

2.- Diseño del Software

En este capítulo estudiaremos el software necesario para que las diferentes partes del proyecto funcionen correctamente. Existen dos elementos en este sistema: el ordenador personal y el microcontrolador, que requieren de un software específico para su funcionamiento. Por una parte el ordenador necesitará un driver adecuadamente configurado para que reconozca correctamente el dispositivo de conexión USB y le asigne los recursos necesarios en el sistema operativo para que un programa que diseñaremos pueda enviar datos a través de él.

Por otra parte dicho programa debe servir de interfaz de un modo intuitivo para configurar la conexión con el dispositivo y recoger todos aquellos datos que sean necesarios para el control del mezclador y enviarlos de modo que el microcontrolador sea capaz de entender, procesar y reenviar estos datos para que lleguen en el modo que son requeridos para que los potenciómetros digitales respondan a nuestras consignas.

Es necesario pues diseñar un protocolo de comunicación sencillo que nos permita realizar dicha comunicación. Como cada potenciómetro podrá manipularse independientemente del resto, una manera que si bien no es la más eficiente de realizar dicha comunicación si es bastante favorable a la hora de analizar el funcionamiento del sistema, ya que no es posible su simulación ni su monitorización es la siguiente.

En primer lugar se mandarán desde el ordenador al microcontrolador en código ascii dos dígitos primero la decena y después la unidad que identifican la posición del potenciómetro dentro de la cadena. Y después la posición de este dentro de las 64 posibles posiciones que existen.

El microcontrolador tendrá en la memoria en todo momento las posiciones de los potenciómetros por lo que al recibir los datos, direccionará en la memoria la posición correspondiente al número de potenciómetro recibido y actualizará los datos con la nueva posición. Una vez hecho esto mandará una cadena con todas las posiciones de todos los potenciómetros hacia los integrados para que se actualicen las posiciones.

