UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Técnicas Digitales II - R4054

Informe de protocolo USB

Estudiante: Lugano Damian – *Leg.:* 1756990
Docente: Furfaro Alejandro
Ayudante: Ferreyro Luciano

Fecha de realización: 26/10/23
Fecha de entrega de informe:

1 Objetivos

El objetivo del presente informe es conectar un dispositivo a un puerto del sistema para capturar y analizar los paquetes de configuración del protocolo USB que se intercambian entre el dispositivo y el Host.

2 Análisis

Paso 1: Conexión y reconocimiento

Se conecta un pendrive a un puerto USB del sistema. El llamado Host o Hub detecta al device mediante la detección del cambio en la impedancia de la línea.
Una vez detectado, se considera al device en estado *Attached* y el Host le provee una alimentación inicial de 100mA. En este punto el device se encuentra en estado *Powered.* En caso de que el device sea self-powered comienza directamente en estado *Powered.*
Cada hub utiliza un endpoint de interrupción para reportar eventos al host. Dicho reporte indica qué puerto del hub experimentó algún evento. En este caso se observa que el addres del device es 1.1.0 , esto quiere decir que es el device 1 configurado en el endpoint 0.

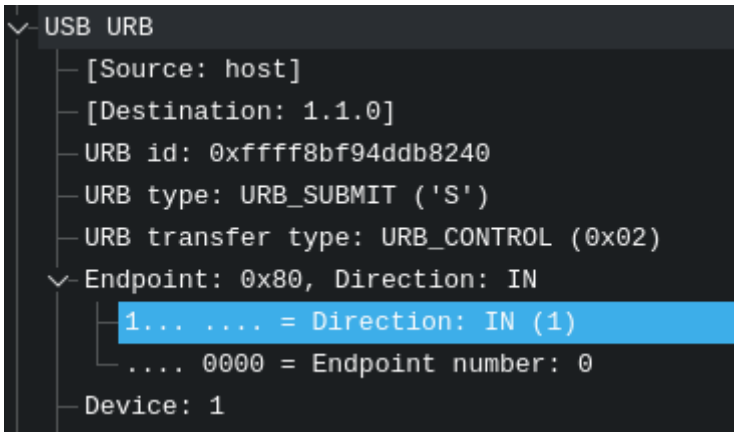


Figura 2.1 - Device Address

Luego, el Hub envía un Get Status Request para determinar si un nuevo device fue conectado a dicho puerto. Además, el hub detectará la velocidad del device al monitorear la tensión en las líneas, la cual se informará en el próximo Get Status Request enviado por el Host.

