

InfraRexx

Leon Woestenberg

Copyright © Copyright 1994-1995 by Leon Woestenberg

COLLABORATORS

	<i>TITLE :</i> InfraRexx		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Leon Woestenberg	August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	InfraRexx	1
1.1	InfraRexx	1
1.2	InfraRexx Copyrights and Disclaimers	2
1.3	InfraRexx Introduction	2
1.4	InfraRexx Features	3
1.5	InfraRexx History	3
1.6	InfraRexx Known Bugs	5
1.7	InfraRexx Things to do	6
1.8	InfraRexx Software	6
1.9	InfraRexx Editor	7
1.10	InfraRexx Editor Mainwindow	7
1.11	InfraRexx Editor Editwindow	9
1.12	InfraRexx Daemon	11
1.13	InfraRexx ARexx Interface	12
1.14	InfraRexx Hardware	13
1.15	About InfraRexx...	14
1.16	The beginning of InfraRexx...	15
1.17	InfraRexx Credits	15
1.18	InfraRexx Development...	16
1.19	InfraRexx Future...	17
1.20	Localizing InfraRexx...	17
1.21	How to order InfraJoy...	18
1.22	How to register to InfraRexx...	19
1.23	How to pay for what you get...	19
1.24	InfraRexx Registration/Order Form	20
1.25	Obtaining Updates...	21
1.26	Reaching the authors...	22

Chapter 1

InfraRexx

1.1 InfraRexx

```
InfraRexx 1.6b © 1994-1995 by Leon Woestenberg (Digital ↔  
Disturbance)  
~~~~~ and Jeroen Steenblik
```

Infrared Remote Control ARexx Interface

General Information

Legal notes

History

Introduction

Known Bugs

Features

Things to do

The User Help Guides

The software

The hardware

Very Much More About InfraRexx

About InfraRexx...

Much more that you would like to know.

Reaching the authors...

For smashing bugreports and the like.

1.2 InfraRexx Copyrights and Disclaimers

InfraRexx Copyrights

InfraRexx - The
software
archive

Copyrights © 1994-1995 by Leon Woestenberg and Jeroen Steenblik. InfraRexx is freely distributable providing that the archive is kept intact. A nominal fee may be asked to cover distribution costs. This is known as freeware.

InfraJoy - The
hardware
design

Copyrights © 1994-1995 by Jeroen Steenblik and Leon Woestenberg. The hardware design is freely distributable and may be used to build a copy of the InfraJoy interface hardware. A copy of the hardware itself may not be sold with a profit.

Both InfraRexx and InfraJoy may not be sold, nor included with any commercial product without written permission of the
author(s)

.

InfraRexx Disclaimers

The authors do not take responsibility for the InfraRexx
software
or
hardware
damaging your Amiga computer, infrared controlled devices or data stored on these machines. The authors can not guarantee that InfraRexx and/or InfraJoy work(s) correctly on your configuration.

1.3 InfraRexx Introduction

InfraRexx Introduction

InfraRexx is a
hardware
and
software
project that allows your Amiga
to control external devices and vice versa, via infrared signals.

ARexx commands sent to the
InfraRexx Daemon
commodity, cause
corresponding infrared commands to be sent to external devices. Vice versa, infrared signals recognized by the Daemon can control software via ARexx.

The separate

InfraRexx Editor
program is used to learn new infrared
command codes, and to attach meaningful ARexx commands to them.

A possible use of InfraRexx is the control of audio/video devices, such as television sets, video tape recorders or compact disc players, via multimedia software running on an Amiga.

Another possibility is to control a module player running on your Amiga so that you can adjust volume and select music modules via a common infrared remote control unit as found with modern audio/video consumer electronics.

1.4 InfraRexx Features

InfraRexx Features

- Uses (and therefore needs) Kickstart release 2.04 (or higher).
