Reverse Engineering

Making ASUS laptops work better in Linux
And no I did not finish these slides
Yes I worked right up until I left home

Who am I?

- **Work for JASIC Technology Europe on welding machines**
  - Embedded Linux, QT based HMI, mobile application in Flutter
- **Official maintainer of most ASUS platform stuff in kernel**
  - Slowly shifting things to be standardised
- **Wrote helper suite for laptops**
- **Rewrote the entire Doom engine in Rust because why not?**

So many patches!

```
311da4b001d3 (HEAD -> asus-next-unstable, origin/asus-next-unstable) ACPI: PM: Quirk ASUS ROG M16 to default to S3 sleep
a9465f7b6347 hid-asus-ally: Add full gamepad support
1d9c258f1a14 asus-wmi: deprecate bios features
924593271536 platform/x86: asus-armoury: add core count control
a16d3bc311d0 platform/x86: asus-armoury: add apu-mem control support
c5880eea3611 platform/x86: asus-armoury: add dgpu tgp control
234138b09f9c platform/x86: asus-armoury: move existing tunings to asus-armoury module
e0762234bcd0 platform/x86: asus-wmi: Refactor Ally CSEE quirks
fd09dc95a61b platform/x86: asus-wmi: Add quirk for ROG Ally X
85abe37448c2 platform/x86: asus-wmi: don't fail if platform_profile already registered
ddb9aa5efc7f platform/x86: asus-wmi: add debug print in more key places
8c98924f483e Input: xpad - add support for ASUS ROG RAIKIRI PRO
5941d48bca58 hid-asus: use hid for brightness control on keyboard
81338fa08af5 platform/x86/amd: pmf: Make ASUS GA403 quirk generic
b97ee04673e0 platform/x86/amd: pmf: Add quirk for ROG Ally X
9a0c3140acb8 Change line limit
d34af755a533 platform/x86/amd: pmf: Make ASUS GA403 quirk generic
d2dfed310aae platform/x86: asus-wmi: Add quirk for ROG Ally X
d1aa95e86f17 hid-asus: add ROG Ally X prod ID to quirk list
4c83ee4bf32e platform/x86/amd: pmf: Add quirk for ROG Ally X
e6e18021ddd0 ALSA: hda/realtek: cs35l41: Fixup remaining asus strix models
d8b17a364ec4 (tag: platform-drivers-x86-v6.11-1) platform/x86: asus-wmi: fix TUF laptop RGB variant
cee77149ebe9 Input: xpad - add support for ASUS ROG RAIKIRI PRO
2be46155d792 ALSA: hda/realtek: Adjust G814JZR to use SPI init for amp
e901f10adb1f HID: asus: add ROG Z13 lightbar
08b50c6b0b09 HID: asus: add ROG Ally N-Key ID and keycodes
2c82a7b20f7b HID: asus: make asus_kbd_init() generic, remove rog_nkey_led_init()
59d2f5b7392e HID: asus: fix more n-key report descriptors if n-key quirked
88c0ef69dd88 platform/x86: asus-wmi: cleanup main struct to avoid some holes
a94e8a56f9e1 platform/x86: asus-wmi: Add support for MCU powersave
7e7a5dee4973 platform/x86: asus-wmi: ROG Ally increase wait time, allow MCU powersave
892fc4b57dc5 platform/x86: asus-wmi: adjust formatting of ppt-<name>() functions
5fc378183d94 platform/x86: asus-wmi: store a min default for ppt options
e0ae0ecce486 platform/x86: asus-wmi: support toggling POST sound
ae834a549ec1 platform/x86: asus-wmi: add support variant of TUF RGB
eb3bac90549a platform/x86: asus-wmi: add support for Vivobook GPU MUX
f81d13df1aa8 platform/x86: asus-wmi: add support for 2024 ROG Mini-LED
0bfe105018bd ALSA: hda/realtek: cs35l41: Support ASUS ROG G634JYR
20fe5e9be47e MAINTAINERS: add Luke Jones as maintainer for asus notebooks
e0894ff038d8 platform/x86: asus-wmi: disable USB0 hub on ROG Ally before suspend
2c97d3e55b70 platform/x86: asus-wmi: add support for ASUS screenpad
4106a70ddad5 (tag: platform-drivers-x86-v6.6-2) platform/x86: asus-wmi: Support 2023 ROG X16 tablet mode
acce85a7dd28 (tag: platform-drivers-x86-v6.6-1) platform/x86: asus-wmi: corrections to egpu safety check
fa69653f87ee platform/x86: asus-wmi: Fix support for showing middle fan RPM
e0b278e7b5da platform/x86: asus-wmi: expose dGPU and CPU tunables for ROG
abac4259fc0a platform/x86: asus-wmi: support setting mini-LED mode
609b3670c29f platform/x86: asus-wmi: add safety checks to gpu switching
d49f4d1a30ac platform/x86: asus-wmi: don't allow eGPU switching if eGPU not connected
d4eca58aafe2 platform/x86: asus-wmi: add WMI method to show if egpu connected
ee887807d05d platform/x86: asus-wmi: support middle fan custom curves
536fce82d729 platform/x86: asus-wmi: add support for showing middle fan RPM
