

Interaction with other projects

Table of contents

1 Revision history.....	2
2 Introduction.....	2
3 LIRC: Linux Infrared Remote Control.....	2
4 JP1.....	3
5 IRScope.....	3
6 Tonto.....	3
7 IRDB.....	4
8 wakeonlan.....	4
9 Java Readline.....	4
10 Sunrise/Sunset.....	4

1 Revision history

Date	Description
2011-01-09	Initial version, derived from old <code>harc.xml</code> .
2014-02-02	Update.

2 Introduction

HARCToolbox interacts with other projects within the area. It can be pointed out that in the case of Java projects, Harctoolbox uses unmodified jar-files; in the case of shared libraries (`.so` or `.dll`) these are also used in an unmodified state. In no case, Harctoolbox "borrows" code from the projects. Also, in this way additional functionality is implemented, none of which is of essential (like import/export of a certain file format). Differently put: should the need arise to eliminate "derivedness", only minor, nonessential functionality will be sacrificed (or needs to be implemented anew).

Also see the copyright notices in the individual programs/projects, where more third-party software is listed.

3 LIRC: Linux Infrared Remote Control

[LIRC](#) is a well established, mature, and active free software project. It is used in very many free software projects. It contains support for a large number of infrared senders and receivers, some sane hardware designs, other possibly less sane. There are also a large number of user contributed [configuration files](#) for different IR remote controls and devices, in general consisting of leaned commands. A network enabled LIRC server consists of the software running on a host, listening on a network socket, containing one or more IR transmitter or transmitter channels. A client sends requests to, e.g., transmit a certain command for a certain device type. Since Harctoolbox can talk to a network LIRC server (see source in the file `lirc.java`), there is a large number of IR senders that Harctoolbox in this way "indirectly" supports. Unfortunately, the configuration files are residing on the LIRC server only; there is no way to request the transmission of a signal the server does not know in its data base. (A patch for this was submitted by myself, but rejected by the maintainer. It is available [here](#)).

From its IR data base, Harctoolbox can generate configuration files for LIRC.

[IrScrutinizer](#), using [Jirc](#), which is a translation of LIRC to Java, LIRC files can be imported and translated to other formats.

LIRC is licensed under [GNU General Public License, version 2](#) or later. However, Harctoolbox is not a derived work; it contains no LIRC code, and is not linked to any libraries. It optionally "talks" to a LIRC server, but this functionality is entirely optional.

4 JP1

The [JP1 project](#) is a very unique project. It aims at complete control over the remotes made by Universal Electronics (UEIC), which include the brand names "One For All" and "Radio Shack". Through careful study of its hard- and firmware, techniques for custom programming a remote, equipped with a 6-pin connector (on the remote's PCB called "JP1", giving the project its name) was developed. Thus, an off-the-shelf remote can be taken much beyond its original capacities. Most importantly, it can be programmed from a computer to generate "arbitrary" IR-signals.

[RemoteMaster](#) is a program for, among other things, creating so-called device updates. These device updates can be produced by Harctoolbox, as `rmdu-exports`. Thus, for an IR-controlled device in the Harctoolbox database, a suitable JP1-enabled remote control can be made to send the appropriate IR-signals. (There are some details, that will be documented somewhere else.) RemoteMaster is presented as an interactive GUI program, however, it can also (although this is not supported) be called through a Java API. Harctoolbox presently uses version 1.89, which is not the current version. Although it seems to lack all copyright notices, it is referred to as "open source" and GPL.

IrScrutinizer can "almost" import and export RemoteMaster's device upgrade files. Since this is dependent on some rather intricate protocol dependent transformations between the JP1-specific "EFC"-numbers and the function codes, I consider it as a better idea to extend RemoteMaster to import and export the general [Girr format](#) of IrScrutinizer.

Another tool from the JP1 project is [DecodeIR](#) by John Fine, available under the [GNU General Public License, version 2](#) or later. It consists of a shared library (`DecodeIR.dll` or `DecodeIR.se`), written in C++, together with a Java wrapper (`DecodeIR.jar`). To build that jar file, also [this file](#) is needed. The tool attempts to decode an IR-signal in CCF form into a well known protocol with device number, command number, possibly subdevice number and other parameters. See the [IR protocols pane](#) in the GUI.

5 IRScope

Together with appropriate hardware, the Windows program [IRScope](#) by Kevin Timmerman is very useful to "record", and optionally analyze unknown IR-signals (again, using the same DecodeIR as above). The log files generated by this program can be directly parsed, see the code in `ict_parse.java` or the [IR protocols pane](#). The program is licensed under [GNU General Public License, version 2](#) or later. Harctoolbox neither uses code or links to it, and is not a derived work.

IrScrutinizer can replace IRScope.

6 Tonto

[Tonto](#) is normally referred to as an alternate configuration ("CCF") editor for the [first generation of the Philips Pronto](#) remote controls. It is a free replacement for the original

ProntoEdit program, written by Stewart Allen, licensed under the [GNU General Public License, version 2](#). Being written in Java, it runs "everywhere", in contrast to the original Windows-only program. It also contains a [Java API](#). Harctoolbox and IrScrutinizer use the Tonto API (in the form of the file `tonto.jar`) to import CCF files, and to generate CCF files for devices in its data base. (The latter are supposed to be more of a target for aliasing, than a directly usable user interface.) Unfortunately, the project is inactive since 2004.

7 IRDB

[IRDB](#) is a web site that describes itself as "one of the largest crowd-sourced, manufacturer-independent databases of infrared remote control codes on the web, and aspiring to become the most comprehensive and most accurate one". IrScrutinizer can directly import a set of IR signals from this data base. Interestingly, the site uses software from this site extensively.

8 wakeonlan

Harctoolbox uses [wakeonlan](#) (licensed under the [GNU Lesser General Public License](#)), a small Java library for implementing WOL-functionality.

9 Java Readline

The interactive command line uses the [Java Readline](#) (licensed under the [GNU Lesser General Public License](#)), which of course only makes sense when used in conjunction with the [GNU Readline library](#), which is licensed under [GNU General Public License, version 2](#) or later.

10 Sunrise/Sunset

Harctoolbox uses the [sunrise/sunset Java library](#) (see also [Blog](#)) by LuckyCat Labs/Mike Reedell (licensed under the [Apache License 2.0](#)). (which is considered as "compatible" with GPL3). This is a Java library for computing sunrise and sunsets. I have implemented some improvements, which I intend to publish shortly.