- Easy and allround control of devices via
ARexx
- Control your Amiga programs via a remote control unit and ARexx.
- Small
commodity daemon
to save memory.
- Fully multi-tasking during infrared transmission and reception.
- Low system overhead while infrared monitoring (can also be disabled).
- Font-sensitive and style-compliant graphic user interface.
- Localization under Workbench release 2.1 (and higher) systems.
- Newlook menus under AmigaOS release 3.0 (and higher) systems.
- Uses the Amiga Installer Utility for maximum convenience.
- Lowcost
hardware
with optional joystick throughput.
- New hardware design with large transmission range, that supports most remote controllers around, because of multifrequency features.

1.5 InfraRexx History

InfraRexx History

Release 1.0 (11-May-94)

- Initial release onto Aminet.

Release 1.1 (4-Jun-94)

- Interim release for registered InfraJoy v2 hardware users.

- InfraRexx Editor v1.1 (4-Jun-94)
 - Adapted transmitter code to control v2 hardware as well.
 - Now learns code bitstreams with long headers (up to 24 bit).
 - Removed incompilant copyright notice from the versionstring.
- InfraRexx Daemon v1.1 (4-Jun-94)
 - Adapted transmitter code to control v2 hardware as well.
 - Removed incompilant copyright notice from the versionstring.

Release 1.2 (15-Jul-94)

- Released on Aminet and soon on a Fidonet support BBS.
- New language catalogs v2 are no longer in subdirectory.
- Includes nice building plans for the new InfraJoy v3 hardware.
- InfraRexx Editor v1.2 (28-Jun-94)
 - Waitpointer (using 3.x functions if present) during busy time.
 - Removed harmless bug that could prevent a window from opening.
 - Removed a crashful bug, dangerous when sending with high repeat.
 - Changed interrupt names to clear up who is using the CIA timer.
 - Now needs v2 of the language catalog, no longer in subdirectory.
 - Specific transmit/receive code for each of the v1/v2/v3 hardware.
- InfraRexx Daemon v1.2 (27-Jun-94)
 - Changed interrupt names to clear up who is using the CIA timer.
 - Now needs v2 of the language catalog, no longer in subdirectory.
 - Specific transmit/receiver code for each of the v1/v2/v3 hardware.

Release 1.3 (19-Aug-94)

- Released on Aminet, and Fidonet support BBS. Submitted to Fred Fish.
- InfraRexx Editor v1.3 (10-Aug-94)
 - Removed bug that made learning only work once on certain machines.
 - Previously, fast multiple clicks on the edit button could cause the program to open multiple edit windows.
 - Checkboxes are scaled to the font height under AmigaOS 3.x.
- InfraRexx Daemon v1.3 (17-Aug-94)
 - Removed bug that made sending only work once on certain machines.
 - Removed a test line of v1.2, that caused to send a code at startup.
 - If the locale library was present, we could run into illegal memory.
 - String compare didn't check for equal lengths first.

Release 1.4 (24-Oct-94)

- Released on Aminet, and Fidonet support BBS.
 - Fixed a missing track on the solderside PCB layouts.
 - Included Italian and Danish language catalogs.
 - InfraRexx Daemon v1.6 (14-Oct-94)
 - Codes were not sent with own bitrate but used the codeset bitrate.
 - Joystick port lines are set to input mode after usage.
 - Compiled using Amiga E 3.0b (took a split second), smaller program.
 - InfraRexx Editor v1.4 (23-Oct-94)
 - Joystick port lines are set to input mode after usage.
 - Removed Enforcer hits occuring when opening editwindow with new code.
 - Compiled using Amiga E 3.0b (within 1 second ;-), smaller program.
-

Release 1.5 (27-Mar-95)

- Released on Aminet, and submitted to Fred Fish.
- Included Swedish language catalogs.
- Many small bugs fixed, not specifically mentioned below.
- Transmitter/receiver code now adapts timing to NTSC/PAL machines.
