New Tooling for Infrared

Sean Young <sean@mess.org>

Media Summit, September 2024

Outline

First, there was lirc

- User space daemon
- Connect over unix domain socket to receive keystrokes or send transmit command
- Definition of remotes in `.lircd.conf` files - a few parameterized IR protocols supported
- Out of tree kernel drivers

Merging kernel drivers

- IR remotes now work as regular input devices
- IR decoding moved to kernel space
- New rc keymap format
- Only the most popular IR protocols supported - no parameters
- lirc daemon still supported - for `.lircd.conf` files
- Decoding in kernel space improves latency

Current state of IR decoding

- Most popular IR protocols decoded in kernel
- Many other protocols still need lirc daemon
- Remaining protocols not supported
- Separate tooling for rc keymaps and `.lircd.conf` files

IRP Notation

RC5:

```
{36k,msb,889}<1,-1|-1,1>(1,~F:1:6,T:1,D:5,F:6,^114m)*
[D:0..31,F:0..127,T@:0..1=0]
```

NEC:

```
{38.4k,564}<1,-1|1,-3>(16,-8,D:8,S:8,F:8,
~F:8,1,^108m,(16,-4,1,^108m)*)
[D:0..255,S:0..255=255-D,F:0..255]
```

- DSL for describing any IR protocol (incl. air conditioners)
- Existing implementation in Java (IrpTransmogrifier)
- Extensive database of IRPs maintained by IrpTransmogrifier project, larger than rc keymaps or even lirc project - `IrpProtocols.xml`
- Covers every possible protocol and can encode key down, repeat, and key up events

BPF and IR decoding

- Decoding in kernel space using BPF programs
- Feature was merged into the kernel in 2019
- Original plan was to write user space while kernel BPF patches were going through review

Introducing cir

- cir: Consumer InfraRed (not IrDA)
- Parses both `.lirc.conf` files and rc keymaps
- Converts both to IRP Notation
- Can transmit and decode on command line
- IRP Notation is converted to BPF for daemon-less decoding
- Single tool that replaces `ir-ctl`, `ir-keytable` and all of lirc tooling
- Written in rust (links to llvm for BPF codegen)
- Supports pronto hex
- Single binary
- https://github.com/seanyoung/cir

Transmitting using cir (ir-ctl replacement)

cir transmit accepts the same command line arguments as ir-ctl for transmit.

- Transmit IRP:

```
cir transmit --irp
'{40k,600}<1,-1|2,-1>(4,-1,F:8,^45m)[F:0..255]'
-a F=102
```

- Transmit key from lircd remote (like irsend):

```
cir transmit --keymap PRC-100S.lircd.conf -K
KEY_MENU
```

- Transmit pronto hex code:

```
cir transmit --pronto '0000 0071 0000 000b 0020
0020 0040 0040 0020 0020 0020 0020 0040 0040 0020
0020 0020 0020 0020 0020 0040 0020 0020 0040 0020
0ca0'
```

- Transmit kernel protocol (ir-ctl compatible):

```
cir transmit -S rc5:0x1e01
```

- Transmit raw IR as-is:

```
cir transmit -r '+1000 -100 +500'
```

Decoding using cir

Decoding IR on the command line and output on terminal:

- Decode IR from receiver using lircd remote definition:
  `cir decode --keymap PRC-100S.lircd.conf`
- Decode IR using IRP with provided IR:
  ```
  cir decode --irp
  '{40k,600}<1,-1|2,-1>(4,-1,F:8,^45m)[F:0..255]'
  -r '+2400 -600 +600 -600 +1200 -600 +1200 -600
  +600 -600 +600 -600 +1200 -600 +1200 -600 +600
  -30600'
  ```
- Decode IR from file `input` using rc keymap:
  `cir decode --keymap hauppauge.toml -f input`
- Decode trying all protcols in `IrpProtocols.xml`:
  `cir decode`

Configuring keymap (ir-keytable replacement)

Convert keymap (lircd.conf or rc keymap) to IRP, BPF and then load. Supports the same command line arguments as `ir-keytable`.

- Load lircd remote:
  `cir keymap --keymap PRC-100S.lircd.conf`
- Load rc keymap remote:
  `cir keymap --keymap hauppauge.toml`
- Create IR decoder based on IRP and load:
  ```
  cir keymap --irp
  '{38.4k,564}<1,-1|1,-3>(16,-8,D:8,S:8,F:8,
  ~F:8,1,^108m,(16,-4,1,^108m)*)
  [D:0..255,S:0..255=255-D,F:0..255]'
  ```
- rc keymaps can specify an IRP
- Presentation on Thursday during eBPF track about generating BPF from IRP using finite automations

Status

- Test suite compares IRP implementation against IrpTransmogrifier (including fuzzer)
- Test suite compares with both lirc transmit and decode for `.lircd.conf` files
- Test suite compares with kernel codecs, both transmit and decode for rc keymaps
- Found bugs in lircd, kernel and IrpTransmogrifier - all fixed upstream
- More manual testing required (e.g. does it work with cec keymaps, trying it out with a bunch of remotes, fallback to `protocols` for scancode drivers)
- No man pages or gettext translations

Path forward

- cir is pure rust (links to llvm libraries)
- cir is both a binary and a library.
- Fedora and Debian want rust to be packaged separately
- Create Fedora/Debian packages which are parallel installable to v4l-utils
- Packages should contain all the keymaps in `/usr/share/rc_keymaps` rather than `/lib/udev/rc_keymaps`
- Currently on https://github.com/seanyoung/cir - move somewhere else?
- At some point in the far future, could we remove `ir-ctl` and `ir-keytable` from v4l-utils?
- Kernel encoder/decoders are not required with cir, e.g. distribution kernels no longer require them to be enabled