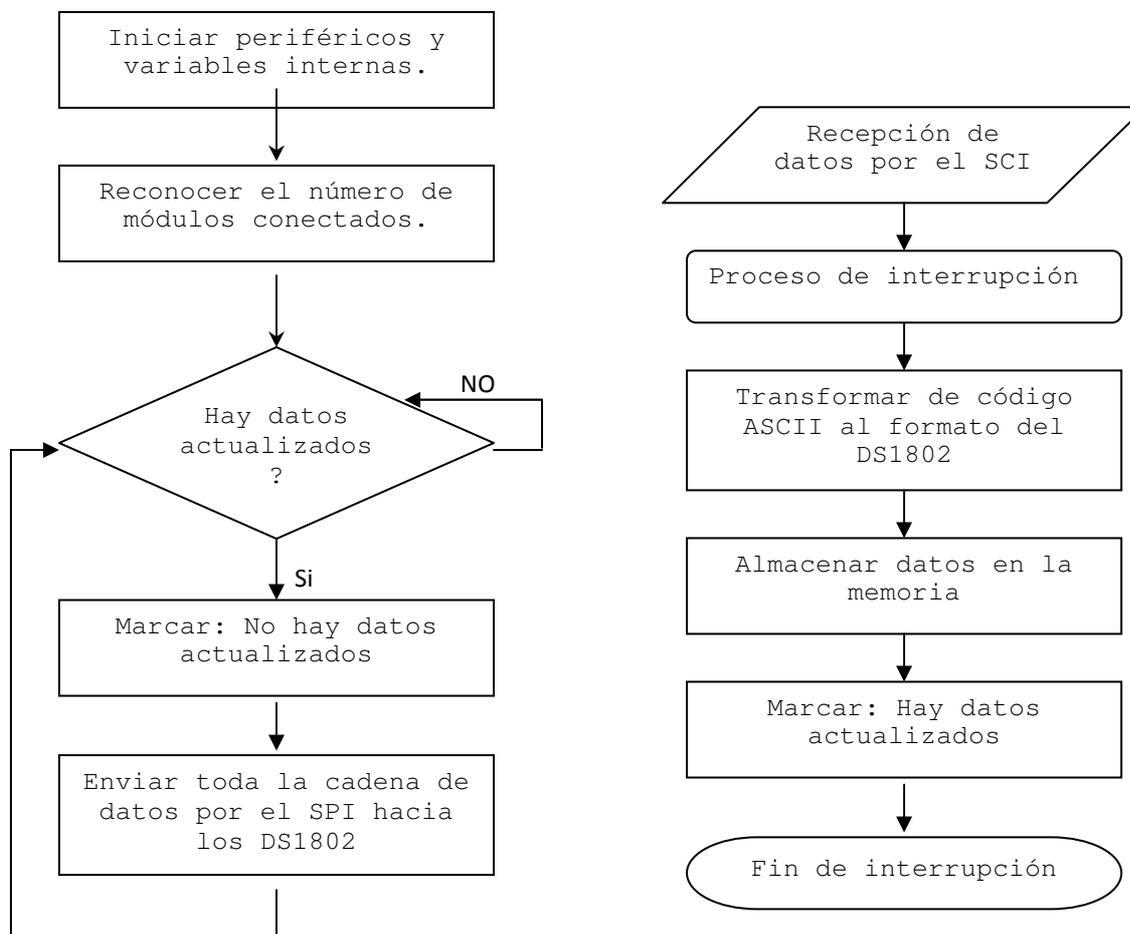
2.1 Software para el 68HC11

Una vez definida la función que debe llevar a cabo el microcontrolador cabe destacar que se va a usar tanto el puerto SCI como el puerto SPI de este. El hecho de utilizar ambos puertos implica que no vamos a tener la oportunidad de utilizar el modo de arranque especial BOOTLOADER del 68HC11. Este modo de arranque permite cargar el programa justo después de resetear el micro por medio del puerto SCI. En nuestro caso el proceso de corrección del programa habrá que hacerlo compilando el código ensamblador mediante el uso del software THRSIM y grabando el código resultante en una memoria EEPROM para posteriormente ejecutar el código en el microcontrolador.

Este programa, el THRSIM, permite la simulación del código en un Pc y la monitorización de todos los registros del 68HC11 durante la simulación, lo que constituye una poderosa herramienta, pero desgraciadamente no nos será de gran utilidad en nuestro caso ya que esta herramienta no simula el funcionamiento del puerto SPI. Por estas razones se ha desarrollado el código en dos fases principales.

En primer lugar se ha desarrollado la parte del código que utiliza el puerto SCI que es la que nos comunicará con el PC mediante el simulador del microcontrolador, y una vez completada se ha desarrollado el resto ejecutando directamente el código en el microcontrolador. En esta segunda fase el programa incluye una serie de rutinas que manda datos hacia el ordenador, que informan de la parte del programa que se ejecuta en cada momento y envía todos los datos que se reciben por el SCI así como los que se deben mandar por el SPI para poder comprobar desde el ordenador si estos son correctos.

De este modo se ha desarrollado un código que pasaremos a comentar a continuación, pero antes tenemos un pequeño diagrama de flujos que nos aclarará un poco las funciones principales del programa:



Podemos ver que el programa tiene tres partes claramente diferenciadas:

- La primera parte de inicialización bastante extensa en la que se configuran los distintos periféricos del microcontrolador y las variables de memoria necesarias y se ejecuta una rutina de búsqueda de módulos de control. Esto se consigue enviando en primer lugar una cadena muy larga de ceros por el SPI para poner a cero todos los registros de los potenciómetros y después teniendo en cuenta que cada módulo tiene 7 integrados que equivalen a 7 registros de 16 bits, se mandan paquetes de 7x16 bits hasta que en el pin de retorno de datos aparezca un uno. Esto sucederá cuando haya mandado 7x16xNúmero de placas conectadas bits más uno. Como se están mandando paquetes de 7x16 bits entre cada comprobación del pin de retorno, tendremos que hay conectadas el número de paquetes de datos enviados menos uno.

Debido a la falta de disponibilidad de integrados DS1802 esta rutina se ha modificado ligeramente para adaptarse al número de integrados existentes realmente de modo que el programa reconozca correctamente la placa con un menor número de estos ya que de no modificarlo reconocerá que no hay ninguna placa conectada y no funcionará correctamente.

- La segunda parte del programa es muy sencilla, se trata de una rutina que comprueba si se han recibido datos nuevos por medio del valor de una variable, de ser así se llama a una rutina que resetea el valor de la variable anterior y envía la cadena completa de datos que se encuentra almacenada en una determinada zona de la memoria del micro.

Hay que tener en cuenta que los datos irán viajando de potenciómetro en potenciómetro por lo que los datos que tienen que llegar al integrado que se encuentre al final de la cadena tendrán que ser enviados en primer lugar y los datos que tengan que llegar al primer potenciómetro tendrán que ser mandados en último lugar para que lleguen en el orden correcto. Además para minimizar el número de transmisiones, mientras se realiza este proceso se podrán recibir nuevos valores de los potenciómetros de modo que si se reciben nuevas posiciones de los potenciómetros a los que aun no se les ha enviado la consigna esta pueda ser mandada sin tener que repetir de nuevo todo el ciclo.