77ee9d299e6d platform/x86: asus-wmi: add support for showing charger mode
5251605f4d29 ALSA: hda/realtek: Add quirk for ASUS ROG GZ301V
72cea3a3175b ALSA: hda/realtek: Whitespace fix
33d7c9c3bf70 ALSA: hda/realtek: Add quirk for ASUS ROG G614Jx
b759a5f097cd ALSA: hda/realtek: Amend G634 quirk to enable rear speakers
```

```
d4eca58aafe2 platform/x86: asus-wmi: add WMI method to show if egpu connected
ee887807d05d platform/x86: asus-wmi: support middle fan custom curves
536fce82d729 platform/x86: asus-wmi: add support for showing middle fan RPM
77ee9d299e6d platform/x86: asus-wmi: add support for showing charger mode
5251605f4d29 ALSA: hda/realtek: Add quirk for ASUS ROG GZ301V
72cea3a3175b ALSA: hda/realtek: Whitespace fix
33d7c9c3bf70 ALSA: hda/realtek: Add quirk for ASUS ROG G614Jx
b759a5f097cd ALSA: hda/realtek: Amend G634 quirk to enable rear speakers
9abc77fb144f ALSA: hda/realtek: Add quirk for ASUS ROG GA402X
8cc87c055d28 ALSA: hda/realtek: Add quirk for ASUS ROG GX650P
82edd1bd7f98 ALSA: hda/realtek: Add quirk for ASUS ROG GV601V
555434fd5c6b ALSA: hda/realtek: Add quirk for ASUS ROG G634Z
e6c7e2711df6 HID: asus: reformat the hotkey mapping block
73920f615159 HID: asus: add keycodes for 0x6a, 0x4b, and 0xc7
74e47b2c52ed HID: asus: Add support for ASUS ROG Z13 keyboard
a4671b7fba59 ALSA: hda/realtek: Add quirk for 2nd ASUS GU603
36abde8d24ad platform/x86: asus-wmi: Add support for ROG X16 tablet mode
2ea8e1297801 ALSA: hda/realtek: Add quirk for ASUS GV601R laptop
66ba7c885073 ALSA: hda/realtek: Correct pin configs for ASUS G533Z
9180991873 0f platform/x86: asus-wmi: Expand support of GPU fan to read RPM and label
ba1f818053b0 ALSA: hda/realtek: Add quirk for ASUS GA503R laptop
bc2c23549ccd ALSA: hda/realtek: Add pincfg for ASUS G533Z HP jack
[illegible] ALSA: hda/realtek: Add pincfg for ASUS G513 HP jack
[illegible] platform/x86: asus-wmi: Increase FAN_CURVE_BUF_LEN to 32
a2bdf10ce90e platform/x86: asus-wmi: Increase FAN_CURVE_BUF_LEN to 32
61f64515299e platform/x86: asus-wmi: Implement TUF laptop keyboard power states
e305a71cea37 platform/x86: asus-wmi: Implement TUF laptop keyboard LED modes
12ff4c803d23 platform/x86: asus-wmi: Support the GPU fan on TUF laptops
601eb4c8e150 platform/x86: asus-wmi: Modify behaviour of Fn+F5 fan key
e397c3c460bf platform/x86: asus-wmi: Add support for ROG X13 tablet mode
00aa846955fb platform/x86: asus-wmi: Adjust tablet/lidflip handling to use enum
01ef026ab363 platform/x86: asus-wmi: Support the hardware GPU MUX on some laptops
ebc443ad379f platform/x86: asus-wmi: Refactor panel_od attribute
36450e7db0fe platform/x86: asus-wmi: Refactor egpu_enable attribute
cdf36fc865f0 platform/x86: asus-wmi: Refactor disable_gpu attribute
d956c889be80 platform/x86: asus-wmi: Document the panel_od sysfs attribute
3206376f099d platform/x86: asus-wmi: Document the egpu_enable sysfs attribute
7e64c486e807 platform/x86: asus-wmi: Document the dgpu_disable sysfs attribute
170f0da25dac platform/x86: asus-wmi: Convert all attr-show to use sysfs_emit
0f0ac158d28f platform/x86: asus-wmi: Add support for custom fan curves
87c7ee7ad85a HID: asus: Prevent Claymore sending suspend event
c63d44ae6024 asus-wmi: Add support for platform_profile
382b91db8044 asus-wmi: Add egpu enable method
98829e84dc67 asus-wmi: Add dgpu disable method
ca91ea34778f asus-wmi: Add panel overdrive functionality
739d0959fbed ALSA: hda: Add quirk for ASUS Flow x13
3fdcf7cdfc22 HID: asus: Remove check for same LED brightness on set
28117f3a5c3c platform/x86: asus-nb-wmi: Revert "add support for ASUS ROG Zephyrus G14 and G15"
98c0c85b1040 platform/x86: asus-nb-wmi: Revert "Drop duplicate DMI quirk structures"
c980512b4512 HID: asus: filter G713/G733 key event to prevent shutdown
4bfb2c72b2bf HID: asus: Filter keyboard EC for old ROG keyboard
2dfbacc65d1d ACPI: video: use native backlight for GA401/GA502/GA503
76fae6185f54 ALSA: hda/realtek: GA503 use same quirks as GA401
9a0b44fbfea1 HID: asus: Add support for 2021 ASUS N-Key keyboard
b92b80246e06 HID: asus: Add support for ASUS N-Key keyboard
c3cdf189276c ALSA: hda: fixup headset for ASUS GX502 laptop
```