- Included REXXEvent 1.17, to attach ARexx commands to events.

- InfraRexx Daemon v1.6 (25-Mar-95)
 - Added ARexx SET SCANFREQ option, to set monitor scanfrequency.
 - Added DEFRECEIVE arguments, to respond to unrecognized codes.
 - Defining a CX_POPKEY caused an Enforcer hit and sometimes failed.
 - A recognized code without receive command caused Enforcer hit.
 - The commodity now acts as if it has a GUI: the control requester. It can appear now, and does so when a new daemon is started.

- InfraRexx Editor v1.5 (25-Mar-95)
 - Added keyboard shortcuts for button and cycle gadgets.
 - Stringgadget changes weren't saved if not confirmed with an enter.
 - Save requesters weren't real save requesters, with makedir fancy etc.
 - Shell HARDWARE keyword wasn't processed, caused hardware not to work.
 - Forgot to set codewindow bitrate slider to correct value.

Release 1.6 (5-Jun-95)

- Released on Aminet.
- New hardware v4.0 release with design plans, features auto-frequency selection and larger transmission range.
- Added german and norwegian catalog translations.

- InfraRexx Daemon 1.7 (5-Jun-95)
 - Added support for modulation frequency sensitive hardware.
- InfraRexx Editor 1.6 (22-Apr-95)
 - Added support for modulation frequency sensitive hardware.
- REXXEvent 1.21
 - Now handles multiple events in one call.

Release 1.6a (14-Jun-95)

- Released on Aminet.
- Fixed a bug in, and somewhat clarified the building schematics.

Release 1.6b (14-Aug-95)

- Released on Aminet.
- Fixed a bug in the printed circuit board design, on the +5V throughput.
- Clarified the building instructions and pictures concerning the SFH.

- InfraRexx Editor v1.7 (13-Aug-95)
 - The mainwindow didn't support listview dragscrolls.

- InfraRexx Daemon v1.8 (26-Jun-95)
 - The shell 'HARDWARE' argument was broken. Fixed.

1.6 InfraRexx Known Bugs

Known Bugs

There is a known bug affecting unnamed codes; Intuition tries to render

those into the listview gadget and runs into zero pointers, causing an Enforcer hit on each.

All other bugs are removed with this release. So, if you experience a bug, please report it to us, if possible with debug tools' output information.

1.7 InfraRexx Things to do

Things to do

As you read this, I'm developing the successor of this program, which will be a complete rewrite with many features (see future). Therefore, this program will not be enhanced significantly the coming time, but will later be replaced by a complete new one. Found bugs will be removed, however.

Let

us know if you have any ideas for the new program.

1.8 InfraRexx Software

InfraRexx Software

The InfraRexx software mainly consists of two programs:

- The

InfraRexx Editor provides a graphic user interface to edit a codeset. New infrared codes can be learned, and ARexx commands can be attached to each of the codes.

- The

InfraRexx Daemon is a small commodity that runs in the background. It waits for incoming ARexx commands that are sent to its ARexx host port, and sends the infrared code attached to each of these commands. Optionally, it is possible to enable an infrared monitor, that waits for infrared signals and sends the attached ARexx command to the ARexx Master.

Some codesets and example ARexx scripts are included to show the power of InfraRexx in conjunction with other programs.

1.9 InfraRexx Editor

The InfraRexx Editor

The editor is a program where you can build up an infrared codeset. In the main window, a description of the codeset and default values can be given. The infrared codes can be edited by the code editor subwindow, where code-specific entries can be selected.

The main window

The code edit window

Three options can be passed to the editor, using either command `↔` line

arguments (from the Shell) or icon tooltypes (from the Workbench). The supported entries for both are:

- FROM
Defines a codeset file to load. Default is no file.
- PUBSCREEN/K
Sets the preferred public screen where the editor should open on. The editor normally opens on the default public screen.
