Quimby

A Bluetooth-USB HID Proxy

Danny Robson

Problem

- I want to use a specific Bluetooth keyboard
- I *also* want to navigate my bootloader

Solution: HID Proxy

Run the Blueooth stack on a USB dongle.

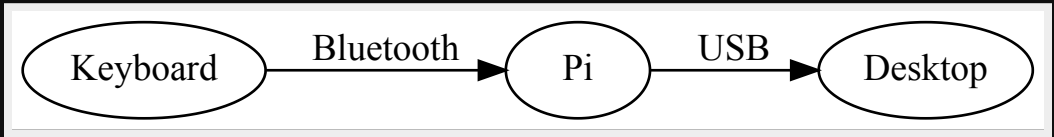
Issues

- Expensive
- Often end-of-life
- Unsupported

Solution: Linux

- The `Pi Zero W' has USB OTG.
- It can present as a device.
- Uses the kernel’s `USB Gadget' interface.

Plan



Workthrough

1. Boot Pi
2. Setup Bluetooth
3. Setup USB OTG
4. Forward events
5. Load at boot

Boot

/boot/config.txt

```
dtoverlay=dwc2
```

/etc/modules-load.d/quimby

```
dwc2
libcomposite
```

Bluetooth

```
CONTROLLER="00:11:22:33:44:55"
DEVICE="66:77:88:99:aa:bb"

bluetoothctl -- power on
bluetoothctl -- select  ${CONTROLLER}
bluetoothctl -- pair    ${DEVICE}
bluetoothctl -- connect ${DEVICE}
bluetoothctl -- trust   ${DEVICE}
```

USB

Use `ConfigFS` to define the USB parameters send to the host.

ConfigFS

/sys/kernel/config/usb_gadget/

```
`-- quimby
    |-- UDC
    |-- bDeviceClass
    |-- bDeviceProtocol
    |-- bDeviceSubClass
    |-- bMaxPacketSize0
    |-- bcdDevice
    |-- bcdUSB
    |-- configs
    |   `-- c.1
    |       |-- MaxPower
    |       |-- bmAttributes
    |       |-- hid.usb0 -> ../../../../usb_gadget/quimby/functio
    |       |-- hid.usb1 -> ../../../../usb_gadget/quimby/functio
    |       `-- strings
    |           `-- 0x0409
```

Success!

journalctf -f

```
usb 1-5.1.2: USB disconnect, device number 9
xhci_hcd 0000:02:00.0: WARN Set TR Deq Ptr cmd failed due to inco
xhci_hcd 0000:02:00.0: WARN Set TR Deq Ptr cmd failed due to inco
usb 1-5.1.2: new high-speed USB device number 10 using xhci_hcd
usb 1-5.1.2: New USB device found, idVendor=1d6b, idProduct=0104,
usb 1-5.1.2: New USB device strings: Mfr=1, Product=2, SerialNumb
usb 1-5.1.2: Product: Virtual Keyboard
usb 1-5.1.2: Manufacturer: Quimby
usb 1-5.1.2: SerialNumber: fedcba9876543210
input: Quimby Virtual Keyboard as [...]
hid-generic 0003:1D6B:0104.000D: input,hidraw5: USB HID v1.01 Key
```

Event Forwarding

Trivial Test: Print `e'

```
echo -ne "\0\0\x8\0\0\0\0\0" > /dev/hidg0
```

Useful Event Forwarding

```
src = open ("/dev/input/event0")
dst = open ("/dev/hidg0")

while event in src:
    hid_code = translate (event.scancode)
    packet = build_packet (hid_code)
    send_packet(dst, packet)
```

Bootup

- Bluetooth
- systemd
- udev

Bluetooth

/etc/bluetooth/main.conf

```
AutoEnable=true
```

systemd

/usr/lib/systemd/system/quimby.service

```
[Unit]
Description=Quimby Input Forwarder
Requires=bluetooth.service

[Service]
Type=simple
ExecStartPre=/usr/local/bin/quimby-setup quimby
ExecStart=/usr/local/bin/quimby-relay /dev/input/event0 /dev/hidg
ExecStop=/usr/local/bin/quimby-cleanup quimby
```

udev

/etc/udev/rules.d/quimby.rules

```
ACTION=="add", \
SUBSYSTEM=="input", \
KERNEL=="event0", \
TAG+="systemd", \
ENV{SYSTEMD_WANTS}="quimby.service"
```

TODO

- More reliable setup/teardown
- Hardcode fewer paths
- Support multiple devices

Questions?

YMMV; https://gitlab.com/dcro/quimby