ANEXO B Aplicaciones auxiliares del programador

B.1 INTRODUCCIÓN

Además de la aplicación InterGPS, a lo largo de la realización del proyecto se han implementado otros dos programas que nos han ayudado en gran medida a entender mejor el funcionamiento del equipo OEM4 RT-2. La primera de ellas, TxonRxonSerie está basada en una de las primeras versiones de InterGPS y la segunda, CalculaCRC, se utilizó para comprender mejor la creación del código CCR de cada trama.

B.2 LA APLICACIÓN TxonRxonSerie

Las primeras versiones de InterGPS consistían en un software capaz de enviar y recibir mensajes por el puerto serie. Esta versión, no permitía configurar el puerto con diferentes parámetros, tenía una configuración por defecto y sólo podía trabajar con estos parámetros. Debido a la gran facilidad de manejo y a la robustez de las comunicaciones mostrada por este programa, se pensó en crear una aplicación fundamentada en este software, que nos permitiera tener una comunicación directa no solo con el GPS, sino también con las versiones venideras de nuestro proyecto.

La aplicación TxonRxonSerie ha sido de gran utilidad a lo largo de toda la realización del proyecto y sobretodo en el proceso de depuración del programa. Durante las primeras fases de este periodo, InterGPS no estaba capacitado del todo como para trabajar con el GPS directamente, debido a esto se pensó en adaptar las primeras versiones de este programa para simular el funcionamiento del receptor GPS.

Para que TxonRxonSerie fuera capaz de emular un GPS con cierta efectividad, la aplicación debería permitir al usuario enviar y recibir cadenas de caracteres a través del puerto serie y además, que estas cadenas de carácter pudieras ser recuperadas desde el portapapeles de Windows, mediante las opciones “Copiar-Pegar”. Este último requiso era de gran importancia debido a que el envío de las tramas, introduciendo carácter a carácter, resultaba demasiado lento y tedioso, de ahí que simular el comportamiento del GPS mediante el “HyperTerminal” de Windows se hiciera impracticable.

En la siguiente figura podemos ver la simplicidad de este software. No es más que un reducido número de botones y dos campos de texto. En el campo de texto superior introduciremos el comando que queremos transmitir por el puerto serie, mientras que en el inferior se mostrarán los caracteres recibido por el mismo. Como cabe esperar, si pulsamos el botón “Enviar”, el comando que hayamos introducido en el campo de envío

será transmitido, mientras que si pulsamos el botón “Recibir” se irán mostrando una a una, y en el mismo orden en el que llegaron, todas las tramas recibidas.

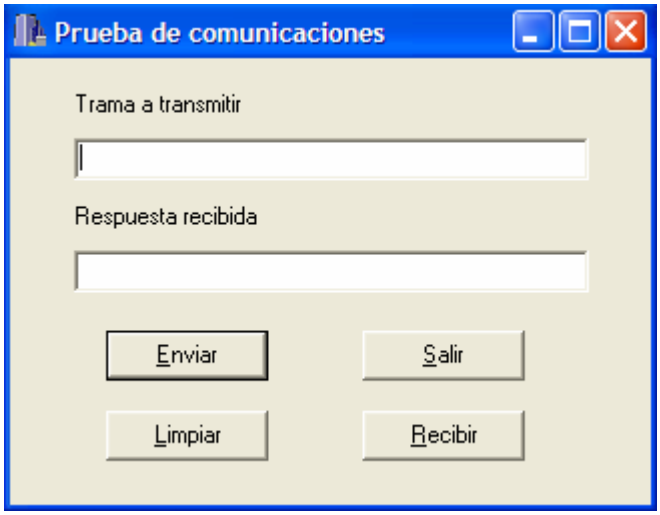

Imagen B.1: Aplicación TxonRxonSerie

El botón “Limpiar” borrará todo lo que haya tanto en el campo de recepción como en el campo de transmisión. Una vez hayamos terminado de trabajar con este programa podremos cerrar la aplicación pulsando el botón “Salir”. Este botón se encargará de cerrar el puerto y liberar toda la memoria reserva antes de terminar la ejecución.

Una de las limitaciones que presenta este programa es que la configuración del puerto se encuentra totalmente preestablecida. Si en un determinado momento queremos trabajar con una configuración distinta o con un puerto distinto, será necesario cambiar el código de la aplicación para que esta configuración sea posible.