- La tercera parte del programa es una rutina de interrupción que detiene la ejecución del programa cuando llegan datos nuevos por el SCI provenientes del ordenador. Esta rutina los procesa los datos que vienen en formato ASCII. Primero llegan dos números que representan el número de potenciómetro, se pasa a formato binario este número y con el se direcciona la memoria donde se encuentra la posición correspondiente al potenciómetro que se va a modificar su consigna. La posición de memoria es una posición de memoria base que apunta al potenciómetro número uno más el número del potenciómetro menos uno ya que se han reservado 8 bits por cada potenciómetro y en caso de ser el potenciómetro número uno la dirección sería la dirección base más cero.

A continuación se recibe la consigna del potenciómetro, de igual modo se transforma la posición que varía desde 00 hasta 64 lo que forman las 64 posiciones más la correspondiente a MUTE de el DS1802. Este valor se almacena en 8 bits ajustándose los datos a la derecha y se le realiza el siguiente proceso: el bit 0 se intercambia con el 7, el 1 con el 6, el 2 con el 5 y el 3 con el 4. De este modo cuando se manden los datos por el SPI estos llegarán del modo que lo requieren los potenciómetros según indica el fabricante de estos en su datasheet.

Para finalizar esta rutina, se almacena el dato recibido en la memoria para poder ser enviado posteriormente y se marca una variable que indica que se han recibido datos nuevos.

A continuación tenemos el código ensamblador del programa comentado en el que se ve con total detalle los pasos que sigue el microcontrolador, los comentarios están resaltados en diferente color para mayor claridad:

```
*****
```

```
*      Definicion de registros      *
```

```
*****
```

```
SCDR EQU $102F *Registro para leer o enviar datos por SCI
SCCR2 EQU $102D *Registro de control del SCI
SCSR EQU $102E *Registro de estado del SCI
BAUD EQU $102B *Configura la velocidad del SCI y el SPI
SPCR EQU $1028 *Registro de control del SPI
SPSR EQU $1029 *Registro de estado SPI,SPFLG <- Indica fin de transmision
SPDR EQU $102A *Analogo al SCDR del SCI para el SPI
PORTD EQU $1008 *Puerto D
DDRD EQU $1009 *Registro de direcciones del puerto D ->SPI
PORTA EQU $1000 *Direccion del puerto de proposito general A
```

```
*****
```

```
*      Definicion de variables      *
```

```
*****
```

```
ORG $0000
```

```
NCAN RMB 1 *Aqui almacenaremos el numero de canales detectados
ENVIA RMB 1 *Esta variable indica si hay que mandar datos por SPI
AUX RMB 1 *Las variables AUX son para pasos intermedios del programa
AUX2 RMB 1
AUX3 RMB 1
DEBUG RMB 1 *Variable para indicar el progreso del programa
DATA1 RMB 14 *Cada variable data guarda las posiciones de los potenciómetros de un modulo de
                proceso de señal
DATA2 RMB 14
DATA3 RMB 14
DATA4 RMB 14
```

```
*****
```

```
*      Inicio de la configuracion del micro      *
```

```
*****
```

```
ORG $E000
```

```
START LDS #$FF
SEI *Hasta que no se configure todo no se atenderá a los datos que lleguen por el SCI
LDAA #%00101100
STAA SCCR2 *Habilita el sci entrada, salida y la interrupcion por recepcion de datos
LDAA #%00110000
STAA BAUD *Configurar la velocidad del sci para recibir por el USB a 9600B
LDAA #%00111000
```

```

STAA DDRD *Configura los pines del puerto D para funcionar en modo SPI y maestro
LDAA #%01010011
STAA SPCR *Activa SPI,sin interrupciones,como master,CLK=E/4,Active High,CPHA=0
            configuracion que es requerida por el DS1802

```

```
*****
```

```
*      Inicializacion de Variables      *
```

```
*****
```

```

INIVA LDAA  #%00000000
      STAA  ENVIA *ENVIA=0 ->No hay datos nuevos a mandar
      STAA  AUX  *En aux meteremos el numero de placas temporalmente
      LDAA  #%00010000
      STAA  NCAN *Numero de canales=16, necesario para la deteccion de canales
      LDY   #PORTD
      BCLR  0,Y %00100000 *poner a cero el SS -> Habilita los DS1802
      LDAA  #$00
      LDX   #DATA1

BORRA STAA  0,X
      INX
      CPX   #DATA4+14
      BNE  BORRA *Esta rutina pone a cero todas las variables data
      BSR  RUT_SPI *Saltar a la rutina de mandar datos por el SPI, para poner a cero todos los
                  valores de los potenciómetros.

