



**Convergence
Instruments**

VSEW_mk4

Protocolo COM

28 de enero de 2023

Bruno Paillard

1	INTRODUCCIÓN	2
2	ENUMERACIÓN DE PUERTOS COM	2
3	CONFIGURACIÓN DEL PUERTO COM	2
4	ESTRUCTURA DE COMUNICACIÓN	2
5	ENDIANZA	2
6	TIPOS BÁSICOS	2
7	ESTRUCTURA DE PAQUETES	3
7.1	Paquete de comandos	3
7.2	Paquete de datos	4
7.3	Reconocer	4
7.4	Comandos	4
8	PERSISTENCIA DE DATOS	9

1 Introducción

El *VSEW_mk4* es un nuevo modelo de la serie *VSEW* que introduce un protocolo de comunicación abierto de puerto virtual Com. Eso significa que el instrumento se puede utilizar en cualquier plataforma que tenga un controlador genérico para soportar la clase CDC (Comunicación) USB. Actualmente la mayoría de las plataformas soportan esa clase, incluyendo Windows, Mac y Linux.

Ese puerto Com virtual abierto sólo admite un subconjunto de los comandos del instrumento. Todos los comandos relacionados con el almacenamiento en memoria, WiFi y alertas, y los ajustes del instrumento no pueden aplicarse a través de esa interfaz. Sin embargo, esa interfaz permite la lectura de todos los ajustes, las señales de aceleración en bruto, así como los niveles RMS medidos en tiempo real.

Esa interfaz permite a los desarrolladores diseñar su propia aplicación compatible con el instrumento.

2 Enumeración de puertos Com

Cuando el PC host enumera el instrumento, una de las interfaces que presenta es un puerto Com virtual (un dispositivo USB CDC-Class). A partir de Windows 10, el controlador genérico de puerto COM de Windows se instanciará automáticamente y se vinculará a esa interfaz. En Windows 7 y 8, aunque Microsoft proporciona un controlador genérico, el usuario debe cargar manualmente el controlador cuando el dispositivo se conecta al PC por primera vez. Una vez cargado el controlador, aparece un nuevo puerto Com en la lista de dispositivos conectados al PC.

3 Configuración del puerto Com

El puerto Com puede configurarse (velocidad de bits, número de bits de parada... etc.), bien mediante los controles del *Administrador de dispositivos de Windows*, bien en una aplicación utilizando las funciones API adecuadas. Sin embargo, esta configuración **no** afecta a la comunicación real. Sólo se exponen en la interfaz por motivos de compatibilidad. A nivel de hardware no hay ninguna línea serie física presente, y la velocidad de comunicación final sólo está determinada por el rendimiento del enlace USB. Este rendimiento suele rondar los 3 Mbps cuando no hay otros dispositivos en el bus USB.

4 Estructura de la comunicación

Los intercambios entre el PC anfitrión y el instrumento siguen siempre un modelo Maestro-Eslavo. El host inicia un intercambio utilizando un *Paquete de Comandos*. El host también puede enviar datos a continuación de ese *Paquete de Comandos*. El instrumento responde con datos o con un byte Ack si no hay datos que transmitir al host.

En todos los casos, después de enviar un comando, el PC anfitrión no debe enviar otro comando antes de que el instrumento envíe una respuesta de vuelta. Esa respuesta puede ser datos o puede ser un Ack si no hay datos solicitados por el comando.

5 Endianidad

Salvo que se indique lo contrario, el endianés es Little-Endian.

6 Tipos básicos

En este protocolo pueden utilizarse los siguientes tipos básicos:

Tipo Nombre	Descripción	Endianidad
U8	Un byte sin signo	N/A
U16	Palabra de 16 bits sin signo	Pequeño indio
U32	Palabra de 32 bits sin signo	Pequeño indio
U64	Palabra de 64 bits sin signo	Pequeño indio
I8	Un byte con signo	N/A
I16	Palabra de 16 bits con signo	Pequeño indio
I32	Palabra de 32 bits con signo	Pequeño indio
I64	Palabra de 64 bits con signo	Pequeño indio
Sgl	Palabra de 32 bits en formato de coma flotante IEEE 754	Pequeño indio
Doble	Palabra de 64 bits en formato de coma flotante IEEE 754	Pequeño indio
Cadena	Las cadenas son concatenaciones de caracteres ASCII de 8 bits, terminadas por un byte de fin de texto (0x00).	N/A

