Grant Reboucas
P4
Project 4: Device Drivers

Synopsis

Create a joystick driver to work with chompapp and display the values of the x and y axis as well as whether the button is off or on.

Description

Emit

The purpose of this function is to take in parameters, and set a certain even, either changing the x or y axis or pressing the button and write the current state to the usb's respective folder. This is how the values are displayed for each axis and button, and it is used to update the current state once the user presses a button or uses the joystick. [2]

ToBin

Purpose of this function is to convert decimal to binary. When a joystick value is changed, it is all sent in decimal format, and the driver needs to interpret it in binary, so this function is called when the state is changed and the values of the chompstick device change. [3]

Main

Uinput

Uinput is the module that makes it possible to read/emulate input devices from the user space. The basic idea is to set up a uinput structure and enable the keys that are to be read from the input device. In this case, this was the arrow keys and the spacebar button. The way the driver works however, is that it is to read the x and y axis as well as a joystick button being pressed. So instead of setting up arrow keys and the spacebar button, ABS_X and ABS_Y is used, as well as BTN_JOYSTICK [8]. The driver is supposed to work for joysticks, so while a user may be using the keys and spacebar on a keyboard, it is to be read as a joystick. Uinput_setip is the used structure because it is how uinput functions. Js_event was looked into, but it is not needed in order for the device to work.

LibUSB

This is how the connection to chompapp is made. Creating new libUSB device and setting the product and vendor IDs makes it so when chompapp is running, the driver is connected to the right sub device. It is also used to get data from chompapp to the driver to use. When buttons are pressed, the changes are set/transferred via lib_usb_interrupt [1]. It is how the current status of the x and y axis as well as the button are sent to the driver to be interpreted.

Testing

Multiple test were ran to make sure that the driver did not only work once, as well as loading to a fresh kernel and running it on there. Driver was always able to be made and always worked, even on a freshly installed kernel. The values for button and the x and y axis were monitored and correctly displayed the position of each.

Bug

None

Link

https://youtu.be/bIPM92yJXak

References

[1]https://www.dreamincode.net/forums/topic/148707-introduction-to-using-libusb-10/
[2]https://www.kernel.org/doc/html/v4.12/input/uinput.html
[3]https://stackoverflow.com/questions/7911651/decimal-to-binary
[4]https://www.freedesktop.org/software/libevdev/doc/latest/
[5]https://stackoverflow.com/questions/16032982/getting-live-info-from-dev-input
[6]https://scaryreasoner.wordpress.com/2008/02/22/programming-joysticks-with-linux/
[7]https://www.kernel.org/doc/html/v4.16/input/joydev/joystick-api.html#initialization
[8]https://stackoverflow.com/questions/39559063/libsuinput-creating-joystick-with-more-than-one-button