X36−X35 NT Driver

Documentación para programadores

CAPITULO I

El documento siguiente está pensado para personas con conocimientos de programación, aunque no por ello deja de ser interesante para aquellos que quieran conocer en profundidad cómo funcionan los joysticks X36 y X35 de Saitek.

Los joysticks

Los joysticks X36 y X35 puede trabajar conjuntamente o por separado y existen en dos versiones distintas ***F*** y ***T*** .
El X36 es un joystick normal de dos ejes mientras que el X35 es una palanca de gases (acelerador) que además incorpora un eje que trabaja como pedales. Las versiones ***F*** son versiones programables y las versiones ***T*** no son programables, aunque el ***X35−T*** puede ser programado si se combina con el ***X36−F***.

Estos joysticks tienen una doble función :

− Por un lado tienen una función analógica que los hace funcionar como un joystick normal con acelerador y pedales.
Esta función se realiza a través del puerto de juegos y de forma independiente de la función digital, por lo tanto, para que el joystick realice está función no es necesario tener enchufado el conector del teclado al puerto correspondiente.

− Por otro lado está la función digital que es la parte "interesante" del joystick, y hace que los botones o ejes funcionen como si fueran las teclas de un teclado.
Esta función se realiza a través del puerto del teclado y es dependiente del puerto de juegos.

Inicializando los joysticks

Para usar las funciones digitales hay que usar el puerto de teclado tanto para entrada como para salida. La entrada nos permite saber qué botón se ha pulsado y la salida permite programar las funciones.

Hay dos formas de inicializar los joysticks, una con la configuración de fábrica y otra con una configuración personalizada. *Si no se inicializan los joysticks quedan configurados a 4 botones con acelerador y pedales.*
La inicialización de fábrica se consigue pulsando el botón ***Lanzar*** una vez que el sistema operativo se a iniciado y da lugar a una configuración de 6 botones con 2 POV, acelerador y pedales.

Para inicializar el joystick de forma personalizada hay que enviar por el puerto de teclado exactamente 14 bytes. Por cada byte enviado se recibirá un byte de confirmación del joystick excepto en el último que se reciben 9 bytes. La descripción de los bytes es la siguiente (notación decimal):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14
110	16	11	Btn1	Btn2	Btn3	Btn4	Btn5	Btn6	POV1	POV2	AyP	8	16

- **Btn? :** Botones del puerto, indican qué botones* de los joysticks serán asignados como los botones 1−6 de la función analógica.
- **POV? :** Con valores de 0 a 4, indican cúal de las 4 setas será asignada como POV 1 y como POV 2 de la función analógica (0 = sin asignar).
- **AyP :** Configuración del acelerador y los pedales :
  - ♦ 0 = acelerador y pedales como función analógica.
  - ♦ 1 = acelerador analógica, pedales digital.
  - ♦ 2 = acelerador digital, pedales analógica.
  - ♦ 3 = acelerador y pedales en función digital.

**Los códigos de los botones de los joysticks a usar en los bytes Btn? aparecen en la tabla del final del capítulo II (0 = sin asginar).*

Cada vez se reinicia el sistema el joystick se resetea con lo cual se pierde la configuración y hay que reinicializarlo.

Por defecto, al resetearse, los joysticks tienen la configuración de 4 botones con acelerador y pedales.

Una vez iniciado el joystick no se puede cargar la configuración de fábrica usando el botón Lanzar, a no ser que se halla reseteado el joystick.

Función digital

Una vez inicializado el joystick todos los botones, ejes y rotatorios funcionan. Sin embargo, ***no se pueden programar***, es decir, no se puede enviar un conjunto de datos al joystick para decirle "al pulsar el botón A envia la tecla INTRO".

Sin embargo, una vez inicializado el joystick cada vez que se pulsa/suelta algún botón o se mueve algún eje (funcionando en digital) o rotatorio se recibe por el puerto de teclado un código de 4 ó 6 bytes, en concreto :

Códigos para botones

Byte 1	Byte 2	Byte 3	Byte 4
111	16	1	ID

Códigos para ejes y rotatorios

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
111	16	3	ID (33, 34, 35 ó 36)	Subposición (48−63)	Posición (48−63)

Conociendo esto, el problema de programar cada botón se resuelve con un programa externo capaz de reconocer los códigos y reemplazarlos por la tecla o teclas que queramos. No obstante, se nos presenta otro problema....

Al enviar el joystick los códigos al puerto de teclado éstos se interpretan como teclas con lo cual al pulsar un botón se escriben 4 o 6 letras que no hemos pulsado en el teclado. Esto significa que aunque traduzcamos los códigos en sus teclas correspondientes, es necesario eliminar los códigos antes de que se escriban.