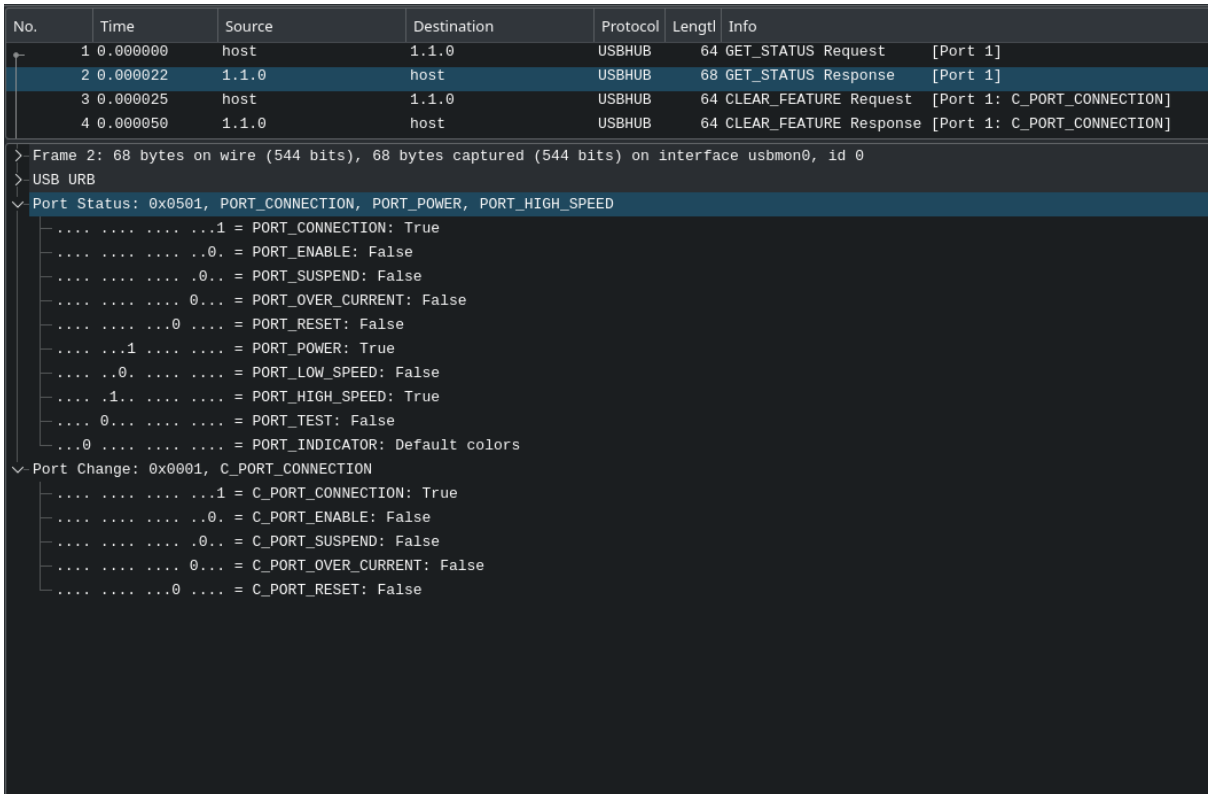


Figura 2.2 - Get Port Status

En la Figura 2.2 se puede observar la respuesta del Get Status, donde se encuentran las configuraciones iniciales como PORT_CONNECTION, PORT_POWER y PORT_HIGH_SPEED.

Paso 2: El hub resetea al device

El host envía al hub un Set Port Feature Request en el cual solicita enviar la condición de reset. El device acepta y se reinicia la máquina de estados del dispositivo.

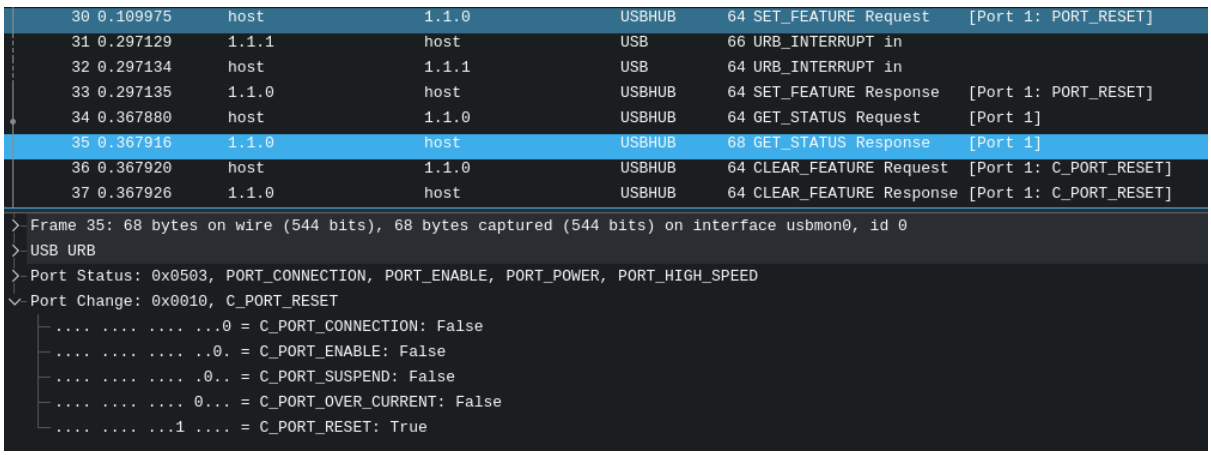


Figura 2.3 - Set Feature Request (Reset)

Al salir del estado de reset, el device está listo para responder a los request del host. Su dirección será por default 0x00 hasta que se le asigne una nueva.

Paso 3: El host envía un Get Descriptor Request solicitando el Device Descriptor

Arranca la transferencia de control.

