

Dispositivo virtual

Haciendo uso de el ejemplo uhid-example, este lo puedes encontrar aquí, con este código te muestran como es posible simular un mouse con tres botones y un teclado.

El motivo del por que estoy usando esto, es por que aún esta en desarrollo la PCB, además de conocer como funciona el descriptor HID, que en grandes rasgos le dice las caracteristas de un dispositivo y el manejo que le debe de dar a los datos, tanto de entrada como salida.

El código al compilarse y ejecutarlo como **root** crea un “dispositivo virtual” que al presionar las teclas “WASD” mueve el puntero un poco en la dirección de la tecla correspondiente, te recomiendo que veas el link de la parte de arriba para mas información.

Para presentar este dispositivo virtual como un joystick es necesario un cambio

```
static unsigned char rdesc[] = {
    0x05, 0x01, /* USAGE_PAGE (Generic Desktop) */
    0x09, 0x02, /* USAGE (Mouse) */
    0xa1, 0x01, /* COLLECTION (Application) */

a

static unsigned char rdesc[] = {
    0x05, 0x01, /* USAGE_PAGE (Generic Desktop) */ //
    0x09, 0x04, /* USAGE (Mouse) */ De esta forma se le dice que es un joystick
    0xa1, 0x01, /* COLLECTION (Application) */
```

y si usamos una herramienta como “jstest-gtk” se ve que lo reconoce como un joystick

Nota: Le añadí previamente un nombre personalizado en este caso “Steering Wheel CAE32”, en tu caso vendra con el nombre de “test-uhid-device”

Código

Con esta previa introducción es hora de crear un joystick con las características de CAE32, estas son:

- 4 ejes, para pedales y volante
- 7 botones, para las marchas, 6 marchas y reversa, la neutral se toma como “0”

Notas

1. Para esta primera versión de este dispositivo, los ejes tienen una resolución de 1-byte, rango de 0 a 255 o -127 a 126, lo que puede ser poco para los estandares del momento.

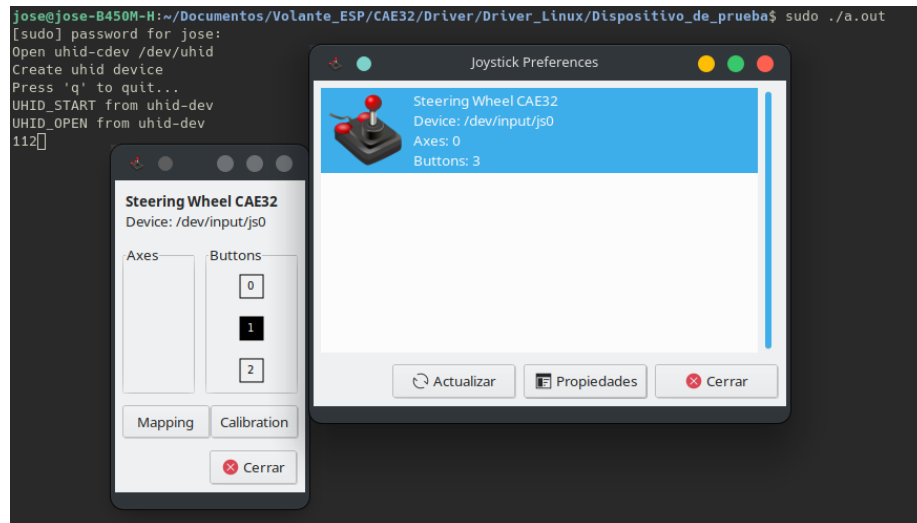


Figure 1: Funcionamiento hid

2. Por el momento no se implementa el force feedback (FFB).
3. Los números con *0xnn* son hexadecimales

Como se observa todos los tipos de datos son de entrada, por lo cual hay que echar un vistazo, para elegir como vamos a tratar los datos

- Para los pedales creo que la mejor opción es escoger el bit 4 de lineal o no lineal
- Para los botones se usara el bit 1, de matriz o variable

Al final de este apartado agregaré como queda el código

Ahora para definir el tamaño de los valores de entrada hay que usar la etiqueta “Report Size”

- Pedales=8 bits=0x08
- Botones=1 bit=0x01

Si existen varios datos de entradas similares, como son los botones y pedales se pueden declarar con la siguiente etiqueta “Report Count”