X36−X35 NT Driver

Documentación para programadores

CAPITULO II

X36−X35 NT Driver

Para iniciar y "traducir" los códigos de los joysticks dentro de un sistema Window NT se ha encontrado que la única manera de hacerlo es añadir un controlador de filtro de teclado (**kbfiltr.sys**).
De esta forma se puede pasar por alto el bloqueo por parte del núcleo NT del envío de datos a través del puerto de teclado y también se pueden moficar los datos del buffer del teclado antes de que sean procesados.

En lo referente al código del archivo, éste se ha programado de forma deja intacta la interfaz "de fábrica" del sistema NT, es decir, una vez instalados funcionan exactamente igual que los originales que vienen con el sistema operativo lo que asegura la mayor compatibilidad posible.
Una consecuencia notable de esto es que, por ejemplo, aunque los joysticks funcionan, sigue sin ser posible enviar datos a través del puerto de teclado.

Control del Driver

En la versión 2.0 del Driver se ha cambiado el sistema para el control del Driver, anterior mente se usaba el API de control de dispositivos de Windows redirigiendo las llamadas al driver del teclado, pero a partir de esta versión el Driver tiene su propia interfaz de control.

Para llamar a las funciones del Driver, tal como se ha dicho, se usa el API del control de dispositivos...

```
DeviceIOControl( dispositivo, llamada, &datos, tamaño, NULL, 0, &respuesta, NULL);
```

donde dispositivo es el nombre de la interfaz del Driver. Es un string definido como :

```
"\\.\X36DriverInterface"
```

Además también se ha añadido un sistema para la emulación del ratón usando los servicios de Windows NT, el sistema es transparente y no necesita llamadas adicionales. No obstante si que es posible cambiar la sensibilidad del ratón, lo cual puede hacerse a través del sistema de mensajes de windows.

A continuación se especifican las funciones que se pueden llamar.

LLamada IOCTL_SAITEK_INI (*Inicialización*)

```
CTL_CODE(FILE_DEVICE_UNKNOWN, 0x0100, METHOD_BUFFERED, FILE_ANY_ACCESS)
```

Esta llamada sirve para inicializar los joysticks. *&datos* es la dirección del buffer de 11 bytes de datos que se enviará al driver.

Buffer

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
Btn1	Btn2	Btn3	Btn4	Btn5	Btn6	POV1	POV3	AyP	Modos	Pinkie

- ***Modos*** es un booleano que indica si se usará la palanca de modos como selector de modos (*true (1)*) o como botón (*false (0)*).
  Cuando se inicializa a false, se usa la configuración del Modo 2.
- ***Pinkie*** es un booleano que indica se usará la función pinkie o no.

LLamada IOCTL_SAITEK_COMBI (*Teclas combinables*)

```
CTL_CODE(FILE_DEVICE_UNKNOWN, 0x0104, METHOD_BUFFERED, FILE_ANY_ACCESS)
```

Esta llamada sirve para informar al driver de cuales son los scancodes de las teclas especiales del teclado (CTRL, ALT, MAY y WIN).
&datos apunta a un buffer de 8 bytes que contiene los scancodes de las teclas, 4 para las teclas de la izquierda y 4 para las de la derecha (el orden es indiferente).

LLamada IOCTL_SAITEK_ENVIAR_AUTO (*Autorepeticiones*)

```
CTL_CODE(FILE_DEVICE_UNKNOWN, 0x0103, METHOD_BUFFERED, FILE_ANY_ACCESS)
```

Esta llamada sirve para informar al driver de cuales son los botones que tendrán auto−repetición. El buffer consta de 37 bytes y hay que realizar 6 llamadas (una por modo) para configurar todos los botones.

- El primer byte indica el modo a configuar (ver tabla siguiente).
- Los 36 bytes restantes son booleanos que se corresponden con los 36 botones de los joystick (ver tabla del final del capítulo). Se hacen **true** si el botón tendrá auto−repetición.

Modo	0	1	2	3	4	5
Configuración	Modo 1	Modo 1 & Pinkie	Modo 2	Modo 2 & Pinkie	Modo 3	Modo 3 & Pinkie

LLamada IOCTL_SAITEK_ENVIAR_TECLA (*Teclas*)

```
CTL_CODE(FILE_DEVICE_UNKNOWN, 0x0101, METHOD_BUFFERED, FILE_ANY_ACCESS)
```

Esta llamada sirve para enviar al driver las teclas asignadas a cada botón, eje o rotatorio. Como en la llamada de autorepeticiones hay que hacer varias llamadas de este tipo para completar la configuración.
En este caso el buffer lo compone la siguiente estructura (de 24 bytes).

```
typedef struct _PAQUETE_SAITEK
{
UCHAR Modo;
UCHAR Posicion;
UINT64 Dato;
UCHAR DatoExt;
} PAQUETE_SAITEK, *PPAQUETE_SAITEK;
```

- ***Modo*** es igual que el índice de modo de la llamada anterior.
- ***Posición*** identifica el botón,eje o rotatorio para el cual se envían las teclas (ver tabla al final del capítulo..
- ***Dato*** indica los scancodes las teclas que serán enviadas al pulsar el botón, eje o rotatorio. Al ser un `UINT64` puede haber 8 scancodes.
- ***DatoExt*** indica cuales de las 8 teclas son teclas extendidas. Cada bit se corresponde con un byte de *Dato*, si el bit es 1 la tecla es extendida y si es 0 no lo es.