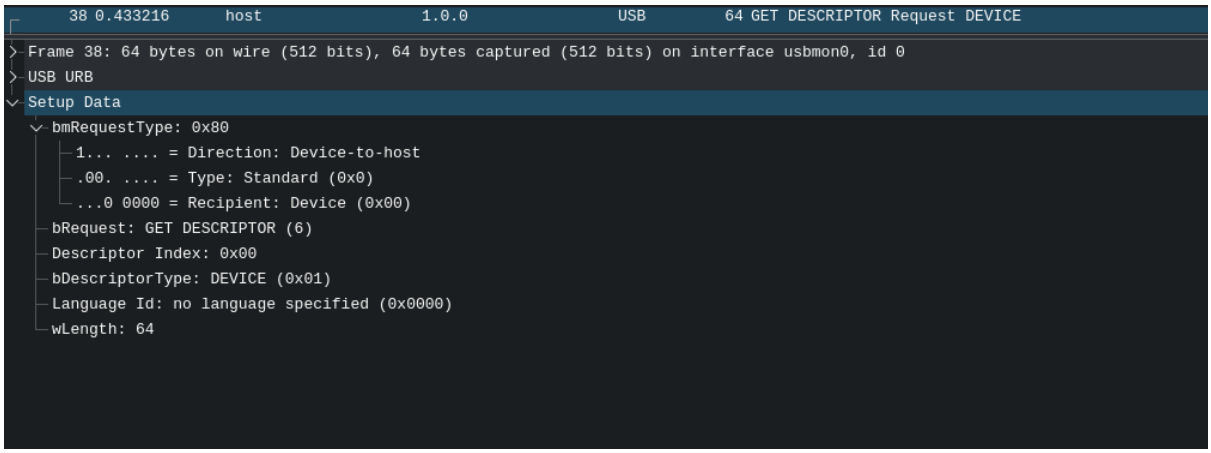


Figura 2.4 - Get Device Descriptor Request

En la figura se puede observar que el Get Descriptor al device de index 0x00

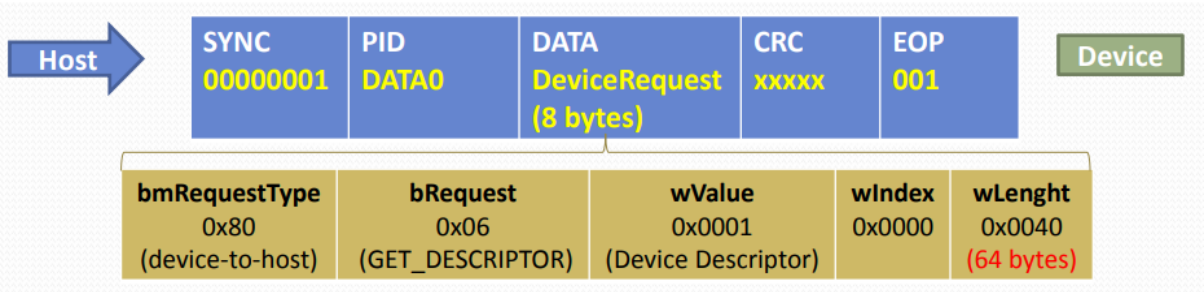


Figura 2.5 - Trama USB del Device Descriptor

La respuesta obtenida del device es la siguiente:

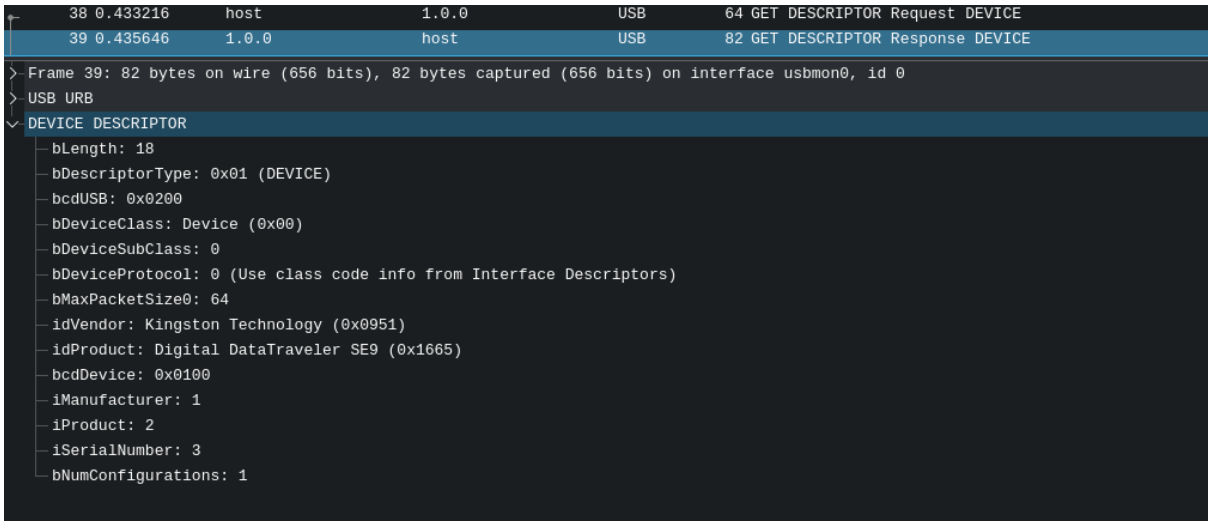


Figura 2.6 - Get Device Descriptor Response

Una vez que el host conoce el tamaño maximo de datos, en este caso bMaxPacketSize=64, genera un estado de reset.