La configuración que presenta por defecto la aplicación TxonRxonSerie es la siguiente:

Parámetro	valores
Puerto Com	Com2
Bits por segundo	19200
Paridad	Ninguna
Bits de datos	8
Bits de stop	1
Control de flujo	Ninguno

Tabla B.1 Configuración predefinida de TxonRxonSerie

B.3 LA APLICACIÓN CalculaCRC

Cada una de las tramas recibidas desde el GPS va acompañada por una pequeña cadena de caracteres que hace las veces de código de redundancia cíclico. Este código CCR, va situado al final de la trama y siempre es precedido por el carácter '*'.

Existen distintos códigos de CCR según el tipo de la trama que se este enviado en un determinado momento. Lo que en nuestra memoria hemos llamado tramas de tipo '$', dan lugar a códigos de CCR compuestos por 2 caracteres ASCII, mientras que las de tipo '#' forman CCR de 8 caracteres ASCII. Ambos tipos de CCR serán estudiados en el siguiente apartado.

La aplicación CalculaCRC fue implementada para comprobar el funcionamiento de pequeñas funciones encargadas de codificar y decodificar los diferentes tipos de CCR con los que trabajaría nuestro programa. A medida que se iba probando este software, se puedo apreciar que algunos de los ejemplos de CCR que suministraba el fabricante en su documentación, no eran correctos, de ahí que la aplicación CalculaCRC cobrara mayor importancia en la realización de nuestro proyecto.

Otra de las posibilidades que nos ofrecía este software, era obtener el CCR de tramas que habían sido modificadas previamente, con el fin de que pudieran ser interpretadas posteriormente por nuestra aplicación.

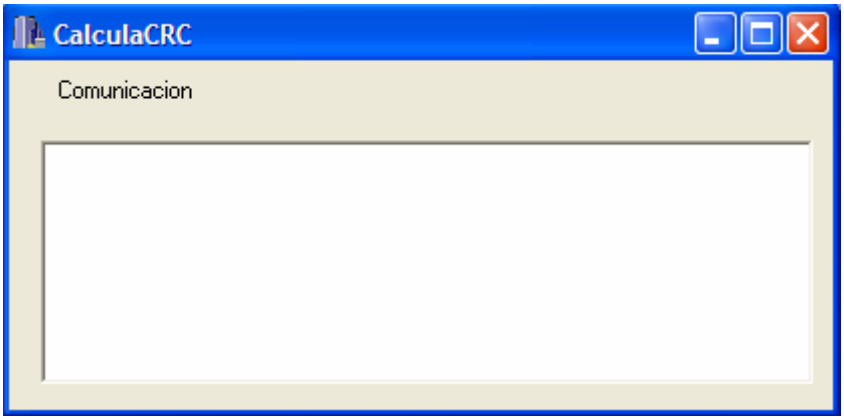

Imagen B.2: Aplicación CalculaCRC

En la imagen superior se ha representado la ventana principal de CalculaCRC. Podemos ver que esta ventana esta formada únicamente por un campo de texto en el que ese introducirá la cadena a estudiar. Si la trama con la que se quiere trabajar no tiene el formato esperado, es decir, no empieza por '$' o '#', el programa mostrará por pantalla el siguiente mensaje.

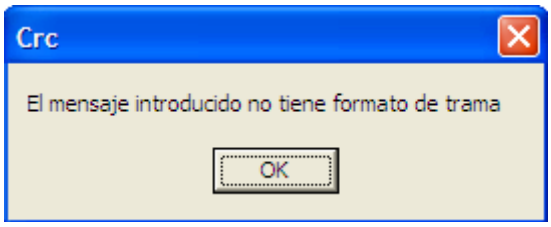

Imagen B.3: Mensaje de error, mensaje sin formato

Si por el contrario la trama tiene el formato esperado, entonces pueden ocurrir dos cosas. En primer lugar que el mensaje que estamos estudiando sea correcta, o lo que es lo mismo que el CCR que hemos introducido al final de trama coincida con el calculado por nuestro programa, en este caso el programa mostrará una ventana como la siguiente.