- HARDWARE/N
Selects the InfraJoy hardware version that is used. Default is 4.

1.10 InfraRexx Editor Mainwindow

InfraRexx Editor Mainwindow

In this window you can edit an infrared command codeset. A codeset is the collection of functions found on a common remote control unit. It is built up of some global settings (found in the mainwindow) and command-specific data (found in the

editwindow

). To set the global

parameters, use the following gadgets...

- Codeset description
Here you can define the name and additional information of the current codeset.
- Bitrate (bits per second)
Use this slider to set the bitrate of the codeset. This value is very important because it indicates how fast the infrared signals should be sampled. Before learning any commands, be sure to set this value to the bitrate that is used by the remote control unit.
- Modulation (Hertz)
Controls the modulation frequency, used to surpress interference from other IR sources, such as lamps and sunlight. This frequency is brand

specific.

This feature needs hardware version 4 or better. With older hardware, you have to adjust this manually on the hardware device. This value will be ignored if you have such an older hardware version.

- Monitor frequency (times per second)
This value indicates how many times per second the daemon commodity should check for incoming infrared signals. The higher the value the more processor time the commodity will take, but the quicker the response is. 100 times per second seems to be a nice balanced setting.
- Infrared codes
This is the listview where all the learned codes are shown. Select a code here to make it the currently edited code. Use these buttons to edit the list...
 - New
Adds an empty code to the list, ready to be learned or edited.
 - Edit...
Opens the
 edit window
 with the selected code.
 - Delete
Deletes the selected code.
 - Sort
Sorts the list alphabetically.
 - Top, Bottom, Up, Down
Edit the position of the selected code in the list.

The pulldown menus are used to store the current project, the codeset, to disk, or to quit the editor.

- Project
 - Load...
Opens a filerequester in order to load a codeset from disk. If there are still unsaved changes, you will be asked to proceed first.
 - Save
Saves the current codeset project to disk using the current name.
 - Save as...
Opens a filerequester in order to save the current codeset to disk.
 - About...
Displays information about InfraRexx.
 - Quit...
Quits the editor program. If there are still unsaved changes, you will be asked to proceed, save before quitting or cancel.
-

1.11 InfraRexx Editor Editwindow

InfraRexx Editor Editwindow

The editwindow is the place to be when editing the infrared commands you want to use via

```
ARexx
```

. Here you interface the current infrared signal code and ARexx commands. Use these gadgets to do so:

- Infrared Code Interfacing

In this frame, the user side of the interface can be edited.

- Infrared Code Name

Enter a meaningful name of the current code here. For example, 'CD Player Shuffle' or 'Amplifier Select Tuner' (without quotes).

- ARexx Transmit Command

Enter a unique keyword here for transmitting the current code. For example (see above), 'CD_SHUFFLE' or 'AMP_SELECT_TUNER'. When sending this keyword via

```
ARexx
to the
daemon
, the infrared
```

code will be sent, and for example, your CD Player will go into shuffle mode, or your amplifier selects the tuner as input.

- ARexx Receive Command

Enter an

```
ARexx
```

command here that will be executed when receiving this code. In this way, you can control your Amiga programs via your infrared remote control unit. Use SINGLE QUOTES around multiple keyword arguments.

You can also type the name of an ARexx script here, without quotes. Either this script should be in the REXX: directory, or you should give the full pathname of the script.

For example, use your (CD Player) remote control unit to control the moduleplayer DeliTracker 2.x. Select the 'CD Next Track' code and add this ARexx Receive Command: 'ADDRESS DELITRACKER NEXTSONG'. When you press the 'next track' button on the remote, the next module of the list will be played.

- Transmit bitrate

Defines the bitrate for the current code only. In this way it is possible for a code to have a different bitrate than the codeset. Note that codes with a different bitrate can only be sent, and cannot be received by the

```
daemon
```

.

- Use Codeset bitrate

Makes this code use the same bitrate as the codeset.