- Pedales=3 campos =0x03
- Botones=8 campos =0x08

Description	6.2.2.4 Main Items		
	Main items are used to either define or group certain types of data fields within a Report descriptor. There are two types of Main items: data and non-data. Data-type Main items are used to create a field within a report and include Input , Output , and Feature . Other items do not create fields and are subsequently referred to as non-data Main items.		
Parts	One-Byte Prefix (<i>nn</i> represents size value)		
	Main item tag	Valid Data	
Input	1000 00 <i>nn</i>	Bit 0	{Data (0) Constant (1)}
		Bit 1	{Array (0) Variable (1)}
		Bit 2	{Absolute (0) Relative (1)}
		Bit 3	{No Wrap (0) Wrap (1)}
		Bit 4	{Linear (0) Non Linear (1)}
		Bit 5	{Preferred State (0) No Preferred (1)}
		Bit 6	{No Null position (0) Null state(1)}
		Bit 7	Reserved (0)
		Bit 8	{Bit Field (0) Buffered Bytes (1)}
		Bit 31-9	Reserved (0)

Figure 2: input

Ya tenemos definido que tipo y la cantidad de datos de entrada, pero aún falta definir para que lo vamos a usar

La manera en que se usan los datos dependen de varias etiquetas como:

- “Usage Page”
- “Usage”
- “Logical Minimum”
- “Logical Maximum”
- “Usage Minimum”
- “Usage Maximum”

Nota: Existen mas tipos de etiquetas de uso, pero estas son las basicas.

Para definir que es un joystick hay que buscar en “las Usage Pages” y lo encontramos en “Generic Desktop Page” con un valor de **0x04**

Ahora hay que asignar el número de botones, este lo encontramos en la pagina de botones

Como se observa podemos definir un rango muy grande de botones para usar, pero en este caso solo en necesario 7 botones, para este primer prototipo

Para definir los Ejes, estos los podemos encontrar en la “Generic Desktop Page” como X,Y,Z,Rx,Ry,Rz

Te recomiendo que veas estos 2 Documentos para que logres entender mas en profundidad el porque de los valores ya que no abarque todos los temas, aunque

Generic Desktop Page (0x01)

Usage ID	Usage Name	Usage Types	Section
00	<i>Undefined</i>		
01	Pointer	CP	4.1
02	Mouse	CA	4.1
03-03	<i>Reserved</i>		
04	Joystick	CA	4.1

Figure 3: Usage Pages

12 Button Page (0x09)

The Button page is the first place an application should look for user selection controls. System graphical user interfaces typically employ a pointer and a set of hierarchical selectors to select, move and otherwise manipulate their environment. For these purposes the following assignment of significance can be applied to the Button usages:

- Button 1, Primary Button. Used for object selecting, dragging, and double click activation. On MacOS, this is the only button. Microsoft operating systems call this a logical left button, because it is not necessarily physically located on the left of the pointing device.
- Button 2, Secondary Button. Used by newer graphical user interfaces to browse object properties. Exposed by systems to applications that typically assign application-specific functionality.
- Button 3, Tertiary Button. Optional control. Exposed to applications, but seldom assigned functionality due to prevalence of two and one button devices.
- Buttons 4 – 255. As the button number increases, its significance as a selector decreases.

In many ways the assignment of button numbers is similar to the assignment of **Effort** in **Physical** descriptors. Button 1 would be used to define the button a finger rests on when the hand is in the *at rest* position, that is, virtually no effort is required by the user to activate the button. Button values increment as the finger has to stretch to reach a control. See Section 6.2.3, *Physical Descriptors* in the HID Specification for methods of further qualifying buttons.

Usage ID	Usage Name	Usage Type
00	No Button Pressed	See Note
01	Button 1 (primary/trigger)	See Note
02	Button 2 (secondary)	See Note
03	Button 3 (tertiary)	See Note
04	Button 4	See Note
...	...	
FFFF	Button 65535	See Note

Table 12.1: Button Page

Figure 4: botones

puede que en futuro los agrege aquí

Device Class Definition for Human Interface Devices (HID) Version1.11 HID
Usage Tables FOR Universal Serial Bus (USB)