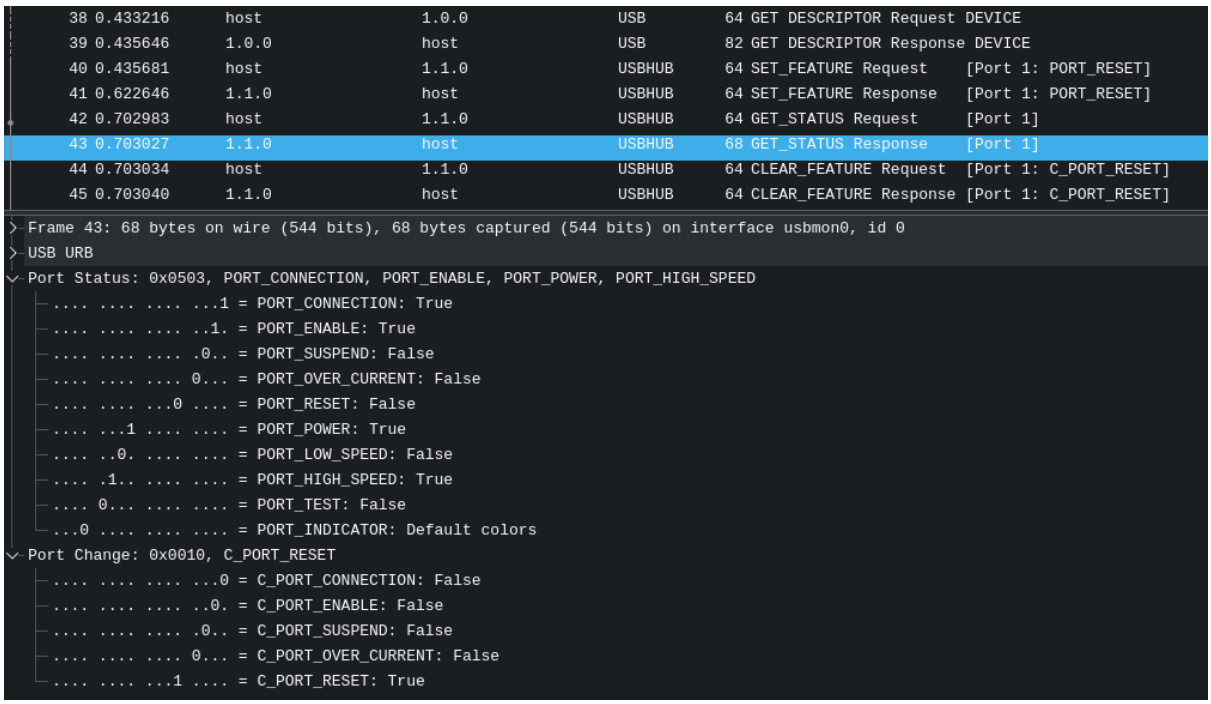


Figura 2.7 - Get Port Status Response

En la figura se puede observar, en la última linea, que se hace un acknowledge del Reset.

Paso 4: El host asigna una dirección al device

Esto se hace mediante un Set Address Request.