- Bitrate slider
If the previous gadget is not checked, you can define a different bitrate here for the current code.
- Infrared Code Data
The place where the infrared signal can be edited.
 - Data Locked
Checking this gadget will keep you from accidentally modifying the learned code.
 - Infrared Bitstream
The actual infrared signal bitstream using 1's and 0's. You should not edit this gadget unless you know what you are doing.
 - Delay Between Codes
This indicates the number of 0 bits that should be sent inbetween two successive code bitstreams. Normally, this delay is around the same size as the bitstream itself.
 - Code Repetition
The number of times the code should be repeated. Normally, 3 to 5 times should be sufficient to guarantee a clear response from the controlled device.
- Code Learner
This is the place where codes are learned directly from the remote control unit. This needs the receiver hardware to be present.
 - Learn
Start learning a new code. The following messages instruct you what to do...
 - Waiting for IR signal...
~~~~~Direct your remote control unit towards the receiver, at about  
~~~~~1 meter (3 feet) distance. Now send the code by pressing the  
~~~~~function button you want to learn on your remote control unit.
    - Learning IR signal...  
~~~~~An infrared signal has been received and is now being processed.
 - Bad signal, not learned.
~~~~~This signal was received bad, and was not learned. Please try  
~~~~~again. If this occurs often, check the batteries in the remote  
~~~~~control unit, or try different distances/angles.
    - Timeout, nothing learned.  
~~~~~Five seconds expired before any infrared light was received.
 - If no message appears after learning, the code learned is shown
~~~~~into the bitstream gadget. If the code is nonsense (lot of 1's,  
~~~~~or 0's) you have to learn the code again. If the code looks OK,  
~~~~~try it out by sending it, and check if the appropriate device  
~~~~~responds to the infrared signal. If not, learn again.
 - Send

Transmit the current code with the current settings. This can be used to immediately check if a newly learned code is correct.

Note that if InfraRexx Daemon is running, its monitoring function must be disabled. Otherwise, the editor will report that it cannot use the CIA timer for sending, as the Daemon program is using it.

1.12 InfraRexx Daemon

The InfraRexx Daemon

This commodity serves as the actual
ARexx-to-infrared interface
. This

daemon uses as few memory and processor time as possible. Therefore, no graphic user interface is available except for a requester window where you can control the infrared monitor or quit the daemon.

InfraRed Monitoring

The infrared monitor is an interrupt that checks for incoming infrared signals on a regular basis. The monitor can be disabled, if you don't need to watch for incoming infrared signals. In this way, the daemon doesn't use processor time (until it receives activation signals).

Re-running the daemon while already running will pop up a requester where you can decide to either enable, disable or quit the InfraRexxDaemon.

The daemon is configured using either command line arguments (from the Shell) or icon tooltypes (from the Workbench). The supported entries for both are:

- CX_PRIORITY/N/K
Sets the commodity priority. Default this is zero.
 - CX_POPKEY/K
Defines the hotkey to popup the monitor requester. The default is 'control alt i'.
 - FROM/A
The acquired codeset file where all the infrared data is stored. This file can be edited and saved using the
InfraRexx Editor
.
 - MONITOR/K
Sets the initial state of the monitor. Either 'YES' for enabled or 'NO' for disabled. Default is YES.
 - PUBSCREEN/K
Sets the preferred public screen where the monitor requester should popup. Normally, it pops up on the default public screen.
 - HARDWARE/N
Selects the InfraJoy hardware version that is used. Default is 4.
-

- DEFRECEIVE/K
 Defines a default receive ARexx command that is executed when the incoming infrared code is not recognized. Use single quotes to group multiple commands. From the shell, also use double quotes, like this for example: DEFRECEIVE="'ADDRESS DELITRACKER EJECT'".

