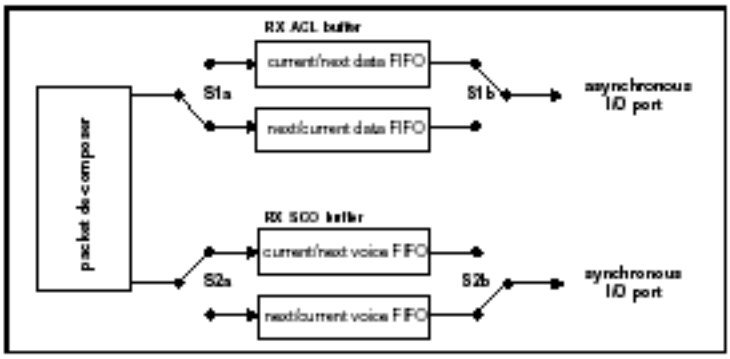
controlador de enlace es forzado a considerar el siguiente paquete de información e invalida el control del ACK

4.5 Rutina de Recepción del Autómata

Las rutinas de recepción son diferentes para los enlaces SCO y ACL. Sin embargo, en contraste con el buffer de transmisión ACL, en la recepción un solo buffers compartido por todos los esclavos.

El buffer ACL de RX está compuesto por dos registros FIFO. Un registro que puede ser accesado y cargado por el controlador de enlace de Bluetooth con el payload del ultima transmisión, y otro registro que pueda ser accesado por el gestor de enlace Bluetooth para el payload anterior.

El buffer SCO de RX también esta compuesto por dos registros FIFO. Un registro que es llenado con la información de voz recién llegada y un registro que puede ser leído por la unidad de procesamiento de voz.



El switch S1 cambia cada vez que el gestor de enlace lee el ultimo registro. Si el siguiente payload llega antes de que el registro RX se vacíe, una señal de paro debe ser incluida en el encabezado del siguiente paquete de TX que se

regresa. La indicación de paro se quita tan pronto como el registro de RX se vacíe. El campo SEQN se comprueba antes de que el nuevo payload ACL se Almacene en un registro ACL.

El switch S2 cambia cada Tsco. Si debido a errores en el header no llega ninguna información de la voz, el switch sigue cambiando. La unidad de procesamiento de voz tiene que procesar las señales de voz para completar las partes faltantes de dialogo.

4.6 Hardware Bluetooth para la Transmisión y Recepción

Para tener una comunicación usando tecnología Bluetooth hay que tener en cuenta diversos aspectos como pueden ser: el tipo de dispositivo, el alcance, cuantos van a estar conectados etc.

Los dispositivos que se van a utilizar para la realización práctica de esta tesis son los siguientes:

Módulo Transmisor-Receptor junto con la tarjeta motherboard y la antena omnidireccional que transforma la terminal en una estación base. Ver figura 4.11.

Figura 4.11 "Módulo Transmisor-Receptor junto con la tarjeta motherboard y la antena omnidireccional"

También se usará una interfaz USB 1.1 la cual permite a el Autómata ser un receptor de Bluetooth y poder enviar diversos datos según el área de trabajo de este. Ver figura 4.12.

Figura 4.12 "Receptor de Bluetooth"

Después de una explicación del autómata, así como del protocolo de comunicación inalámbrica Bluetooth, las rutinas de enlace y de los dispositivos que este necesita para llevar a cabo dicha comunicación, se explicará la instalación hardware así como las mejoras en el receptor.