What is reverse engineering?

Reverse engineering is..

- **Not just reading assembly**
- **The art of figuring out how that black-box works**
- **Analysis of data streams**
  - Pattern recognition (we’re good at this)
  - Cause and effect
- **A lot of searching and piecing together information**

What I'll cover (try to)

- **USB HID data and structure**
- **Searching for clues in binaries**
- **Analysing DLL in Cutter or Ghidra after finding clues**
- **Microsoft WMI (like it or not, can't avoid it)**
  - Probably next year
- **What goes in to the kernel**

5 years of progress

- **Started with a barely functioning GX502**
- **Wireshark captures of USB data**
  - Cause and effect, look for patterns
- **USB HID structure**
  - Report ID
  - Endpoints
  - Packets: |5a|d1|02|0n|2c|01|03|00|00|00|00|00|00|00|00|

5 years of progress

- **Reverse engineered DLL**
  - Found many debug strings intact
  - Cross referenced logs
  - Find where those strings were used and summon ancient gods
- **Initially added willy-nilly to asus-wmi**
- **Now written a new driver, asus-armoury**
  - Using firmware_attributes class

Firmware Attributes

```
❯ sudo fwupdmgr get-bios-setting --json
{
  "BiosSettings" : [
    {
      "Name" : "ppt_fppt",
      "Description" : "Set the CPU slow package limit",
      "Filename" : "/sys/class/firmware-attributes/asus-armoury/attributes/ppt_fppt",
      "BiosSettingId" : "com.asus-armoury.ppt_fppt",
      "BiosSettingCurrentValue" : "80",
      "BiosSettingReadOnly" : "false",
      "BiosSettingType" : 2,
      "BiosSettingLowerBound" : 5,
      "BiosSettingUpperBound" : 150,
      "BiosSettingScalarIncrement" : 1
    },
```

HID Packet Structure

Don’t nuke Windows on new hardware

- **Windows is your baseline**
  - Fastboot can be a pain in the arse
- **The applications will provide many things:**
  - Action outputs to sniff
  - Libraries to slice and dice
- **Worst case: toggle something in windows and see if you can find the effect in Linux**

HID (Human Interface Descriptor)

```
Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        5
      bAlternateSetting       0
      bNumEndpoints           1
      bInterfaceClass         3 Human Interface Device
      bInterfaceSubClass      1 Boot Interface Subclass
      bInterfaceProtocol      1 Keyboard
      iInterface              1 ASUSTeK Computer Inc.
        HID Device Descriptor:
          bLength                 9
          bDescriptorType        33
          bcdHID               1.10
          bCountryCode            0 Not supported
          bNumDescriptors         1
          bDescriptorType        34 Report
          wDescriptorLength     240
          Report Descriptor: (length is 240)
            Item(Global): Usage Page, data= [ 0x01 ] 1
                          Generic Desktop Controls
            Item(Local ): Usage, data= [ 0x05 ] 5
                          Gamepad
            Item(Main  ): Collection, data= [ 0x01 ] 1
                          Application
-- Continued over there ->
```