LLamada IOCTL_SAITEK_ENVIAR_TROT (*Rotatorios*)

```
CTL_CODE(FILE_DEVICE_UNKNOWN, 0x0102, METHOD_BUFFERED, FILE_ANY_ACCESS)
```

Esta llamada sirve para describir el funcionamiento de los rotatorios y los ejes (cuando funcionan en digital).
En este caso el buffer de datos consta de 24 bytes definidos como una matiz `UCHAR ejes[4][6];`. El primer índice de la matriz indica el rotatorio/eje y el segundo el modo (ver tabla más abajo). El byte será:

- ***bits 7 y 8*** : no usados.
- ***bit 6*** : 1 si el eje/rotatorio será de tipo incremental, 0 para posiciones.
- ***bits 1−5*** : número de 5 bits que indica el número de posiciones (*Para los ejes puede ser como máximo 6 y para los rotatorios como máximo 10*) o bien sensibilidad del tipo incremental según corresponda.

		0	1	2	3	4	5
		Modo 1	Modo 1 & Pinkie	Modo 2	Modo 2 & Pinkie	Modo 3	Modo 3 & Pinkie
0	Acelerador	UCHAR	UCHAR	UCHAR	UCHAR	UCHAR	UCHAR
1	Pedales	UCHAR	UCHAR	UCHAR	UCHAR	UCHAR	UCHAR
2	Rotatorio 1	UCHAR	UCHAR	UCHAR	UCHAR	UCHAR	UCHAR
3	Rotatorio 2	UCHAR	UCHAR	UCHAR	UCHAR	UCHAR	UCHAR

Llamadas al "X36 NT Driver service"

Para llamar a la funciones del servicio de Windows se usa la siguiente función del API de mensajes de windows:

```
BroadcastSystemMessage
(
  BSF_IGNORECURRENTTASK|BSF_POSTMESSAGE,
  NULL,
  RegisterWindowMessage("X36ServiceCall"),
  llamada,
  datos
);
```

Las llamadas posibles son tres:

```
#define SVC_PAUSA 0
#define SVC_SENSIBILIDAD 1
#define SVC_CARGAR_INI 2
```

- **svc_pause** cambia la tasa de refresco de la emulación del ratón. *datos = valor de refresco (0 a 255)*
- **svc_sensibilidad** cambia la precisión del movimiento del ratón. *datos = valor de precisión (0 a 255)*
- **svc_cargar_ini** almacena los 11 bytes de inicialización del joystick (uno por llamada) para cargarlos automáticamente después de una suspensión del sistema. *datos = v[n]+(n*0x100) con v[n] = valor del byte n*

Posiciones e IDs de los botones

La tabla siguiente indica qué botón, eje, o rotatorio está asignado a cada posición para distintas llamadas. También se indica cual es el ID del botón para usarlo en la inicialización de los joysticks.

Botón, eje, rotatorio	Pos.	ID
Gatillo	0	13
Lanzar	1	4
Botón A	2	2
Botón B	3	3
Botón C	4	1
Pinkie	5	14
Seta 1 arriba	6	5
Seta 1 abajo	7	7
Seta 1 izquierda	8	8
Seta 1 derecha	9	6
Seta 1 arriba−izquierda	10	
Seta 1 arriba−derecha	11	
Seta 1 abajo−izquierda	12	
Seta 1 abajo−derecha	13	
Seta 2 arriba	14	9
Seta 2 abajo	15	11
Seta 2 izquierda	16	12
Seta 2 derecha	17	10
Seta 2 arriba−izquierda	18	
Seta 2 arriba−derecha	19	
Seta 2 abajo−izquierda	20	
Seta 2 abajo−derecha	21	
Botón D	22	15
Botón ratón	23	16
Modo izquierda	24	
Modo derecha	25	
Auxiliar izquierda	26	
Auxiliar derecha	27	
Seta ratón arriba	28	21
Seta ratón abajo	29	23
Seta ratón izquierda	30	24
Seta ratón derecha	31	22
Seta 3 arriba	32	17
Seta 3 abajo	33	19
Seta 3 izquierda	34	20
Seta 3 derecha	35	18