1.13 InfraRexx ARexx Interface

InfraRexx ARexx Interface

The

InfraRexx Daemon

communicates with other programs via ARexx. The name of the ARexx hostport is 'INFRAREXX'. Programs supporting ARexx or ARexx scripts may send commands to this port to control external devices via infrared codes. There are three types of ARexx commands it supports:

- Transmit commands that can be edited via the
 InfraRexx Editor
 . Sending
 these commands to the daemon, will make the daemon transmit the corresponding infrared signal from the used codeset.
- Receive commands are send by the daemon if it recognizes an incoming infrared signal that is sent by an infrared remote control unit. These receive commands will be send to the ARexx Master, but can also be directed to other programs using the ARexx 'ADDRESS' command.

You can also execute an ARexx script in this way . Just save a script in the REXX: directory with a '.irx' suffix, and this script will be executed when the receive command of a code equals this script name.

Receive commands are edited from the
 Codeset Editwindow
 of the

InfraRexx Editor
 program.

· Control commands which control the configuration of the daemon. The templates of the daemon control commands are:

- QUIT
 Quits the daemon.
- SET MONITOR/K
 Sets the configuration of the daemon:
 - MONITOR/K
 Enables or disables the monitor. Either 'YES' or 'NO'.
 - SCANFREQ/N/K

Set monitor scanfrequency.

- SEND BITRATE/N/K STREAM/A/K DELAY/N/K REPEAT/N/K MODULATION/N/K
Send a raw infrared signal as specified by these parameters:
 - BITRATE/N/K
The rate at which this signal must be send. If not specified, the default codeset rate will be used.
 - STREAM/A/K
This is the actual signal, a bitstream built up of ones and zeros.
 - DELAY/N/K
The delay (in number of bits) between the bitstreams. In this period, no infrared signal is sent.
 - REPEAT/N/K
The number of times that the bitstream is sent. With this parameter you can control the time that normally a button would be pressed on your remote control unit. This can be used to control audio volume or tape cueing functions. For normal functions, however, a value of around 3 should be good enough to ensure clear transmission.
 - MODULATION/N/K
Controls the modulation frequency, used to surpress interference from other IR sources, such as lamps and sunlight. This frequency is brand specific. Select one of these:
1 (approx. 32000 Hertz), 2 (approx. 36000 Hertz), 3 (approx. 38000 Hertz) or 4 (approx. 40000 Hertz).

1.14 InfraRexx Hardware

InfraRexx Hardware

The InfraRexx hardware, named InfraJoy, is a small circuitry that is plugged into the second joystick port. You may use the hardware plans of this archive to build your own hardware. Note the

copyrights

when

doing so. The hardware can also be

ordered

by the

author(s)

.

The hardware is based on a small print circuit board, with a size of about 5 x 2.5 cm (2 x 1 inch), that can optionally hold a transmitter and/or receiver.

- The transmitter sends infrared signals as directed by the InfraRexx software. This can either be the

Daemon

(when processing ARexx

transmit commands
) or the
Editor
(using the
send
function).

Note that if you have a transmitter-only version of the InfraJoy hardware, you are able to control devices via your Amiga. However, to control your Amiga using a common infrared remote control unit, you need a receiver as well (see below). This is also true if you want to learn new codes.

· The receiver is based on a hybrid infrared detector, that will detect infrared codes sent by remote control units, and send these to the InfraRexx software. This can either be the

Daemon
(when
monitoring
) or

the

Editor
(when
learning
).

Note that if you have a receiver-only version of the InfraJoy hardware, you are able to control your Amiga via an infrared remote control unit. To be able to control devices via your Amiga, you need the transmitter as well (see above).

Although the transmitter-only and receiver-only versions can be used without problems, the most powerful solution is a combination of both.

1.15 About InfraRexx...

About InfraRexx...

The beginning of InfraRexx...
How it all started.

Credits
Are you in here too? Just have a look who is.

Development
What can you do to make it happen in the future?

Future
A look ahead of what is still to come in this world.