Imagen B.4 : Trama con código CCR correcto

La segunda posibilidad es que exista algún error en la trama o bien que no se le haya introducido ningún CCR al final. Este caso CalculaCRC indicará que la trama no es correcta y nos dará el CCR propio de la trama con la que estamos trabajando.

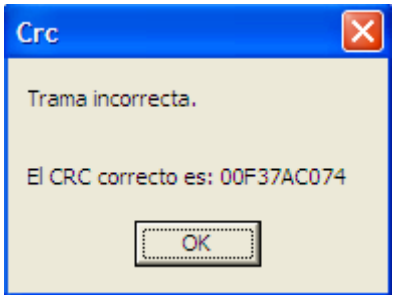

Imagen B.5: Mensaje de trama incorrecta

B.3.1 Tipos de CCR

Una de las diferencias existentes entre los equipos de Novatel RT-20 y los RT-2, es la posibilidad que presentan estos últimos de trabajar con dos tipos de tramas distintas. Los Novatel RT-20 sólo son capaces de trabajar con tramas de tipo '$' mientras que los RT-2 pueden trabajar con este tipo de tramas y además con las tramas de tipo '#'. Una de las diferencias fundamentales entre estas tramas, además de su carácter de inicio de mensaje, '$' y '#' respectivamente, reside en el CCR de cada una de ellas.

- **CCR-8 bits**

Todos los mensajes con carácter de inicio de trama '$' van acompañados por un CCR formado por dos caracteres ASCII. Estos caracteres se correspondes con dos dígitos hexadecimales, es decir 0, 1, … , F, que se obtienen a su vez como el resultado de realizar una operación XOR entre todos los caracteres que conforman dicha trama.

Según esto, para comprobar si una trama de tipo '$' es correcta, recorreremos todos los caracteres del mensaje, incluyendo el de inicio de trama, realizando un XOR entre todos ellos. La operación terminará una vez lleguemos al carácter '*', este no incluido, que indica el final de la información y el inicio del CCR. Como resultado de la operación XOR obtendremos un número que en hexadecimal estará formado por 2 dígitos (8 bits en binario) donde cada uno de ellos se corresponderá con uno de los caracteres ASCII que forman el CCR. Si una vez realizada esta operación, los caracteres del CCR que hemos recibido por el puerto serie coinciden con los calculados anteriormente, entonces y sólo entonces, podremos decir que el mensaje se ha recibido de forma correcta.

Para mayor facilidad de comprensión y estudio en trabajos posteriores, se ha añadido la función que ha venido siendo utilizada en nuestra aplicación para este fin. Podemos ver que como único parámetros se le pasa un puntero a la cadena de caracteres donde está contenida la trama, excluyendo el '*' y todo lo que venga detrás de este. Como resultado se nos devuelve el CCR que hemos calculado.

```
unsigned long CalculateBlockCRC8(unsigned char *cadena)
{
  unsigned long CRC=0;
  int i=0;

while(cadena[i]!='\0')
  {
    CRC=CRC^cadena[i];
    i++;
  }
  return(CRC);
}
```

- **CCR-32 bits**

Los CCR de las tramas de tipo '#' están formados por 8 caracteres, donde cada uno de ellos es la representación en código ASCII de un número hexadecimal: 0, 1, …, F. En este caso no ha sido necesaria la implementación de un código encargado del cálculo de este tipo de CCR, puesto en la documentación que el fabricante nos suministra con los equipos vienen incluidas dos funciones encargadas de esta operación.

A continuación hemos añadido las funciones suministradas por Novatel. Podemos ver que la primera de ellas, *CalculateBlockCRC32,* recibe como parámetros un puntero a la cadena de caracteres donde se encuentra la trama recibida, truncada justo antes del '*', y un unsigned long donde se le indica a la función la longitud de esta cadena. Por otro lado, el valor que nos devuelves será el CCR que hemos calculado. La segunda función, *CRC32Value,* es transparente al programador de la aplicación puesto que no accede directamente a ella, sino que será la primera función la que la invoque.

```
unsigned long CalculateBlockCRC32(unsigned long ulCount,unsigned char *ucBuffer)
{
  unsigned long ulTemp1;
  unsigned long ulTemp2;
  unsigned long ulCRC = 0;

  while ( ulCount-- != 0 )
  {
    ulTemp1 = ( ulCRC >> 8 ) & 0x00FFFFFFL;
    ulTemp2 = CRC32Value( ((int) ulCRC ^ *ucBuffer++ ) & 0xff );
    ulCRC = ulTemp1 ^ ulTemp2;
  }
  return( ulCRC );
}


//--------------------------------------------------------------------------------------------------

unsigned long CRC32Value(int i)
{
  int j;
  unsigned long ulCRC;
  ulCRC = i;

  for ( j = 8 ; j > 0; j-- )
  {
    if ( ulCRC & 1 )
      ulCRC = ( ulCRC >> 1 ) ^ CRC32_POLYNOMIAL;
    else
      ulCRC >>= 1;
  }
```