```

```
*****
```

```
*      PREGUNTAR EL NUMERO DE CANALES EXISTENTES      *
```

```
*****
```

```

LDAA  #$FF
LDX   #DATA1

```

```

BORRA2 STAA  0,X
      INX
      CPX   #DATA4+14
BNE BORRA2 *Rutina que pone un 1 en todas las variables DATA similar a la anterior que los
            ponía a cero
      LDAA  PORTA
      ANDA  #%00000001
      BEQ  RECON2
      BRA  INIVA *Miramos a ver si el pin retorno que es el ultimo de puerto A esta puesto a uno.

```

Esto no debe suceder ya que solo hemos mandado ceros a los potenciómetros, si esto sucede es muy probable que no se haya realizado correctamente el montaje, por lo que retrocederemos e intentaremos borrar de nuevo los potenciómetros.

```

RECON2 LDAA  #$00
      STAA  AUX

```

```

RECON LDAA  AUX
      INCA
      STAA  AUX

```

```

BSR    RUT_SPI
LDAA  PORTA
ANDA  #%00000001
BEQ   RECON
LDAA  AUX    *Esta es basicamente la rutina de reconocimiento, se envia un paquete de 7x16
           unos y se comprueba si el pin de retorno se ha puesto a uno, sino se manda otro
           paquete. Cuando se ponga a uno es que los datos han pasado por todos los
           integrados y hay un numero de canales igual al numero de paquetes mandados
           menos uno. Como en el prototipo hay menos de 7 integrados el pin retorno se
           pondra a uno cuando se mande un solo paquete de datos por lo que se incluye
           comentada la linea siguiente para el correcto funcionamiento del prototipo.
*
*  DECA
STAA  NCAN  *Almaceno el numero de canales en NCAN
ADDA  #$30  *Sumo $30 y tengo el ASCII del número de canales
LDAB  SCSR  *Preparo el SCI para mandar datos por primera vez
STAA  SCDR  *Envio el numero de canales al pc para que su software tenga en cuenta cuantos
           canales hay conectados.

ESP   LDAA  SCSR
      BPL   ESP  *Espero a que se mande el numero de canal

*****
*     BUCLE PRINCIPAL                               *
*     RECIBIR DATOS DEL SPI                          *
*     ACTUALIZAR EL ESTADO DEL MEZCLADOR             *
*****
*La siguiente rutina comprueba si se han recibido datos nuevos mediante la variable ENVIA, si se han
recibido se llama a la rutina RUT_SPI que se encarga de realizar el envio de datos por el SPI.

      CLI           *Habilito las interrupciones

LOOP  LDAA  ENVIA
      DECA
      BEQ   SALTO
      BRA  LOOP

SALTO JSR    RUT_SPI
      BRA  LOOP

*****
*     RUTINA RUT_SPI                               *
*     MANDA DATOS POR EL SPI                       *
*****

RUT_SPI  LDY  #PORTD
          BSET 0,Y %00100000          *Subir el SS para habilitar la comunicacion con los DS1802
          LDX  #DATA1
          LDAA #%00000000
          STAA ENVIA          *Envia=0, reseteamos la bandera de datos pendientes
          LDAA NCAN
          LDAB #$00

```

```

VUELVEADDDB  #%00001110
    DECA
    BEQ  MANDA
    BRA  VUELVE          *Creamos un indice con el numero de datos a mandar, 14 veces el numero
                        *de placas conectadas
MANDA        TBA          *Paso el indice al acumulador A
MANDAA LDAB  0,X          *Pongo en el acumulador B la consigna del ultimo potenciómetro
    STAB  SPDR          *Inicio la transmision
ESPERA LDAB  SPSR
    BPL  ESPERA *Espero a que el dato se haya mandado por completo
    INX          *Direcciono el valor del siguiente potenciómetro
    SUBA  #$01      *Decremento el indice de numero de envios pendientes
    BEQ  FINSPI *Compruebo si he terminado de enviar los datos
    BRA  MANDAA   *Si no he terminado continuo mandando
FINSPI LDX   #PORTD
    BCLR 0,X %00100000 *Orden a los potenciómetros de adoptar las nuevas consignas
    RTS          *Fin de la rutina

*****
*   RUTINA RUT_SCI          *
*   RECIBE Y ESTRUCTURA LOS DATOS DEL SCI *
*****
*Esta rutina comienza con una interrupcion que se dispara al recibir un dato a traves del puerto SCI

RUT_SCI     SEI          *Deshabilito temporalmente las interrupciones

SCIFLG1     LDAA  SCSR
    ANDA  #%00100000
    BEQ  SCIFLG1      *Compruebo que los datos se han recibido correctamente
    LDAA  SCDR        *Leo el dato recibido correspondiente a la decena del numero de
                        *potenciómetro en ASCII
    STAA  AUX2
    LDAB  #$00
    SUBA  #$30        *Convierto el numero de ASCII a binario
    BEQ  CANAL2      *Si este numero es cero no es necesario procesarlo

CANAL1      ADDB  #%00001010
    DECA
    BEQ  CANAL2      *Si no es cero lo multiplico por 10 pues se trata de las decenas
    BRA  CANAL1

CANAL2      LDAA  SCSR
    ANDA  #%00100000
    BEQ  CANAL2      *Espero a recibir la segunda cifra del numero
    LDAA  SCDR
    SUBA  #$30      *Una vez recibida la convierto a binario
    ABA          *Le sumo las decenasX10 y tengo el numero de potenciómetro
    DECA

