USB Mouse Firmware

Team Members:
Tariq Ahmed
Nandan Desai

Implementing USBD

Power-up Sequence

When the USB device is connected to the host, the VBUS rises into its valid range and the USB device is notified through a USBDETECTED event.

Once it's detected, implement the USBD start-up sequence.

Enable the USBD peripheral through the ENABLE register. The regulator should be turned on now and a USBPWRRDY event is sent, indicating to the software that it can enable the USB pull-up to signal a USB connection to the host.

Now, enable the USB pull-up signal on D+.

At this point, USB enumeration can take place. The host starts off usb enumeration with a USBRESET (i.e., pulling D+ and D- to low).

USB Enumeration

USB enumeration helps the host in detecting and identifying the connected USB device. The data transfer during this phase is known as a “control transfer”.

Each USB device has a number of “endpoints”, and an endpoint is a source or sink of data. And all control transfers happen on Endpoint 0 of the device.

After the USBRESET, the host uses SETUP requests to get device configuration details. These details are termed as “descriptors”. The SETUP packet consists of request and request type. For our implementation, we focused only on

1. Request: GET_DESCRIPTOR
2. Request Type: device, configuration, interface, endpoint, hid report descriptors

The structure of the various descriptors is specified in `kernel/src/usbd.h`.

Implementation

An EP0SETUP event is received when the host sends a SETUP request on endpoint 0. The USBD peripheral on our board makes this request data available in various SETUP-related registers like BMREQUESTTYPE, BREQUEST, etc. Based on the contents of these registers, we respond back with a descriptor. The STARTEDEPIN task is used to send this data to the host and we receive an EP0DATADONE event when the data transfer is complete and acknowledged by the host.

Finally, send EP0STATUS task to indicate end of control transfer.

Emulating Mouse

After USB enumeration, the host recognizes the board as a mouse. Mouse actions can now be sent to the host via “interrupt transfers” on endpoint 1.

The protocol that the mouse uses to communicate with the USB host is called the REPORT protocol. We need to specify this protocol type in the HID descriptor and HID report descriptor sent during the USB enumeration phase (the previous phase). The mouse action data sent to the host is known as the *input report*. The input report structure looks as follows:

```
typedef struct {
    uint8_t buttons;
    int8_t X;
    int8_t Y;
    int8_t wheel;
} input_report_t;
```

Implementation

We implemented three system calls that the user-space application can use:

sys_mouse_move(x,y): Allows user to move the mouse pointer by specified coordinates. The coordinates provided are relative to the current position of the cursor on the screen.

sys_mouse_scroll(wheel): Allows user to scroll up or down. A positive value causes a scroll up and negative value a scroll down.

sys_mouse_click(button): Allows user to perform a left click or right click.

User-Space Application

The main user-space application is a keyboard-controlled mouse which utilizes the mouse system calls.

Two threads are used:

Thread 0: A task to wait for the user-input and fill up a buffer with the character that the user has typed. This will use the *read()* function from the C library, which in turn will use the *sys_read()* system call.

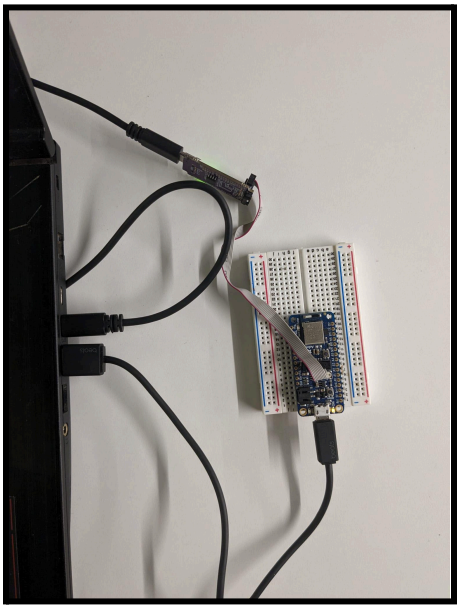
Thread 1: A task to read the input character from the buffer and perform the following action based on the input:

- 'a': move mouse pointer left
- 'd': move mouse pointer right
- 'w': move mouse pointer up
- 's': move mouse pointer down
- 't': scroll mouse wheel up
- 'g': scroll mouse wheel down
- 'q': click left mouse button
- ‘e': click right mouse button

The worst execution time and time period of the two threads based on a clock frequency of 600 Hz are:

Thread	C	T
0	10	65
1	40	65

Demo



YouTube URL: https://www.youtube.com/watch?v=458j4irl9rg

References

1. http://sdpha2.ucsd.edu/Lab_Equip_Manuals/usb_20.pdf
2. https://www.usbmadesimple.co.uk
3. https://www.keil.com/pack/doc/mw/USB/html/index.html
4. https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.8.pdf