

Welcome to the HARCToolbox project!

home automation, remote control, infrared protocols, remotes, open source

Table of contents

1 History.....	3
2 Overview.....	4
3 Content.....	5
4 General background links.....	6

Date	Description
2011-01-10	Initial version. Creation of domain www.harctoolbox.org .
2012-05-01	Incorporated IrpMaster and IrMaster as sub projects, just as lirc2xml (moved from my personal home site) and the LIRC CCF patch (published for the first time). General reconstruction of site. Release of version 0.1.2 of IrpMaster, version 0.1.2 of IrMaster, version 0.7.0 of Harctoolbox (unfinished...), as well as the version 0.1.2 of lirc2xml.
2012-06-07	Updated for the release of version 0.2.0 of IrMaster and IrpMaster. The document on transforming XML new.
2012-08-19	Updated for the release of version 0.3.0 of IrMaster and release 0.2.1 of IrpMaster.
2012-11-18	Updated for the release of version 0.3.1 of IrMaster and release 0.2.2 of IrpMaster.
2014-02-02	Reorganized, for version 1.0.0 of IrScrutinizer, etc.
2014-06-12	Updated for the new releases: version 1.1.0 of IrScrutinizer, version 1.0.1 of IrpMaster, IrMaster, and Girr, version 0.3.0 of Jirc, version 0.9.1 of HarcHardware, version 0.2.1 of GuiComponents.
2014-09-27	Updated for the new releases: version 1.1.1 of IrScrutinizer, version 1.0.2 of IrpMaster, IrMaster, and Girr, version 0.9.2 of HarcHardware, version 0.2.2 of GuiComponents.
2015-04-16	Updated for the new releases: version 1.1.2 of IrScrutinizer, version 1.0.3 of IrpMaster, IrMaster, and Girr, version 0.9.3 of HarcHardware.
2015-09-10	Updated for the new releases: version 1.1.3 of IrScrutinizer.
2016-01-08	Arduino hardware new pages. Glossary extended and improved.
2016-04-20	Updated for the new releases: version 1.1.3 of IrScrutinizer. New page IR resources . Glossary extended.
2016-05-04	New page Infrared2Arduino and its API docu.
2016-08-30	New version 1.3 of IrScrutinizer .
2017-03-16	New version 1.4 of IrScrutinizer .

Table 1: Revision history

Note:
New release of IrScrutinizer 1.4.

1 History

Since 2006 I have been writing software, designed file formats, and classified remote control signals, revolving around infrared remote control and home automation. The [old "main project" \(old harctoolbox\)](#) is the original project. It deals with descriptions of IR protocols and signals, device descriptions, and descriptions of their interconnection.

In early 2011 I came to the conclusion, that my work on IR protocols was a dead end street. It was better to connect to and use the Internet knowledge, notable the so-called IRP notation, that was excellently [formalized](#) by Graham Dixon in early 2010. This started the [IrpMaster](#) project, in which the IRP notation was completely implemented (with one [exception](#), which is rather a specification flaw than an omission of practical importance).

For IrpMaster, the goal was correctness and completeness, without any compromises for user friendliness. In particular, it does not contain a graphical user interface. Its command line interface appears as inaccessible to the GUI-centered user. Instead, the project [IrMaster](#) was started, to allow a user friendly, interactive access to the functionality of IrpMaster. As a graphical "meta program", it also fulfills some other tasks, see its documentation.

Early 2013 I planned to extend IrMaster with more functionality, in particular the possibility to capture IR signals and to import IR signals from many other sources and file formats. Originally, I planned to extend IrMaster with this new functionality; however, I decided to start on a brand new program, [IrScrutinizer](#), which basically combines almost all that fits in an IR program. One of the reasons was that I was unhappy with the somewhat suboptimal code structure of IrMaster. The first version was released on November 2013. IrMaster was declared as no longer developed, promoted to version 1.0.0, which was released in February 2014.

In the light of these projects, the main project harctoolbox has gone slightly "stale", and it needs to be restructured.

The program *lirc2xml* was posted on the LIRC mailing list in 2009 (without any response :-), and has been residing on my private web page since then. For IrScrutinizer, I wanted this functionality incorporated. A natural way would be to turn lirc2xml to a shared library with [JNI](#) interface. However, discouraged by the experiences from [DecodeIR](#), I wanted a pure Java solution, and ported the part of LIRC needed to parse and interpret the configuration files to Java. This is the [Jirc](#) project.

[LIRC CCF](#): The idea to modify the LIRC server to allow for directly sent CCF signals was originally sent to me in private mail from Bitmonster (of Eventghost). I made a

preliminary version some time later, this was communicated to him and submitted to the then-maintainer Christian Barthelmus, who promptly rejected it (some more details in the documentation). This is its first real "publishing".

There are presently over 80000 lines of source code in Java on this site, code in other languages, or by other authors not counted.