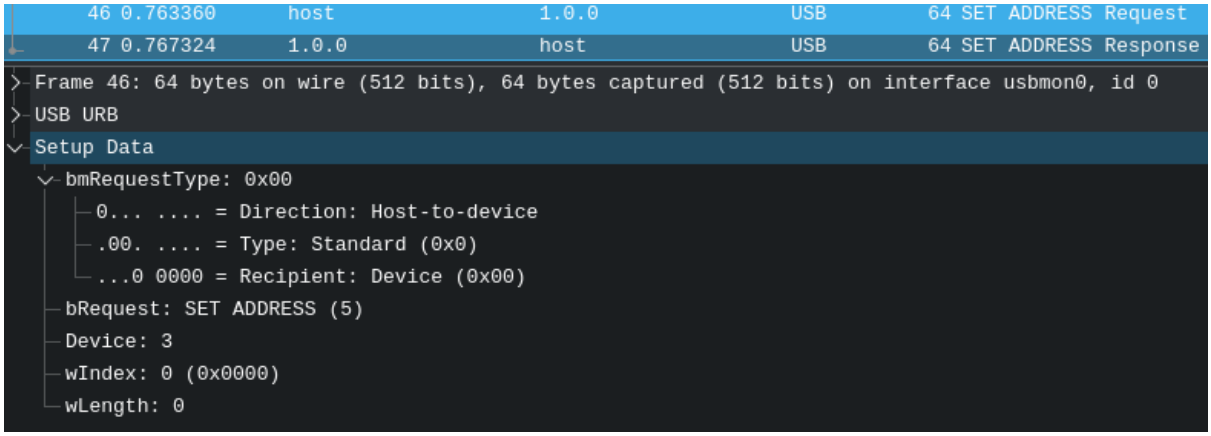


Figura 2.8 - Set Address Request

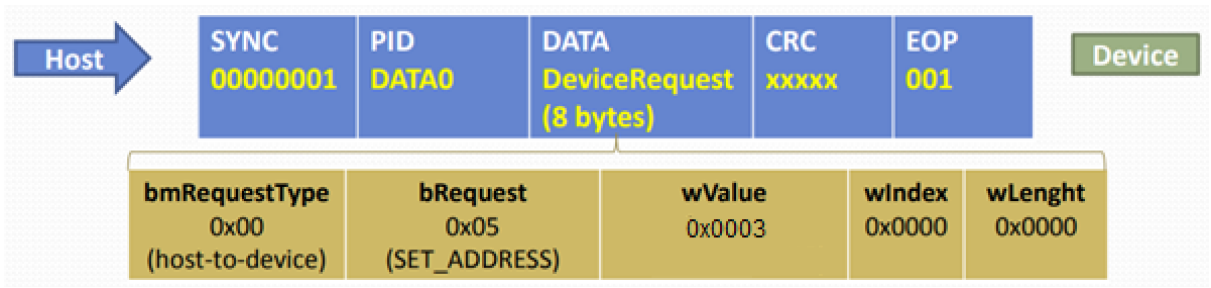


Figura 2.9 - Trama de Set Address Request

En este caso se asignó el address 3 al device. Por lo que a partir de ahora, las transferencias de paquetes se harán a la dirección 3 y se mantendrá asi hasta que se desconecte el dispositivo o se reinicie el sistema. El device pasa a modo *Addressed*.

Paso 5: El host solicita todos los descriptores

Un Get Descriptor Request del descriptor de configuración, implica enviar todos sus descriptores subordinados. En este punto un host Windows suele solicitar los primeros nueve bytes de todo el conjunto (lo que corresponde al descriptor de configuración propiamente dicho) para conocer la longitud total. Luego reenvía el request para que el device responda con todos los datos: Configuration + Interface(s) + Endpoint(s).