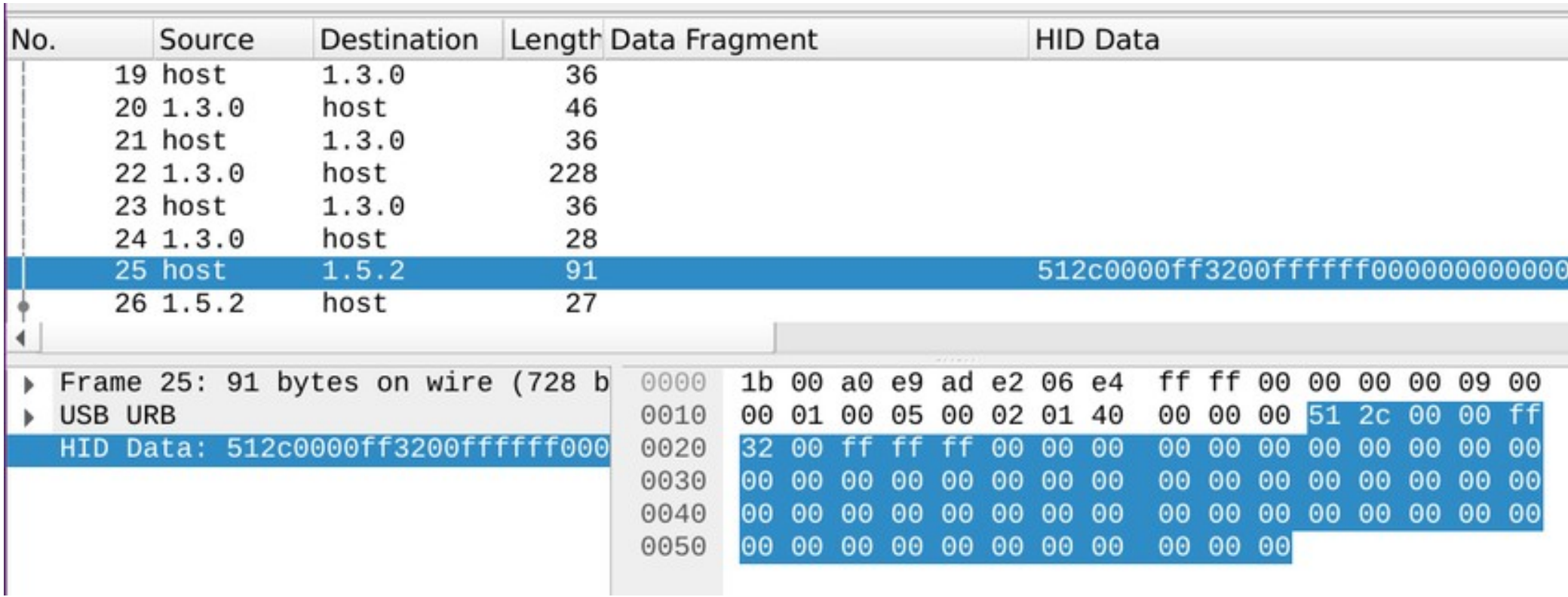
This is the ROG Ally, it uses the same ITE MCU as laptops.

Output report, helpful for hid-asus-ally driver (gamepad part)

```
Item(Global): Report ID, data= [ 0x0b ] 11
        Item(Main  ): Collection, data= [ 0x00 ] 0
                        Physical
                Item(Local ): Usage, data= [ 0x30 ] 48
                              Direction-X
                Item(Local ): Usage, data= [ 0x31 ] 49
                              Direction-Y
                Item(Global): Logical Minimum, data= [ 0x00 ] 0
                Item(Global): Logical Maximum, data= [ 0xff 0xff ] 65535
                Item(Global): Physical Minimum, data= [ 0x00 ] 0
                Item(Global): Physical Maximum, data= [ 0xff 0xff ] 65535
                Item(Global): Report Count, data= [ 0x02 ] 2 (X AND Y)
                Item(Global): Report Size, data= [ 0x10 ] 16 (2x bytes)
                Item(Main  ): Input, data= [ 0x02 ] 2
                              Data Variable Absolute No_Wrap Linear

        Item(Main  ): End Collection, data=none
```

The HID packet (ASUS ROG Azoth)



No.	Source	Destination	Length	Data Fragment	HID Data
19	host	1.3.0	36		
20	1.3.0	host	46		
21	host	1.3.0	36		
22	1.3.0	host	228		
23	host	1.3.0	36		
24	1.3.0	host	28		
25	host	1.5.2	91		512c0000ff3200ffffff000000000000
26	1.5.2	host	27		

```
▸ Frame 25: 91 bytes on wire (728 b
▸ USB URB
  HID Data: 512c0000ff3200ffffff000
```

