Enabling LIRC to send CCF signals not residing on the server

Table of contents

Date	Description
2012-04-30	Initial version.
2015-09-10	Updated for the current version.

Table 1: Revision history

1 Introduction

Some networked devices used for sending IR codes do not store the IR signals locally, but have them sent on every invocation. This goes for e.g. the GlobalCaché IR senders. Other devices, like the LIRC server (`lircd`) keep its own data base of known signal. It can (if started with the --listen option) respond to commands of the type "send *command* from the remote *remote number* of times. It can only send those commands in its data base, typically in the file `/etc/lirc/lircd.conf` (previously `/etc/lircd.conf`). Some devices, like the IRTrans Ethernet with IR Database can both accept commands of the type remote/command and IR signals in raw form.

It is hard to say that one method is universally better than the other. "Keeping the data base on the server, not on the clients" certainly sounds like a good idea in general, however, "the server" is these days not necessarily a big, "fat" machine, and the clients are not necessarily small and dumb. A *server* is by definition something that offers services to clients, but may be small and weak in terms of resources. Furthermore, there may be several such, possibly running on comparatively weak and primitive embedded devices, with limited memory, allowing only a "small" number of flash memory updates, etc. When updating, the database needs to be distributed, possibly a protocol like ssh needs to be implemented for this, etc. Furthermore, as should be clear from my program IrScrutinizer, sometimes the use case is to experimentally try out unknown IR signals. To generate a database for this test, download it to the "server" just in order to perform the tests does not seems like a good idea.

I have submitted a previous version of the patch to the then-maintainer Christoph Bartelmus, who was vehemently against it, meaning that it is cleaner if data bases reside on the server. The issue has been discussed on the LIRC mailing list also when another person submitted a similar patch.

The here presented patch enhances the LIRC server `lircd` to be able to accept commands in the listening socket to send an arbitrary IR signal in Pronto CCF format. Mostly for illustration, the fix also patches the simple client program `irsend` to be able to send these signals.

Implementing the same functionality on WinLirc is non-trivial, since it has completely different code.

The idea of patching LIRC to allow this functionality was originally suggested to me in 2006 by Bitmonster, the creator of the original version of Eventghost.

2 Installation

The following instruction is intended for users with some experience with installation programs under Linux.

The current version is found as a [fork in the Lirc sources on Sourceforge](#).
Clone that fork using a command like `git clone ssh://bengtmartensson@git.code.sf.net/u/bengtmartensson/lirc u-bengtmartensson-lirc`. Then build using the normal instructions. (Note that these have changed somewhat the last few years.)

3 Extensions to the irsend command

The `irsend` command is contained in the LIRC distribution and (in its unpatched form) is documented [here](#). This patch enhances it with the two *DIRECTIVES* `SEND_CCF_ONCE` and `SEND_CCF_START`. Example:

```
irsend SEND_CCF_ONCE 0000 006C 0022 0002 015B 00AD 0016 0016 0016 0041 0016 0016 0016
 0041 0016 0041 0016 0041 0016 0041 0016 0016 0016 0041 0016 0016 0016 0041 0016 0016
 0016 0016 0016 0016 0016 0016 0016 0041 0016 0016 0016 0041 0016 0016 0016 0041 0016
 0041 0016 0016 0016 0016 0016 0016 0016 0041 0016 0016 0016 0041 0016 0016 0016 0016
 0016 0041 0016 0041 0016 0041 0016 0699 015B 0057 0016 0EA3
```

sends a pre-rendered NEC1, Device=122, Command=29-command to the LIRC connected device. This turns out to be the power-on command for Yamaha receivers like the RX-V 1400. Another example, assuming that a suitable Yamaha receiver is connected, the command

```
irsend SEND_CCF_START 0000 006C 0022 0002 015B 00AD 0016 0016 0016 0041 0016 0016 0016
 0041 0016 0041 0016 0041 0016 0041 0016 0016 0016 0041 0016 0016 0016 0041 0016 0016
 0016 0016 0016 0016 0016 0016 0016 0041 0016 0016 0016 0041 0016 0016 0016 0041 0016
 0041 0016 0016 0016 0016 0016 0016 0016 0041 0016 0016 0016 0041 0016 0016 0016 0016
 0016 0041 0016 0041 0016 0041 0016 0699 015B 0057 0016 0EA3
```

sends the Volume+-command (NEC1 Device=122, Command=26) "forever" (until it hits `--repeat-max`). Warning: this may have "interesting" side effects on your loudspeakers, your amplifier, your ears, or your neighbors... :-). Do not try this at home...

4 Known bugs/limitations

The handling of repetitions is not correct. This is due to a limitation in LIRC, that cannot represent a complex intro signal *and* a complex repeat signal.

It would have been a good idea if the patch also modified the LIRC version string, e.g. by appending "`_ccfpatch`".

5 Downloads

- The old version of the patch for Lirc 0.9.0. *Of historical interest only. Do not use with the current version.*