The present work is copyrighted by myself, and available under the [GNU General Public License version 3](#). In the future, it may also be available under additional conditions, so-called dual licensing. (If interested in a commercial license, please contact me.) File formats are in the public domain, including their machine readable descriptions, i.e. DTD and schema files.

As a working project name, as well as in the Java module names, the name *Harc*, which is simply an acronym for "Home Automation and Remote Control", was used. Unfortunately, that name is far too generic to register as an Internet domain. There is even a Sourceforge project named Harc, (inactive since 2002). For this reason the project was renamed to *HARCToolbox* early 2011, and the domain www.harctoolbox.org created.

2 Overview

[IrpMaster](#) is a very advanced renderer of infrared signals. That is, from an abstract description consisting of a protocol name, and some parameter values, it computes the corresponding infrared signal, as a sequence of gaps and flashes. It is intended to be used both from its API, as well as an interactive (command line) program, both offline (generating files etc) as well as in real-time applications (generating signals to be sent to hardware equipment). The program does not attempt to be "user friendly", and does not, e.g., contain a graphical user interface.

[Irmaster](#)'s main purpose is to be a user friendly user interface to IrpMaster. But it does not end there: it also integrates/interfaces with other projects, like the alternative, "classical" IR renderer Makehex, the DecodeIR "inverse renderer", the analyzeIR library, the Harctoolbox main project etc. It can generate export files in different formats, it can address networked hardware directly (presently GlobalCaché, IRTrans LAN module, and a [patched](#) LIRC server. (Please note that *IrMaster* and *Irpmaster* are two different projects. IrMaster depends on IrpMaster, but not vice versa.)

[IrScrutinizer](#) is a powerful program for capturing, generating, analyzing, importing, and exporting of infrared (IR) signals. For capturing and sending IR signals several different hardware sensors and senders are supported. IR Signals can be imported not only by capturing from one of the supported hardware sensors (among others: IrWidget, Global Caché, and Arduino), but also from a number of different file formats (among others: LIRC, Wave, Pronto Classic and professional, RMDU (partially), and different text

based formats; not only from files, but also from the clipboard, from URLs, and from file hierarchies), as well as the Internet IR Databases by Global Caché and by IRDB. Imported signals can be decoded, analyzed, edited, and plotted. A collection of IR signal can thus be assembled and edited, and finally exported in one of the many supported formats. In addition, the program contains the powerful IrpMaster IR-renderer, which means that almost all IR protocols known to the Internet community can be generated.

There are some support projects, independent in the sense of a separate package. First, there is the subproject [HarcHardware](#), containing a number of classes for hardware access. This is believed to be of interest also for other projects. Secondly, the subproject GuiComponents (no documentation page present!) contains a number of, in principle, recyclable components, often in the form of Java Beans. Although it was written and conceived as a support project for IrScrutinizer, it is not unlikely that they can be used directly in other programs.

As described in the introduction, the [main project](#) is presently slightly "stale".

In the context of the [LIRC](#) project, there is a large body of recorded infrared signals in "LIRC format". For many users of home automation and remote control, it would be desirable to use this knowledge outside of the LIRC framework. Unfortunately, the LIRC format (`lircd.conf`) is all but well documented. The program [lirc2xml](#) therefore uses LIRC itself to decode and export its signals, here to (somewhat arbitrarily) an XML file, that can be further processed.

Finally, the [LIRC patch for CCF](#) signals describes an enhancement to the LIRC server.

3 Content

- [Arduino Nano IR hardware](#), [part 1](#) and [part 2](#).
- [IrScrutinizer](#), very powerful general purpose IR program.
- [IrMaster](#), a predecessor to IrScrutinizer.
- [IrpMaster](#), the IR renderer.
- [Glossary and terms](#).
- [HarcHardware](#), hardware related functionality.
- [Girr, a specification](#) of a general exchange format for IR signals
- [Generic IR Server](#), a specification of a general IR server API
- An annotated list of [IR resources](#) on the Internet.
- [HarcToolbox](#), the old main project.
- [Transforming XML Export from Ir\(p\)Master](#), a tutorial article on generating "interesting stuff" (here, C code) from the XML export. Applies to IrpMaster and IrMaster, not IrScrutinizer, thus somewhat obsolete.
- [Jirc](#), LIRC configuration files in Java.
- [Lirc2xml](#), a program for extracting IR codes from LIRC files. Predecessor to Jirc, thus obsolete.
- [Improved lirc driver](#) for the Raspberry Pi.
- [Raspberry Pi daughter board for IR and RF](#) transmission and reception.

- [LIRC CCF patch](#), to make the LIRC server more usable.
- [Relationship to other, similar projects](#).
- [Downloads](#).

4 General background links

- [SB-Projects' IR Remote Control Theory](#). A very good place to start leaning about IR protocols. Covers the basics of infrared signals, and its modulation. Not too long, not too "dummy"-oriented.
- [JP1 wiki](#) Many articles, of different ages and qualities, are collected here, in a clearly arranged way.