

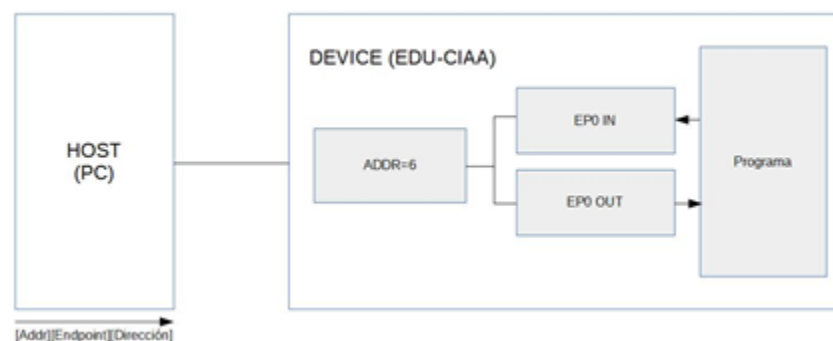
### Laboratorio N° 3: USB

Continuando con el desarrollo del sistema para control de turnos bancarios, le solicitaron la fabricación de una botonera con interfaz USB para interactuar con la pantalla de turnos.

La solución planteada debe emular un teclado, utilizando los botones de la CIAA(4 teclas) y reproduciendo los caracteres “q”, “w”, “e”, “r”.

En este laboratorio usted configurará el puerto USB-OTG.

#### Funcionamiento



Cada paquete enviado entre el Host y el Device tiene los campos:

- Addr: el host le asigna una dirección para identificar el dispositivo dentro del bus (se pueden conectar hasta 127 dispositivos).
- Endpoint: fuente o sumidero de datos. Puede haber hasta 16 endpoints por dispositivo.
- Dirección: se lo mira desde el Host (IN si el paquete ingresa al host y OUT si va hacia el dispositivo).

Usaremos el USB como HID (dispositivo de interfaz humana). Una de las características de estos dispositivos es la transferencia de paquetes por **interrupción**.

#### Desarrollo de aplicaciones USB

El micro de la EDU-CIAA posee en su ROM una arquitectura de programación llamada USB Device ROM Stack, que simplifica el desarrollo de aplicaciones CDC (Dispositivos de comunicación), HID (Dispositivos de Interfaz Humana), MSC (Almacenamiento masivo) y DFU (Actualización de Firmware de Dispositivo).

Esta arquitectura sigue los siguientes pasos para la instanciación de los dispositivos:

- Prerrequisitos: inicialización del micro, clocks, la API USB y los pines (VBUS, D+, D-, ID).
- Definición de los descriptores: un dispositivo USB provee información de sí mismo en estructuras llamadas “USB descriptors”. El formato que tienen los descriptores está predefinido en el estándar USB. En el ejemplo, está definidos en hid\_desc.c.

- Inicialización de USB ROM Device:

```
//Creamos una instancia de la API
USBD_API_INIT_PARAM_T usb_param;
USB_CORE_DESCS_T desc;
//Y seteamos los atributos de la instancia en cero
memset((void *) &usb_param, 0, sizeof(USBD_API_INIT_PARAM_T));
//Definimos el area de la pila donde guardará sus variables globales
usb_param.usb_reg_base = LPC_USB_BASE;
usb_param.mem_base = USB_STACK_MEM_BASE;
usb_param.mem_size = USB_STACK_MEM_SIZE;
//Reservamos espacio de memoria para los endpoints
usb_param.max_num_ep = 2;
//Definición de descriptors
desc.device_desc = (uint8_t *) USB_DeviceDescriptor;
desc.string_desc = (uint8_t *) USB_StringDescriptor;
desc.high_speed_desc = USB_HsConfigDescriptor;
desc.full_speed_desc = USB_FsConfigDescriptor;
desc.device_qualifier = (uint8_t *) USB_DeviceQualifier;
//Inicialización de USB
ret = USBD_API->hw->Init(&g_hUsb, &desc, &usb_param);
if (ret == LPC_OK) {
    //USB stuff
}
```

- Manejador de interrupciones:

```
void USB_IRQHandler(void)
{
    USBD_API->hw->ISR(g_hUsb);
}
```

- Inicializar y adjuntar los controladores: si se pudo iniciar el puerto, debemos inicializar el dispositivo HID. Es decir, las estructuras donde irán los reportes, las funciones que manejarán los mismos, etc.
- Habilitar las interrupciones y conectar el dispositivo:

```
NVIC_EnableIRQ(LPC_USB_IRQ);
USBD_API->hw->Connect(g_hUsb, 1);
```

## **Actividades**

Usando como guía el ejemplo del mouse:

1. Crear la estructura de directorios y archivos para el desarrollo de un teclado, similares a la del ejemplo.
2. Generar el descriptor de reportes, ubicado en el archivo *"hid\_desc.c"*. El mismo debe responder a los parámetros de un teclado, recuerde consultar los documentos adjuntos para ver la estructura de los descriptores y compararlo con el del ejemplo. (*"descriptores.pdf"* y *"usbd\_hid.h"*)
3. Generar un descriptor de string apropiado, utilizando una combinación de su nombre y apellido como número de serie del teclado (que no supere los 20 caracteres).
4. Dentro del archivo *"hid\_keyboard.c"*, completar las implementaciones de las rutinas de atención de interrupciones de las 4 teclas, (utilizando la variable *"presionadas"*) y de la función *"Keyboard\_UpdateReport"*. La misma debe emular las teclas "q, w, e, r" mapeando las teclas 1, 2, 3 y 4. (Consultar el código de teclas en el archivo *"hut1\_21.pdf"*)