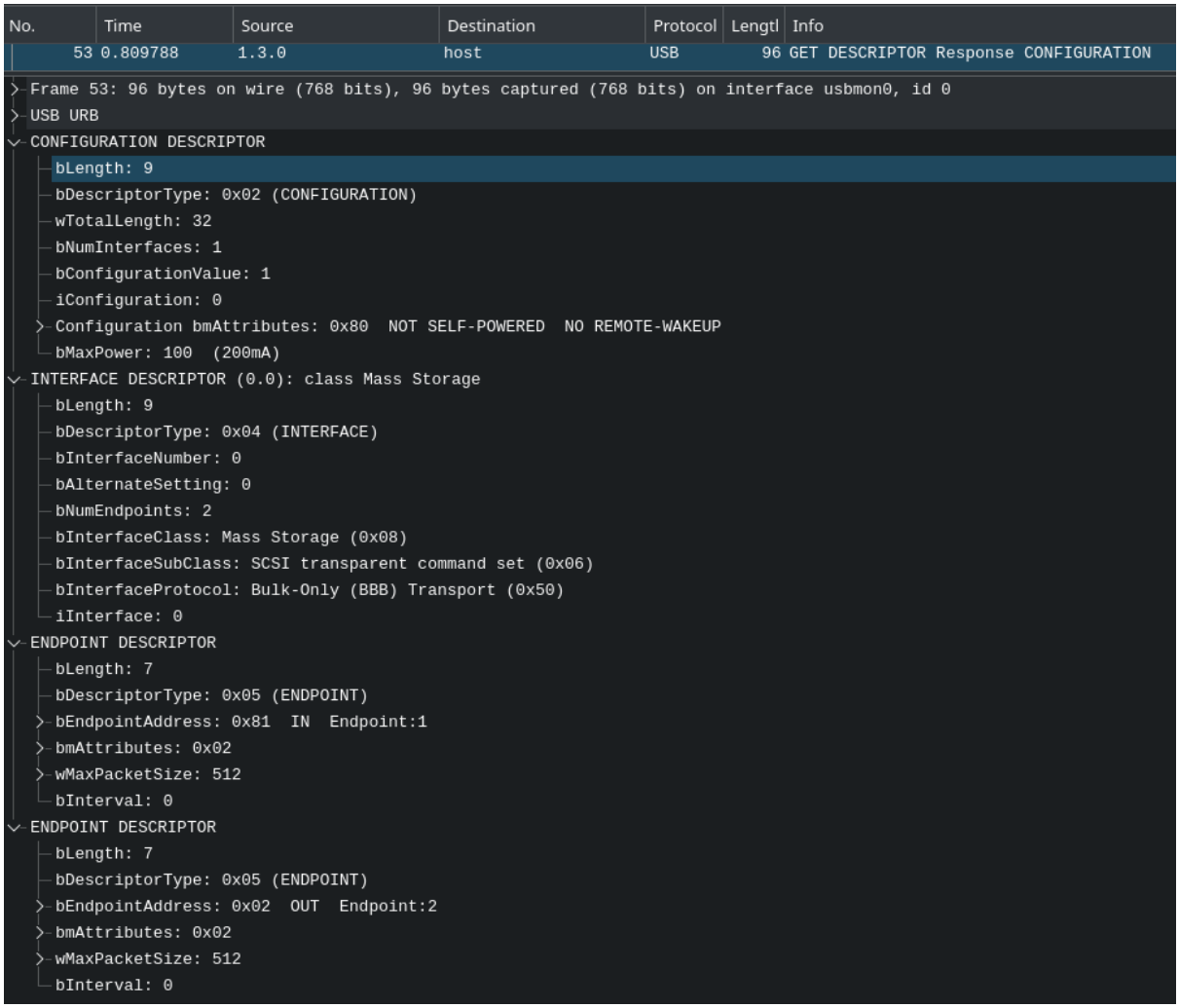


Figura 2.10 - Get Configuration Descriptor Response

Del Configuration Descriptor se pueden destacar algunas configuraciones:
Configuration Descriptor:

- Configuration bmAttributes: 0x80 NOT SELF-POWERED NO REMOTE-WAKEUP
    - bMaxPower: 100(200mA)

Interface Descriptor : class Mass Storage (porque es un pendrive)

- bNumEndpoints : 2 => utilizará 2 endpoints
- bInterfaceClass : Mass Storage => Es un dispositivo de almacenamiento
- bInterfaceProtoco : Bulk-Only => Solo admite transferencias Bulk como es esperado de un pendrive.

Endpoint Descriptor: En este caso hay 2 endpoint descriptor

1. bEndpointAddress : 0x81 IN Endpoint:1
2. bEndpointAddress : 0x02 OUT Endpoint:2

Ambos con wMaxPacketSize : 512
Esto quiere decir que hay 2 endpoint para el dispositivo, uno para recibir paquetes y otro para enviarlos.

Paso 6: El host asigna y carga el device driver

A partir de los campos de Vendor ID, Product ID y Clases de los distintos descriptores, el sistema operativo decide qué driver es el más adecuado para el device detectado.
Por lo tanto, en este punto se producen una serie de transferencias de descriptores tipo STRING con datos como el Vendor o el producto.