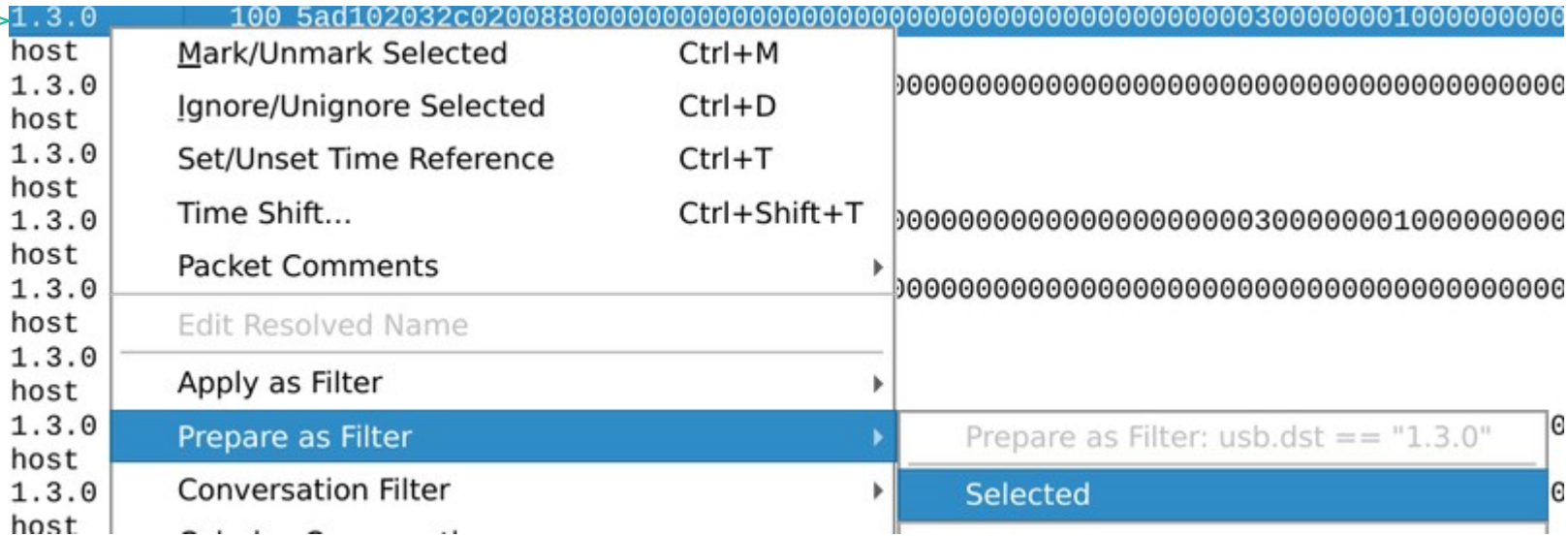
```
0000  1b 00 a0 e9 ad e2 06 e4  ff ff 00 00 00 00 09 00
0010  00 01 00 05 00 02 01 40  00 00 00 51 2c 00 00 ff
0020  32 00 ff ff ff 00 00 00  00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0050  00 00 00 00 00 00 00 00  00 00 00
```

Pattern Recognition (ASUS ROG Ally X)

Length	Data Fragment
38	0100
100	5ad10101000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
100	5ad10103000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
100	5ad10103000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
100	5ad102012c010900000000000000000000050000190000000000000000010a000000000000000000040000000038c88760000000000000000000000000000000000
100	5ad102012c0200980000000000000000000500001900000000000000000200990000000000000000040000000038c88760000000000000000000000000000000000
100	5ad102012c0200980000000000000000000500001900000000000000000200990000000000000000040000000038c88760000000000000000000000000000000000
100	5ad102022c010b000000000000000000000400000000028223000000010c0000000000000000000400000000002820d000000000000000000000000000000000000
100	5ad102022c02009a000000000000000000040000000002822300000002009b000000000000000000400000000002820d000000000000000000000000000000000000
100	5ad102022c02009a000000000000000000040000000002822300000002009b000000000000000000400000000002820d000000000000000000000000000000000000
100	5ad102032c01070000000000000000000000000000000000000000010800000000000000000000000000000000000000000000000000000000000000000000000
100	5ad102032c0200880000000000000000000000000000000000000003000000010000000000000000000000000000000000000000000000000000000000000000
100	5ad102032c0200880000000000000000000000000000000000000003000000010000000000000000000000000000000000000000000000000000000000000000
100	5ad102042c01050000000000000000000000000000000000000000010600000000000000000000000000000000000000000000000000000000000000000000000
100	5ad102042c02000d0000000000000000000000000000000000000003000000010000000000000000000000000000000000000000000000000000000000000000
100	5ad102042c02000d0000000000000000000000000000000000000003000000010000000000000000000000000000000000000000000000000000000000000000
100	5ad102052c010100000000000000000000050000160000000000000001020000000000000000000400000000028231000000000000000000000000000000000000
100	5ad102052c02005a00000000000000000005000016000000000000000200760000000000000000040000000002823100000000000000000000000000000000000
100	5ad102052c02005a00000000000000000005000016000000000000000200760000000000000000040000000002823100000000000000000000000000000000000
100	5ad102062c0103000000000000000000000400000000002824d0000000104000000000000000000000500001e00000000000000000000000000000000000000000000
100	5ad102062c02009700000000000000000004000000000002824d0000000200960000000000000000000500001e00000000000000000000000000000000000000000000
100	5ad102062c02009700000000000000000004000000000002824d0000000200960000000000000000000500001e00000000000000000000000000000000000000000000
100	5ad102072c01110000000000000000000000000000000000000000011200000000000000000000000000000000000000000000000000000000000000000000000
100	5ad102072c01110000000000000000000000000000000000000000011200000000000000000000000000000000000000000000000000000000000000000000000
100	5ad102072c01110000000000000000000000000000000000000000011200000000000000000000000000000000000000000000000000000000000000000000000
100	5ad102082c02008e000000000000000002008e000000000000000002008f000000000000000002008f0000000000000000000000000000000000000000000000
100	5ad102082c02008e000000000000000002008e000000000000000002008f000000000000000002008f0000000000000000000000000000000000000000000000
100	5ad102082c02008e000000000000000002008e000000000000000002008f000000000000000002008f0000000000000000000000000000000000000000000000