```

```

    STAA AUX      *Almaceno en aux el numero de pot -1 = Salto de memoria necesario para
                  direccionarlo desde el primer dato

SCIFLG2      LDAA  SCSR
             ANDA  #%00100000
             BEQ   SCIFLG2
             LDAA  SCDR      *A continuacion espero a recibir la primera cifra de la consigna
             LDAB  #%00000000
             SUBA  #$30
             BEQ   SCIFLG3
CIFRA1 ADDB  #%00001010      *Igual que antes la multiplico por 10
             DECA
             BEQ   SCIFLG3
             BRA   CIFRA1
SCIFLG3 LDAA  SCSR
             ANDA  #%00100000      *Recibo la segunda cifra
             BEQ   SCIFLG3
             LDAA  SCDR
             SUBA  #$30
             ABA           *La sumo en binario a la anterior y obtengo la consigna
             STAA  AUX3      *Guardo este dato en la variable AUX3
             LDY   #AUX3
             LDAA  #$00
             BRCLR 0,Y #%10000000 BITA
             ADDA  #%00000001
BITA  BRCLR 0,Y #%01000000 BITB
             ADDA  #%00000010
BITB  BRCLR 0,Y #%00100000 BITC
             ADDA  #%00000100
BITC  BRCLR 0,Y #%00010000 BITD
             ADDA  #%00001000
BITD  BRCLR 0,Y #%00001000 BITE
             ADDA  #%00010000
BITE  BRCLR 0,Y #%00000100 BITF
             ADDA  #%00100000
BITF  BRCLR 0,Y #%00000010 BITG
             ADDA  #%01000000
BITG  BRCLR 0,Y #%00000001 BITH
             ADDA  #%10000000
BITH  STAA  AUX3

```

*Este codigo escanea bit a bit la consigna recibida del potenciómetro desde el bit 7 hasta el 0 y si hay un 1 en la posición 7, escribe un 1 en la posición 0 si lo hay en la 6 lo escribe en la 1 y así sucesivamente con lo que realiza una operación de simetría especular en el registro. Esto es necesario porque los integrados DS1802 requieren que se les manden los datos de este modo.

```

LDX  #DATA1
LDAB AUX
ABX
STAA 0,X      *Por ultimo almaceno el dato ya procesado en la direccion de memoria
                  correspondiente al potenciómetro
LDAA  #%00000001

```

```
STAA  ENVIA  *Le indico al bucle principal que hay datos pendientes
CLI      *Vuelvo a habilitar las interrupciones
RTI      *Doy por finalizada la interrupcion
```

```
*****
```

```
*      VECTOR DE INTERRUPCIONES      *
```

```
*****
```

```
ORG  $FFFE
FDB  START      *Vector de inicio despues de reset

ORG  $FFD6      *Vector de interrupcion asociada la SCI
FDB  RUT_SCI
```

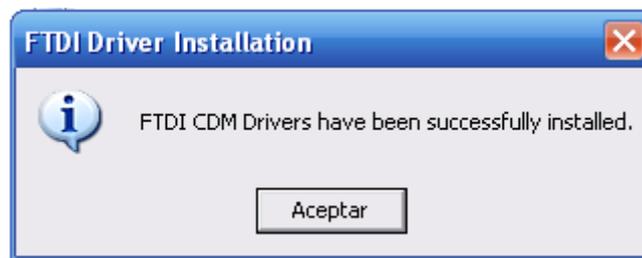
2.2 – Software para Pc

En este capítulo analizaremos el software que es necesario instalar en el pc para poder comunicarse con el mezclador y el desarrollo y manejo del software diseñado específicamente para este proyecto.

2.2.1 – Drivers

Debido a que el Pc será conectado directamente vía USB al FT232BL, será necesario instalar un paquete de drivers para que se pueda acceder a las funcionalidades que ofrece este integrado. Dependiendo del tipo de sistema operativo al que vayamos a conectar la placa estos drivers se instalarán de distinto modo.

