

## Lab 3: Build your own mouse

### General Instructions

In this lab, you will build your own BLE mouse to control your computer, tablet or phone.

In this lab, you can get a total of 10 points. 6 points for tasks 1 to 3, 2 points for the demo of your mouse, and 2 points for the presentation.

### Task 1 - Moving the mouse cursor (2.0 points)

In this task, you shall use buttons to move the mouse cursor on your device left, right, up, and down.

For this, two buttons should be sufficient. You could use the following pattern, but feel free to extend or modify this.

- Button 1
  - Single click: move 5px to the right (x-axis)
  - Double click: move 5px to the left (x-axis)
  - Hold button: continue to previous movement (*if needed, increase step size to 10px*)
- Button 2
  - Single click: move 5px up (y-axis)
  - Double click: move 5px down (y-axis)
  - Hold button: continue to previous movement (*if needed, increase step size to 10px*)

The best suitable boards for this task are either the nRF52840-DK (with 4 buttons) or the micro:bit v2 (with 2 physical buttons and 4 capacitive buttons<sup>1</sup>).

To get started building a mouse in Zephyr, you can combine the GATT Human Interface Device (HID) Service<sup>2</sup> with the Zephyr buttons example<sup>3</sup>. We suggest that you combine these two repositories, test it, and then extend it to reach a moving mouse cursor. We strongly suggest that you use the main branch of these two examples and not the 2.7 branch.

---

<sup>1</sup><https://support.microbit.org/support/solutions/articles/19000116318-touch-sensing-on-the-micro-bit>

<sup>2</sup>[https://github.com/zephyrproject-rtos/zephyr/tree/main/samples/bluetooth/peripheral\\_hids](https://github.com/zephyrproject-rtos/zephyr/tree/main/samples/bluetooth/peripheral_hids)

<sup>3</sup><https://github.com/zephyrproject-rtos/zephyr/tree/main/samples/basic/button>

Please be aware, that you might encounter the following limitations of your mouse. Not all devices might accept your mouse, maybe due to a lack of security certificates. Former tests were successful on Windows, Linux, Android, older macOS devices (before M1), and older iOS devices (iPad, iPhone). Please note, that on some phones (e.g., iPhone) you will not see a cursor, but it is there, moves, and reacts to clicks.

*Hints: Before implementing mouse clicks, it might be helpful to read about human interface devices<sup>4</sup> and report maps<sup>5</sup>. To test your mouse, you can use this online service: <https://www.onlinemictest.com/de/maus-test/>. For your implementation, Zephyr threads<sup>67</sup> might be helpful as our setting is full of concurrency, but handle with care.*

## Task 2 - Mouse button clicks (2.0 points)

In this task, you shall add button click functionalities to your mouse from task 1.

Use two buttons to perform mouse button clicks.

- Button 3: left mouse button (single click and double click)
- Button 4: right mouse button (single click and double click)

## Task 3 - Get innovative (2.0 points)

Instead of moving the mouse with button clicks, do the following. This task is open on purpose.

You can choose one of these five tasks:

1. Use it as a shortcut to start a specific application (or so)
  - Button 1 starts application X, button 2 stops the music, ...
2. If you have a micro:bit: use the accelerometer to control your mouse along the x- and y-axis
3. If you have an Arduino Nano 33 BLE Sense: use the accelerometer or gyroscope to control your mouse along the x- and y-axis

---

<sup>4</sup>[https://wiki.osdev.org/USB\\_Human\\_Interface\\_Devices](https://wiki.osdev.org/USB_Human_Interface_Devices)

<sup>5</sup><https://eleccelerator.com/tutorial-about-usb-hid-report-descriptors/>

<sup>6</sup><https://github.com/zephyrproject-rtos/zephyr/blob/main/samples/basic/threads/src/main.c>

<sup>7</sup><https://docs.zephyrproject.org/latest/kernel/services/threads/index.html>

4. Do a performance evaluation:
  - Find the limits of the mouse
    - Push the stability
    - Does it reconnect after sleep of the host device?
    - How many button press events per second can it handle? (*Note: do not press the button manually, write some code to send button pressed events*)
    - Find more aspects to evaluate
5. Be even more innovative and define and implement your own task 3

## Useful information

If you choose to use PlatformIO for your implementation, take a look at its debugging capabilities. Just using logging/`printk` has its limits. Take a look at general PlatformIO debugging<sup>8</sup> and Zephyr-specific PlatformIO debugging<sup>9</sup>.

A useful function to set your own passkey which you need during pairing is `bt_passkey_set(unsigned int passkey)`. Otherwise, print the passkey via UART.

Please note, that you **can only pair your mouse with a single host!** To avoid "funny" side effects, delete your firmware before pairing the mouse with another host. As an alternative, you can try to use `CONFIG_BT_MAX_PAIRED` to increase the number of paired devices.

---

<sup>8</sup><https://docs.platformio.org/en/latest/plus/debugging.html>

<sup>9</sup>[https://docs.platformio.org/en/latest/tutorials/nordicnrf52/zephyr\\_debugging\\_unit\\_testing\\_inspect.html](https://docs.platformio.org/en/latest/tutorials/nordicnrf52/zephyr_debugging_unit_testing_inspect.html)