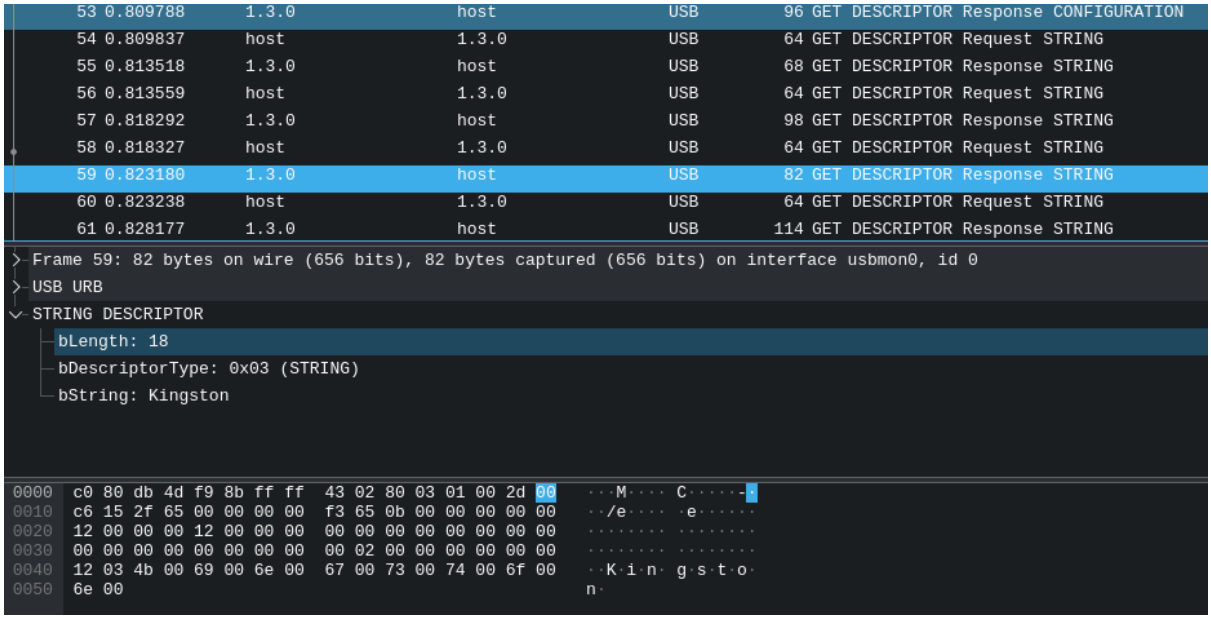


Figura 2.11 - Get String Descriptor Response

Paso 7: El device driver selecciona una configuración.

El host envía al device un Set Configuration Request junto con el índice de la configuración deseada para establecerla como activa.
Una vez recibido el request, el device adopta la configuración solicitada y pasa al estado *Configured*.

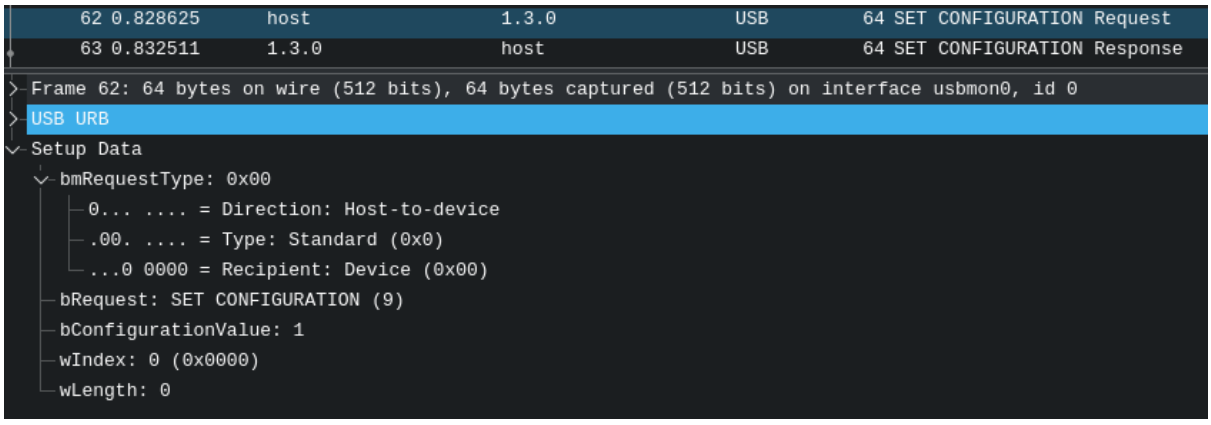
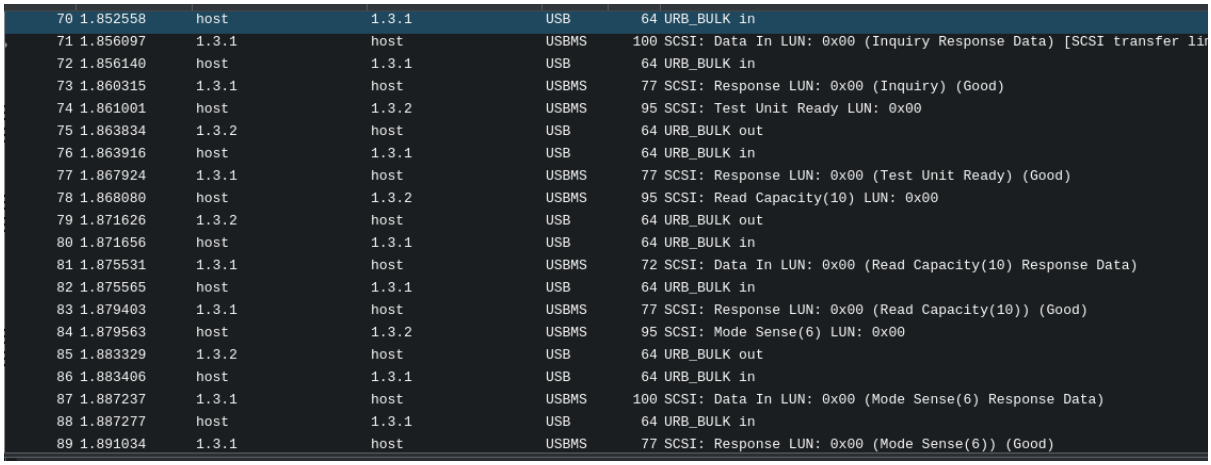