En el caso de utilizarlo en sistemas windows basados en tecnología NT como pueden ser Windows Nt, 2000, XP y XP 64bit edition, basta con ejecutar el archivo ejecutable que se encuentra en la carpeta “CDROM:\Drivers\Winnt\Instalar.exe”, tras un breve periodo de actividad recibiremos el siguiente mensaje.



Posteriormente conectaremos el dispositivo mediante un cable USB y el sistema reconocerá el hardware y nos preguntará si queremos conectar a internet para buscar un controlador, marcaremos “no conectar a internet” y pulsaremos siguiente. A continuación pulsamos siguiente de nuevo y windows buscará el controlador y lo configurará automáticamente el dispositivo asignándole un número de puerto de serie COM que se encuentre disponible, normalmente será COM3 si no existen otros dispositivos de serie instalados en el PC.

En el caso de utilizar un windows del tipo 9x, como windows 98, 98 SE o Millenium, conectaremos el dispositivo directamente y seguiremos los siguientes pasos:

1. Cuando aparezca la pantalla "Añadir nuevo hardware" haremos clic en Siguiente y a continuación insertaremos el CD de instalación en la unidad de CD-ROM.
2. Marcaremos “Buscar el mejor controlador para su dispositivo (se recomienda)” y haremos clic en Siguiente.
3. Haremos clic en “Especificar una ubicación” y escribiremos “D:\Drivers\Win9x”. A continuación, haremos clic en Siguiente. Si su unidad de CD-ROM tiene asignada una letra diferente, teclaremos su letra en lugar de la “D”.
4. A continuación aparece una ventana en la que debemos marcar la opción “El controlador actualizado(Recomendado)” y pulsar siguiente. Dos nuevos diálogos aparecen para confirmar la elección, pulsaremos siguiente en ambos de nuevo para proseguir con la instalación.

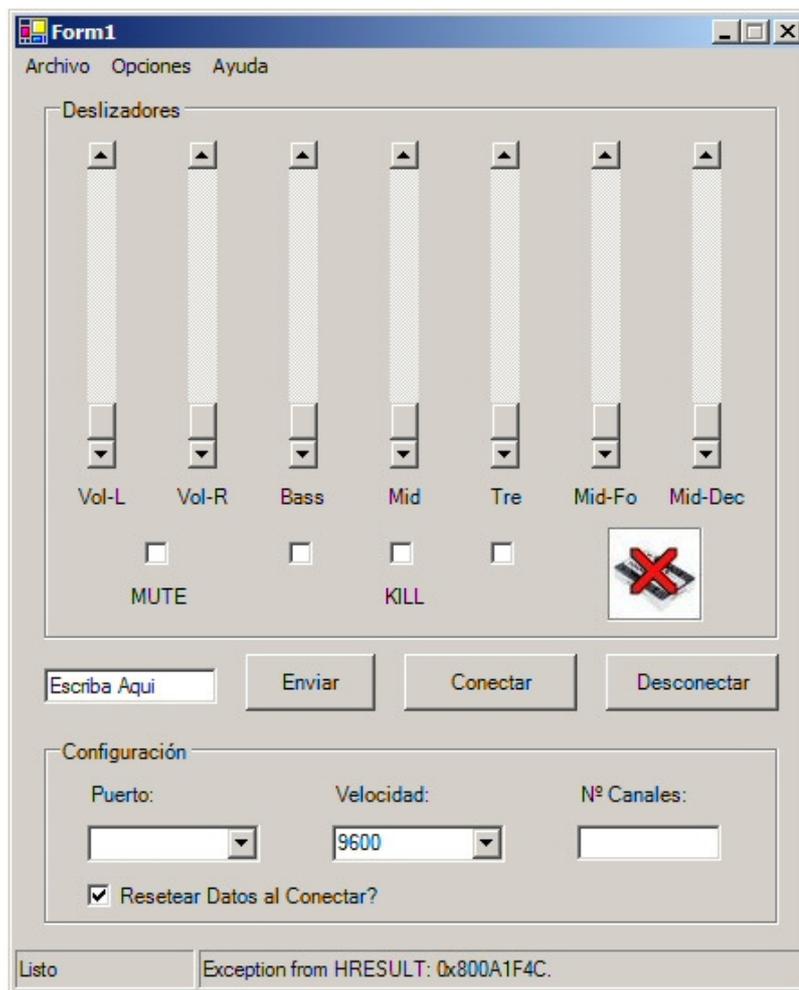
5. En este momento la instalación del controlador a finalizado, pero windows detectará un puerto de serie de modo que hay que repetir el proceso anteriormente descrito de nuevo para completar la instalación.