B.4 APLICACIÓN VIRTUAL SERIAL PORT DRIVER.

Virtual Serial Port Driver es una aplicación encargada de simular la presencia de varios puertos series en una computadora. Una vez instalado VSPD en nuestro PC podemos crear virtualmente tantos pares de puertos conectados entre si como queramos. Gracias a este programa fuimos capaces de simular con un único PC la conexión entre este y un receptor de GPS virtual. Las respuestas de este último eran transmitidas haciendo uso de TxonRxonSerie, aplicación que ha sido presentada con anterioridad en este mismo anexo.

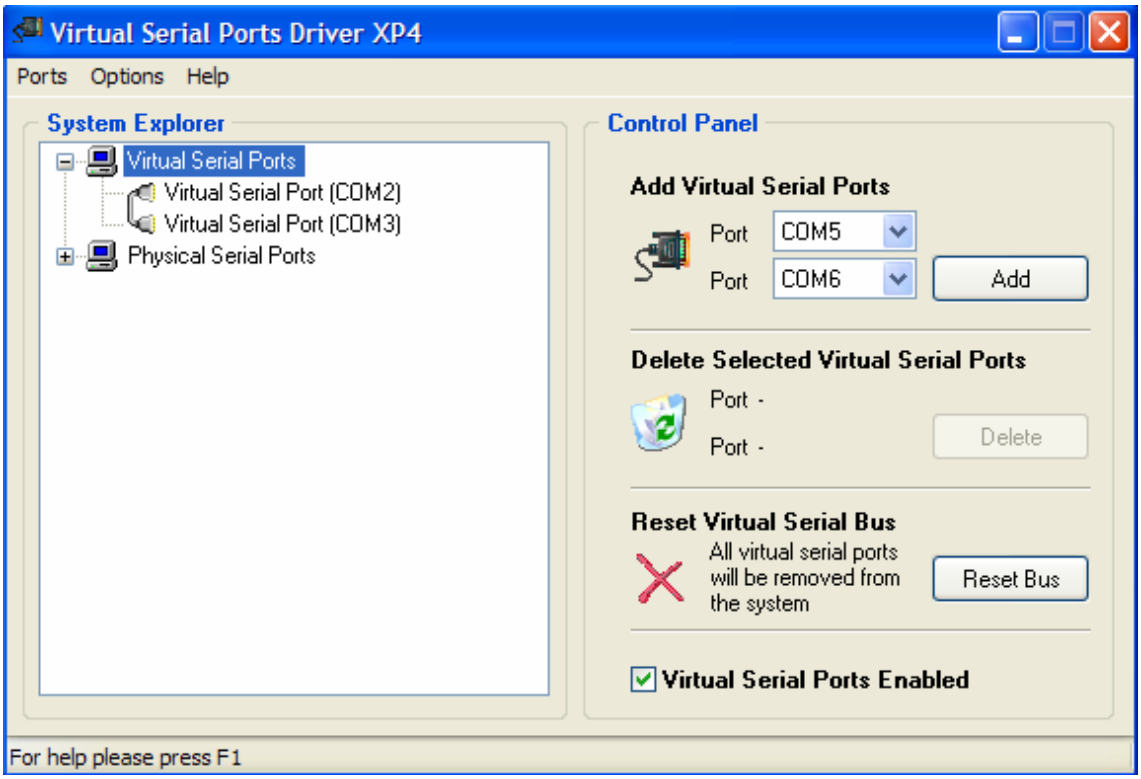

Imagen B.6 : Virtual Serial Port Driver