Figura 2.12 - Set Configuration Request

El device ahora se considera listo para enviar y recibir paquetes de datos a los endpoints configurados. En la figura se observan los BULK_in y los BULK_out que representan dichos paquetes de datos.



70	1.852558	host	1.3.1	USB	64	URB_BULK in
71	1.856097	1.3.1	host	USBMS	100	SCSI: Data In LUN: 0x00 (Inquiry Response Data) [SCSI transfer lir
72	1.856140	host	1.3.1	USB	64	URB_BULK in
73	1.860315	1.3.1	host	USBMS	77	SCSI: Response LUN: 0x00 (Inquiry) (Good)
74	1.861001	host	1.3.2	USBMS	95	SCSI: Test Unit Ready LUN: 0x00
75	1.863834	1.3.2	host	USB	64	URB_BULK out
76	1.863916	host	1.3.1	USB	64	URB_BULK in
77	1.867924	1.3.1	host	USBMS	77	SCSI: Response LUN: 0x00 (Test Unit Ready) (Good)
78	1.868080	host	1.3.2	USBMS	95	SCSI: Read Capacity(10) LUN: 0x00
79	1.871626	1.3.2	host	USB	64	URB_BULK out
80	1.871656	host	1.3.1	USB	64	URB_BULK in
81	1.875531	1.3.1	host	USBMS	72	SCSI: Data In LUN: 0x00 (Read Capacity(10) Response Data)
82	1.875565	host	1.3.1	USB	64	URB_BULK in
83	1.879403	1.3.1	host	USBMS	77	SCSI: Response LUN: 0x00 (Read Capacity(10)) (Good)
84	1.879563	host	1.3.2	USBMS	95	SCSI: Mode Sense(6) LUN: 0x00
85	1.883329	1.3.2	host	USB	64	URB_BULK out
86	1.883406	host	1.3.1	USB	64	URB_BULK in
87	1.887237	1.3.1	host	USBMS	100	SCSI: Data In LUN: 0x00 (Mode Sense(6) Response Data)
88	1.887277	host	1.3.1	USB	64	URB_BULK in
89	1.891034	1.3.1	host	USBMS	77	SCSI: Response LUN: 0x00 (Mode Sense(6)) (Good)

Figura 2.13 - URB_BULK in and out

3 Anexo

Para el desarrollo del presente informe se utilizaron las siguientes fuentes de información:

- Peacock, C. (2002) USB in a nutshell - Making Sense of the USB Standard. Disponible en: https://sge.frba.utn.edu.ar/upload/Materias/95-0435/archivos/usb-in-a-nutshell.pdf (Accedido: 26 de octubre del 2023).

- Ing. Ridolfi, P. (2011) Universal Serial Bus – Técnicas Digitales II - UTN FRH Disponible en: https://gitlab.frba.utn.edu.ar/td_piloto/td2_r4054/publico/-/blob/master/slides/Clase14-2020-11-03-USB/Universal%20Serial%20Bus.pdf?ref_type=heads (Accedido: 26 de octubre del 2023)