Cuadro 1

7 Estructura del paquete

7.1 Paquete de comandos

El *paquete de comandos* está estructurado de la siguiente manera:

Campo	Tamaño (bytes)	Función
Comando	4	<p>El comando indica los datos transmitidos o la operación realizada. La dirección de transmisión indicada está centrada en el host.</p> <p>El bit 31 de la palabra de orden indica el sentido de la transferencia:</p> <ul style="list-style-type: none"> • 0: OUT (host a dispositivo) • 1: IN (Dispositivo a host)
Dirección	4	<p>La función del campo de dirección varía según el comando.</p> <p>Para muchos comandos este número no es aplicable (N/A) y puede establecerse en cualquier valor.</p>
Cuenta	4	<p>Este campo indica el número de elementos que se transferirán en el siguiente paquete de datos (ya sea una ENTRADA o una SALIDA). El comando define cómo se interpretan los bytes devueltos.</p>

		Para muchos comandos este número no es aplicable N/A y puede fijarse en cualquier valor.
--	--	--

Cuadro 2

7.2 Paquete de datos

Los *paquetes de datos* son simplemente una concatenación de bytes. La forma en que se interpretan los bytes depende del comando que precede al *paquete de datos*.

7.3 Acuse recibo

El *Ack* es un único byte con valor 0x06. El byte *Ack* sólo se envía de vuelta al host si el comando es de *Escritura*, y por lo tanto no requiere una respuesta de datos por parte del dispositivo. Cuando el comando es de *lectura*, los datos enviados al host sirven para ese propósito y el instrumento no envía ningún byte *Ack*.

7.4 Comandos

Comando	Descripción	Dirección	Cuenta	Bytes devueltos	Datos/Ack
0x80000010	<p><i>Leer_Amplitud_RMS</i></p> <p>Este comando recupera la amplitud RMS actual. Se trata de una amplitud utilizando la constante de tiempo establecida para el instrumento.</p> <p>La escala de la señal es m/s^2 para la aceleración o m/s para la velocidad.</p> <p><i>Nota: Los valores tienen en cuenta la aceleración estática de la gravedad, por lo que, para eliminar el impacto de ese componente estático en los resultados, debe ajustarse el filtro de paso alto.</i></p>	N/A	N/A	12	<p>Datos:</p> <p>3 IEEE-754 de 32 bits Flotantes que representan la amplitud RMS en los ejes X, Y y Z.</p> <p>Ack: No</p>
0x80000012	<p><i>Leer_temperatura</i></p> <p>Este comando recupera la temperatura.</p>	N/A	N/A	4	<p>Datos:</p> <p>32 bits IEEE-754 Float que representa la temperatura en grados C</p> <p>Ack: No</p>

0x80000013	<i>Leer_Batería</i>	N/A	N/A	4	Datos: 32 bits IEEE-754 Flotador que representa
	Este comando recupera el voltaje de la batería.				el voltaje de la batería en voltios Ack: No
0x80000020	<i>Leer_tipo_señal</i>	N/A	N/A	1	Datos: 1 byte que representa la curva de ponderación: 0: Aceleración 1: Velocidad Ack: No
0x80000021	<i>Leer_FS</i>	N/A	N/A	2	Datos: U16 que representa la frecuencia de muestreo en Hz. Ack: No
0x80000022	<i>Leer_Tau</i>	N/A	N/A	4	Datos: 32 bits IEEE-754 Float que representa la constante de tiempo en s. Ack: No
0x80000023	<i>Leer_Paso_alto</i>	N/A	N/A	5	Datos: 1 IEEE-754 de 32 bits Flotante que representa la frecuencia de paso alto en Hz. 1 byte que representa el estado del filtro: 0: desactivado 1: Encendido Ack: No

0x80000024	<i>Read_LowPass</i> Este comando lee el ajuste actual del filtro pasa-bajos. El instrumento devuelve la frecuencia de corte en Hz, seguido de un único byte	N/A	N/A	5	Datos: 1 IEEE-754 de 32 bits Flotante que representa la frecuencia de paso bajo en Hz.
	que indica si el filtro está activado o desactivado.				1 byte que representa el estado del filtro: 0: Apagado 1: En Ack: No
0x80000025	<i>Leer_KB</i> Este comando lee la configuración actual del filtro KB. El comando devuelve un único byte que indica si el filtro KB está activado o desactivado.	N/A	N/A	5	Datos: 1 byte que representa el estado del filtro: 0: Apagado 1: En Ack: No
0x80000031	<i>Leer_Modelo</i> Este comando lee el modelo del instrumento.	N/A	0-32	0-32	Datos: ASCII cadena que representa el Modelo. Tamaño: Hasta 32 bytes, incluyendo el byte de terminación. Ack: No
0x80000032	<i>Leer_SN</i> Lee el número de serie del instrumento	N/A	0-32	0-32	Datos: ASCII cadena que representa el número de serie del instrumento. Tamaño: Hasta 32 bytes, incluyendo el byte de terminación Ack: No