Filtering interesting things (ASUS ROG Ally 1 or X)

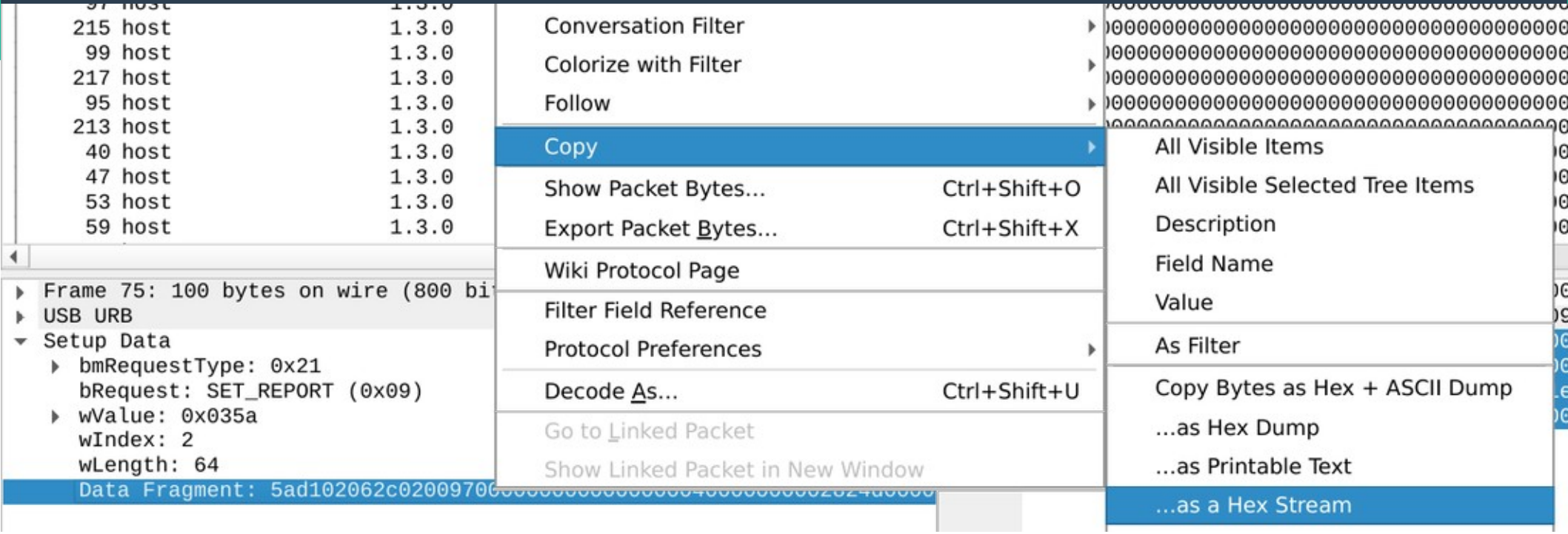
Right-click



(usb.dst == "1.3.0") && (usb.data_fragment contains 5a:d1:02:)

No.	Source	Destination	Length	Data Fragment
45	host	1.3.0	100	5ad102012c020098000000000000000005000019000000000000002009
51	host	1.3.0	100	5ad102022c02009a000000000000000004000000002822300000002009
57	host	1.3.0	100	5ad102032c020088000000000000000000000000000000000000003000
63	host	1.3.0	100	5ad102042c02000d000000000000000000000000000000000000003000

Data Collection (ASUS ROG Ally, remapping)



5ad102062c02009700000000000000040000000002824d0000000200960000000000000000500001e0000000000000000000000000000000000000000

Data Collection (effects of actions)