Botón, eje, rotatorio	Pos.	ID
[1]Pulsaciones al soltar el botón (el orden de los botones es el mismo que el de la tabla de al lado)	36−71	
[2]Acelerador	72−77	
[2]Pedales	78−83	
[2]Rotatorio 1	84−93	
[2]Rotatorio 2	94−103	

[1]*Por ejemplo en la posión 36 está el botón "Gatillo soltado", en la 37 "Lanzar soltado"...*

[2]*Para los ejes y rotatorios, cuando están en modo incremental la primera posición es la posición creciente y la segunda la decrecienciente. Por ejemplo, en el acelerador la 72 es la posición creciente y la 73 la decreciente.*

X36−X35 NT Driver

Documentación para programadores

CAPITULO III

Formato de los arhivos .XMP

El formato actual de los archivos .XMP usados por el X36Map es el siguiente :

- ***WORD*** (2 bytes) − Número de comandos en el archivo.
- ***Comandos***
  - ***STRING*** (32 bytes) − Nombre.
  - ***UINT64*** (8 bytes) − Scancodes (1 byte por scancode).
  - ***UCHAR*** (1 byte) − Teclas extendidas (1 bit por scancode).
- ***Datos de inicialización***
  - ***6 UCHAR*** (6 bytes) − Botones del puerto.
  - ***UCHAR*** (1 byte) − POV 1.
  - ***UCHAR*** (1 byte) − POV 2.
  - ***UCHAR*** (1 byte) − Configuración acelerador y pedales.
  - ***UCHAR*** (1 byte) − Selector de modo y modo pinkie.
    - *bit 8* : 1 = modos On; 0 = modos Off
    - *bit 7* : 1 = modo pinkie On; 0 = modo pinkie Off
    - *bit 1−6* : No usados
- ***6 UCHAR*** (6 bytes) − Tipo de rotación para el acelerador.
- ***6 UCHAR*** (6 bytes) − Tipo de rotación para los pedales.
- ***6 UCHAR*** (6 bytes) − Tipo de rotación para el rotatorio 1.
- ***6 UCHAR*** (6 bytes) − Tipo de rotación para el rotatorio 2.
- ***6*104 WORD*** (1248 bytes) − Indices. Cada WORD indica el número del comando que hay que usar para cada botón, eje o rotatorio
- ***6*36 BOOLEAN*** (216 bytes) − Auto repeticiones. El formato es coherente con las diferentes llamadas IOCTL así que se pueden usar la información de los capítulos anteriores para tener más detalles sobre cada bloque.

*El formato es coherente con las diferentes llamadas así que los datos de cada bloque siguen el mismo sistema que las llamadas. Por ejemplo 6*36 BOOLEAN son 6 bloques (uno por modo) de 36 bytes correspondientes a las 36 posiciones que aparecen en la tabla del final del capítulo II.*

X36−X35 NT Driver

Documentación para programadores

APENDICE A

Códigos enviados por los joysticks

Botones	Apretar					Soltar			
	Byte 1	Byte 2	Byte 3	Byte 4		Byte 1	Byte 2	Byte 3	Byte 4
Gatillo	111	16	1	13		111	16	1	77
Lanzar	111	16	1	4		111	16	1	68
Botón A	111	16	1	2		111	16	1	66
Botón B	111	16	1	3		111	16	1	67
Botón C	111	16	1	1		111	16	1	65
Pinkie	111	16	1	14		111	16	1	78
Seta 1 arriba	111	16	1	5		111	16	1	69
Seta 1 abajo	111	16	1	7		111	16	1	71
Seta 1 izquierda	111	16	1	8		111	16	1	72
Seta 1 derecha	111	16	1	6		111	16	1	70
Seta 2 arriba	111	16	1	9		111	16	1	73
Seta 2 abajo	111	16	1	11		111	16	1	75
Seta 2 izquierda	111	16	1	12		111	16	1	76
Seta 2 derecha	111	16	1	10		111	16	1	74
Botón D	111	16	1	15		111	16	1	79
Botón ratón	111	16	1	16		111	16	1	80
Modo izquierda	111	16	1	32		111	16	1	96
Modo derecha	111	16	1	31		111	16	1	95
Auxiliar izquierda	111	16	1	30		111	16	1	94
Auxiliar derecha	111	16	1	29		111	16	1	93
Seta ratón arriba	111	16	1	21		111	16	1	85
Seta ratón abajo	111	16	1	23		111	16	1	87
Seta ratón izquierda	111	16	1	24		111	16	1	88
Seta ratón derecha	111	16	1	22		111	16	1	86
Seta 3 arriba	111	16	1	18		111	16	1	82
Seta 3 abajo	111	16	1	20		111	16	1	84
Seta 3 izquierda	111	16	1	17		111	16	1	81
Seta 3 derecha	111	16	1	19		111	16	1	83