Se pueden encontrar versiones de los drivers actualizadas de modo gratuito a través de internet en la dirección <http://www.ftdichip.com/Drivers/VCP.htm>. Sin embargo es importante que dichos drivers se encuentren firmados digitalmente bajo la norma "Microsoft WHQL Certified" ya que estos controladores han sido testados por Microsoft y están sujetos a una menor cantidad de fallos. Además el proyecto ha sido desarrollado utilizando estos controladores por lo que es más probable que exista algún tipo de incompatibilidad en el caso de usarse otros controladores.

2.2.2 – Interfaz del usuario

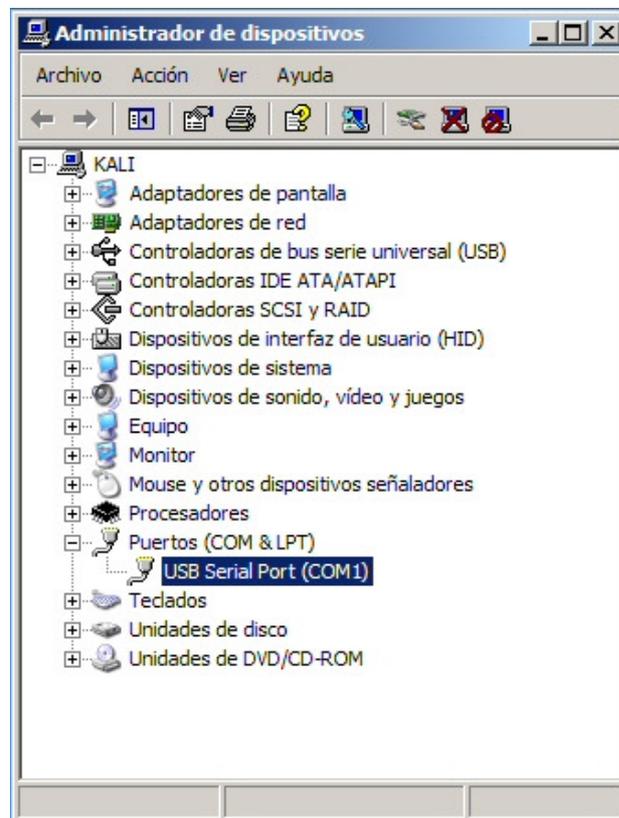
Una vez se ha concluido la instalación de los controladores es necesario ejecutar el programa que nos permitirá el control del mezclador. Pero previamente debemos preparar al sistema operativo. Para ello debemos instalar algunas librerías necesarias que se distribuyen gratuitamente y que se incluyen en el CDROM de documentación y software. En primer lugar es necesario instalar la actualización Microsoft.NET framework que se encuentra disponible en “CDROM:\Interfaz\Librerias\donetfx.exe” y a continuación debemos ejecutar el archivo de registro “CDROM:\Interfaz\Librerias\licencia.reg” y pulsar Si en el dialogo que aparece. Si todos los pasos de la instalación se han ejecutado correctamente aparecerá un diálogo comunicando que se ha actualizado el sistema, tras el cual debemos reiniciar el equipo para que los cambios surjan efecto.

Finalizada la actualización del equipo podemos ejecutar la interfaz que se encuentra en “CDROM:\Interfaz\Mezclador.exe” que nos mostrará el siguiente programa:

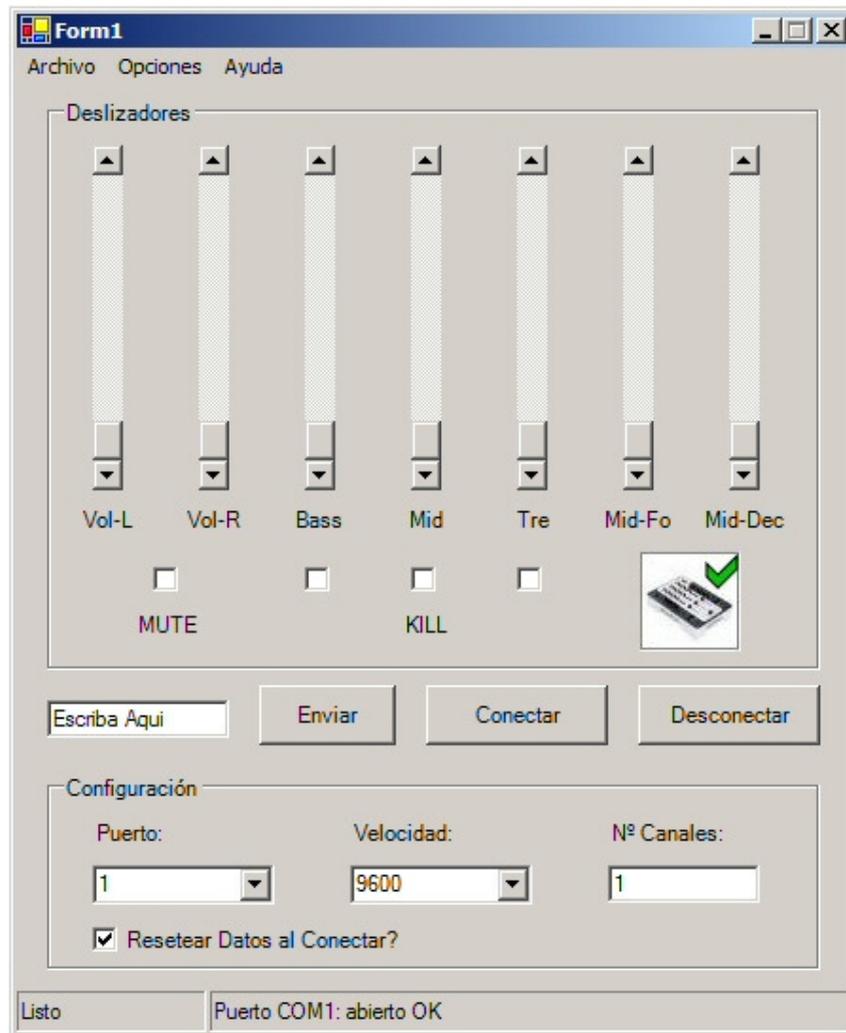


En primer lugar debemos seleccionar el puerto de serie en el que se encuentra conectado el mezclador. El software detectará los puertos de serie que se encuentran activos en el momento de arrancar la aplicación de modo que la mesa de mezclas debe encontrarse conectada en uno de los puertos que aparecen en la lista desplegable. Una vez seleccionado el puerto pulsamos el botón conectar y se realizará la conexión con la mesa de mezclas.

Si bien el programa rastrea los puertos de serie que están disponibles en el ordenador, es posible que existan varios puertos disponibles debido a que normalmente los ordenadores disponen de al menos un puerto de serie. En este caso se puede acudir en el ordenador al panel de control del equipo, ejecutar la ficha sistema y a continuación administrador de dispositivos. Nos aparece una ventana donde aparece la ficha puertos y dentro de esta ficha USB Serial Port, indicando el número de puerto asignado entre paréntesis.



Si la conexión se ejecuta con éxito, el icono de estado cambiará apareciendo un símbolo de validación verde y se rellenará la casilla correspondiente indicando el número de canales conectados como se muestra en la siguiente figura:



Si se marca la casilla resetear datos, todos los potenciómetros del mezclador se pondrán a nivel cero, esto es, mínimo volumen y máxima atenuación en todos los ecualizadores, sino no se actuará sobre el mezclador. Ya podemos entonces actuar sobre los distintos potenciómetros sin más que seleccionar el número de canal sobre el que queremos actuar y moviendo los deslizadores de los distintos potenciómetros. Para actuar sobre los distintos canales es necesario seleccionar el número de canal sobre el que queremos actuar en el cuadro número de canal.

A modo de herramienta de desarrollo se ha dejado el cuadro situado a la izquierda del botón enviar donde se pueden escribir caracteres que serán enviados por el puerto USB hacia el microcontrolador para poder comprobar el funcionamiento de las distintas partes del sistema.

Los botones Kill, adoptan la posición de máxima atenuación de los distintos ecualizadores en un solo paso, y el botón MUTE selecciona el mínimo volumen disponible en ese canal.

Hay que recalcar el hecho de que el sistema es plenamente independiente del pc y que en el caso de que se pulse el botón desconectar, el mezclador permanecerá funcionando mientras reciba alimentación con las consignas que haya recibido. También hay que decir que si bien el sistema ha sido preparado para que en el arranque se posicione de modo de máxima atenuación, en sencillo realizar un pequeño ajuste en los valores iniciales para que deje pasar el sonido y se tenga una respuesta plana a la salida o bien con una ecualización determinada haciendo posible el funcionamiento del mezclador de

modo independiente a la computadora. Usando esta solamente en el caso de tener que realizar un ajuste en la configuración del sistema.

También es posible controlar bajo un mismo computador varios sistemas, ya que se pueden ejecutar varias instancias del programa simultáneamente y cada una se podrá conectar a un mezclador diferente. Además debido al bajo consumo de memoria del programa y la escasa necesidad de cálculos cualquier ordenador dentro de las especificaciones requeridas, podrá ejecutar el software sin producirse disminuciones del rendimiento del equipo.

En caso de que durante la operación del equipo se produjesen errores de cualquier tipo, la barra de estado del programa nos va informando del estado de nuestras acciones, de modo que si no se puede enviar algún dato o no se pueden realizar conexiones correctamente el programa nos informa de ello a cada momento.