5ad102062c0200970000000000000000040000000002824d000000020096000000000000
000000500001e00000000000000000000000000000000000000000000

```
G = key group, up to two keys only. One number per pair of buttons on the device
L = data length
D = keymap group (device type), this corresponds to the HID decriptor used to output data
Blocks of 22 bytes, 10 bytes = group + key code

|              |First key mapping                                         |Second key mapping                                         |              |
|        |G |L |D |Primary                         |D |Secondary                          |D |Primary                         |D |Secondary                          |wasted bytes |
X button:                                                                                                                                              : Mapped to
|5a|d1|02|06|2c|01|01|00|00|00|00|00|00|00|00|00|04|00|00|00|00|02|82|4d|00|00|00|01|04|00|00|00|00|00|00|00|00|00|05|00|00|1e|00|00|00|00|00|00|00|    | A button
|5a|d1|02|06|2c|01|02|00|00|00|00|00|00|00|00|00|04|00|00|00|00|02|82|4d|00|00|00|01|04|00|00|00|00|00|00|00|00|00|05|00|00|1e|00|00|00|00|00|00|00|    | B button
|5a|d1|02|06|2c|01|03|00|00|00|00|00|00|00|00|00|04|00|00|00|00|02|82|4d|00|00|00|01|04|00|00|00|00|00|00|00|00|00|05|00|00|1e|00|00|00|00|00|00|00|    | default
|5a|d1|02|06|2c|01|04|00|00|00|00|00|00|00|00|00|04|00|00|00|00|02|82|4d|00|00|00|01|04|00|00|00|00|00|00|00|00|00|05|00|00|1e|00|00|00|00|00|00|00|    | Y button

Y button:                                                                                                                                              : Mapped to
|5a|d1|02|06|2c|01|03|00|00|00|00|00|00|00|00|00|04|00|00|00|00|02|82|4d|00|00|00|01|01|00|00|00|00|00|00|00|00|00|05|00|00|1e|00|00|00|00|00|00|00|    | A button
|5a|d1|02|06|2c|01|03|00|00|00|00|00|00|00|00|00|04|00|00|00|00|02|82|4d|00|00|00|01|02|00|00|00|00|00|00|00|00|00|05|00|00|1e|00|00|00|00|00|00|00|    | B button
|5a|d1|02|06|2c|01|03|00|00|00|00|00|00|00|00|00|04|00|00|00|00|02|82|4d|00|00|00|01|03|00|00|00|00|00|00|00|00|00|05|00|00|1e|00|00|00|00|00|00|00|    | X button

View button:
|5a|d1|02|07|2c|01|11|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|01|12|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|    | default
|5a|d1|02|07|2c|01|06|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|01|12|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|    | right bump
Menu button:
|5a|d1|02|07|2c|01|11|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|01|06|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|    | right bump
```

Ask Cthulhu for help

Commands are:

- 01 = SetGamepadMode
- 02 = SetMapping
- 03 = SetGamepadJoystickMapping
- 04 = SetGamepadJoystickDZOT
- 05 = SetGamepadTriggerRange
- 06 = SetGamePadVibrationIntensity
- 08 = Led control (see kernel patch)
- 09 = New for Ally X ????????????
- 0A = CheckReadyForMapping
- 0B = SetXboxControllerEnabled
- 0C = CheckSupportFlag(XboxControllerStatus)
- 0D = Load/CommitUserCalData X: (stable = %5hu, min = %5hu, max = %5hu), Y: (st
- 0E = CheckSupportFlag(UserCal)
- 0F = SetGamePadTurboParams
- 10 = CheckSupportFlag(Turbo)
- 12 = CheckSupportFlag(JoystickRespCurve)
- 13 = SetGamePadJoystickRespCurveParamsForSide
- 14 = CheckSupportFlag(JoystickDirToXboxBtn)

```
|5a| HID page
|  |d1| Code page?
|  |  |0n| Command
|  |  |  |1e| Length
|  |  |02|0n|2c| Command 02 is special, 0n is a group, 2c is length
```

```
Set gamepad mode:
|5a|d1|01|01|0n|

- 01 = gamepad
- 02 = wasd
- 03 = mouse/desktop?
```

Great my soul is gone, now what?

Kernel work, or userspace HID raw

- **Why not both?**
  - Userspace requires root. **asusd** is a daemon exposing *safe* dbus interfaces for these things.
  - Kernel patches move things out of userspace
  - Progressive move to kernel
- **Pet peeve: interpreted languages do not belong in ring-0**
  - Don’t slow my systems down with bloat

What's in a kernel patch?

```
+static int rog_nkey_led_init(struct hid_device *hdev)
+{
+       u8 buf_init_start[] = { FEATURE_KBD_LED_REPORT_ID1, 0xB9 };
+       u8 buf_init2[] = { FEATURE_KBD_LED_REPORT_ID1, 0x41, 0x53, 0x55, 0x53, 0x20,
+                          0x54, 0x65, 0x63, 0x68, 0x2e, 0x49, 0x6e, 0x63, 0x2e, 0x00 };
+       u8 buf_init3[] = { FEATURE_KBD_LED_REPORT_ID1,
+                                          0x05, 0x20, 0x31, 0x00, 0x08 };
+       int ret;
+
+       hid_info(hdev, "Asus initialise N-KEY Device");
+       /* The first message is an init start */
+       ret = asus_kbd_set_report(hdev, buf_init_start, sizeof(buf_init_start));
+       if (ret < 0) {
+               hid_warn(hdev, "Asus failed to send init start command: %d\n", ret);
+               return ret;
+       }
+       /* Followed by a string */
+       ret = asus_kbd_set_report(hdev, buf_init2, sizeof(buf_init2));
+       if (ret < 0) {
+               hid_warn(hdev, "Asus failed to send init command 1.0: %d\n", ret);
+               return ret;
+       }
```