0x80000033	<p><i>Leer_FW_Rev</i></p> <p>Este comando lee el número de revisión del firmware.</p>	N/A	0-32	0-32	<p>Datos:</p> <p>ASCII cadena que representa la revisión del Firmware.</p> <p>Tamaño: Hasta 32 bytes, incluyendo el byte de terminación</p> <p>Ack: No</p>
0x80000034	<p><i>Leer_DOC</i></p> <p>Este comando lee la fecha de la última calibración.</p> <p>El UTC representa el número de segundos transcurridos desde el 1 de enero de 1904.</p>	N/A	N/A	8	<p>Datos:</p> <p>U64 número que representa el UTC (Código de Tiempo Universal) de la fecha/hora de la última calibración.</p> <p>Ack: No</p>
0x80000035	<p><i>Leer_DOB</i></p> <p>Este comando lee la fecha de nacimiento del instrumento.</p> <p>El UTC representa el número de segundos transcurridos desde el 1 de enero de 1904.</p>	N/A	N/A	8	<p>Datos:</p> <p>U64 número que representa el UTC (Código de Tiempo Universal) de la fecha/hora de nacimiento.</p> <p>Ack: No</p>
0x80000036	<p><i>Leer_ID_Usuario</i></p> <p>Este comando lee el campo User_ID. Ese campo se puede escribir en la memoria persistente utilizando el comando <i>Write_User_ID</i>.</p>	N/A	0-32	0-32	<p>Datos:</p> <p>ASCII cadena que representa el User-ID, definido por el usuario.</p> <p>Tamaño: Hasta 32 bytes, incluyendo el byte de terminación</p> <p>Ack: No</p>

0x00000036	<p><i>Escribir_ID_Usuario</i></p> <p>Este comando escribe el campo User_ID en la memoria persistente.</p>	N/A	0-32	1	<p>Datos:</p> <p>ASCII cadena que representa el ID de usuario, definido por el usuario.</p> <p>Tamaño: Hasta 32 bytes, incluyendo el byte de terminación</p> <p>Ack: Sí</p>
0x80000050	<p><i>Leer_señal</i></p> <p>Este comando lee el contenido de la señal FIFO.</p> <p>El comando solicita N muestras. El instrumento devuelve el número N de muestras. Cada muestra es un</p>	N/A	N	$(N \times 3) + 4$	<p>Datos:</p> <p>1 U32 que representa el número N de muestras (tripletas) que sigue.</p> <p>$N \times 3$ 32-bit IEEE-754 Floats, para X, Y</p>

	<p>triplete de valores flotantes, un valor para cada eje. La respuesta sólo contendrá el número de muestras realmente presentes en la FIFO de señales. Así, $\hat{N} \leq N$.</p> <p>La respuesta es:</p> <ul style="list-style-type: none"> - Un valor U32 que representa el número \hat{N} de muestras realmente devueltas. - El número especificado \hat{N} de tripletas sigue a ese valor. Cada miembro de una tripleta es un valor <i>Sgl float</i>. <p>La escala de la señal es m/s^2 para la aceleración o m/s para la velocidad.</p> <p>El comando puede devolver un máximo de 256 muestras (tripletras), por lo que no es útil solicitar más de 256 muestras.</p> <p>Nota. El FIFO puede contener internamente un máximo de 1024 muestras (tripletras). Ese FIFO suele llenarse continuamente por el procesamiento interno a la frecuencia de muestreo. Por lo tanto, cuando se envíe este comando las primeras veces, devolverá datos obsoletos. Es necesario solicitar al menos 1024 muestras para llegar a datos recién medidos. El comando debe enviarse repetidamente a una velocidad suficientemente alta en relación con la frecuencia de muestreo para evitar que falten datos.</p>			<p>y Z. Los 3 ejes se intercalan en la respuesta, empezando por el eje X.</p> <p>Ack: No</p>
--	--	--	--	--

Cuadro 3