The Ally handheld driver

- **Nearly 3000 lines**
- **All features supported**
  - Gamepad
  - RGB
  - Configuration (remap, calibrate, deadzones etc)
- **Standalone, but requires nothing else to grab the HID endpoints**

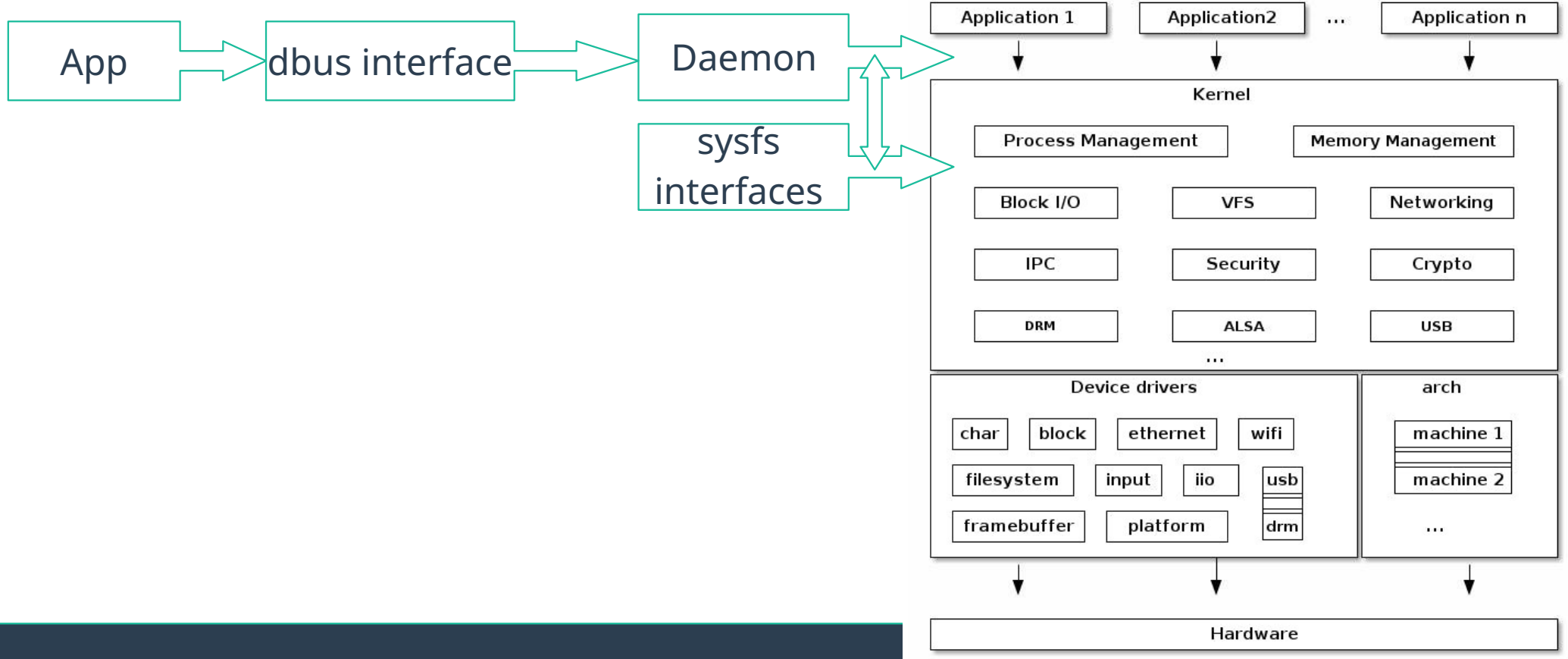
The hid-asus driver

- **Generic, it supports almost every keyboard**
- **Lots of quirks**
- **ASUS is surprisingly good at keeping things standardised**
- **Almost all keyboards use an ITE device for MCU**
  - USB and I2C
  - TUF laptops use WMI to control RGB

So far?

- **Find the device and watch I/O**
  - Find patterns, cause and effect mapping
  - Write small test apps to verify
- **Write the kernel drivers**
- **Write userspace apps to use these safely**
  - Daemons exposing safe dbus interface + user apps

What to aim for



Sometimes you just need secrets whispered to you

Many many hours of ruthless testing and prayers to the great old ones had led me to this conclusion earlier but the data was still spears thrown in the dark.

Hi Luke,

Please check the following update. Thanks

After a few days of analysis, we've made a breakthrough. The root cause on the MCU side, which led to the device losing connection, has been analyzed. However, we still need Luke to investigate the system layer and determine the best way to resolve this issue.

The following timeline outlines the expected power off and power on sequence. The MCU must wait for the USB SOF (Start-of-Frame) signal to stop before unplugging the USB device.(recommended by MCU vendor)