Ordering the hardware
How to order the InfraJoy hardware interface.

Paying the dimes
How to get your money at the right spot.

Registering

How to become a registered user of InfraRexx, and why.

The Registration/Order Form

Fill this in, print it, and post it now :)

Obtaining updates

How to get the hottest, most cool release version.

1.16 The beginning of InfraRexx...

The beginning of InfraRexx...

InfraRexx is a result of an idea I had in mind for months. I wanted to control my audio set via the Amiga, so that I could implement music and sound effects to video presentations. However, during the time many more ideas came above, like:

- a GUI to control audio and video devices (inspired by JukeBox).
- voice-recognized control for lamps, audio/video (using VCLI).
- automatic CD recordings, using editing, peak search, fading etc.
- a discjockey program (inspired by those giant radio channels).
- a small infrared communications project to make Amiga's interact.
- a way to control my Amiga programs via a remote control unit.
- selecting a radio/tv news channel on specific times (using Recall).

As I had no idea where to start, I decided to design a basic interface which implements to basic functions needed to make the link between infrared light signals and Amiga applications. This is where the InfraRexx project started. In fact it does none of the things mentioned above, but serves as an

ARexx

host that offers a large amount of

possibilities.

The name 'InfraRexx' was made up from two wonderful things: 'Infrared' and 'ARexx'. 'InfraJoy' added some 'joy(stickport)' to it :)

1.17 InfraRexx Credits

InfraRexx Credits

Thanks go to the following persons:

- Wouter van Oortmerssen for the E programming language and compiler.
- Dietmar Eilert for the excellent programmer's editor GoldED.
- Claus Østergaard for the danish translation.

- Michele Giorato for the italian translation.
- Michael van Elst for the german translation and some clear and useful bugreports. (Not all bugs have been removed, but nobody knows :-)
- Johan Dahlberg for the swedish translation. Also thanks to other swedish translators who spontaneously sent in translations.
- Odd-Jarle Kristoffersen for the norwegian translation.
- Kevin Phair for reporting the missing listview feature, that went unreported (unnoticed?) all up from version 1.0 :-)
- All registered users and users that gave feedback, bug reports, ideas, and new codesets.
- Those who included 2.04+ example sources with their software, especially Stefan Becker for ToolManager, Olaf 'Olsen' Barthel for Term, and Eddy Carroll for SnoopDos. Thanks chaps!
- The Amiga hardware and software (ex-)engineers for this superb computer and operating system.
- Especially Dave Haynie, for giving some hardware background information.
- All Amiga dudes and dudettes. Keep the Amiga flame burning hot.
- The Escommeteers who develop the Amiga technology beyond our dreams...

1.18 InfraRexx Development...

InfraRexx Development...

The InfraRexx project depends heavily on the availability of infrared code information as used by the electronics manufacturers. Most of the manufacturers don't document the codes for public. Happily, Philips does, so the project could be tested with the specs in our hands.

In future releases, I would like to offer more ready-to-use codesets. Therefore, I need the specs of the infrared codes used. Although codes can be learned by just experimenting, many things remain unclear. The most important spec is the so-called bitrate of the code.

Also, I would like to see the

InfraRexx software
being localized for

different countries. As I don't speak all the worlds languages, I would like it if people want to localize the catalogs, and send them to me. Please read the

localization notes
if you are interested.

Also, I would like to hear some feedback from InfraRexx users. Maybe you've got an exotic use or cool

ARexx
script that you want to share
with other users.

If you think you can contribute to the
future
of InfraRexx, just drop
me a (electronic)
mail
.

1.19 InfraRexx Future...

InfraRexx Future...

I want InfraRexx to become the perfect infrared interface for the Amiga, which is system- and userfriendly and gives good performance. Things I have in mind for further improvement are:

- more ready-to-use codesets and more supported brands.
- a new client/server model, with Prefs editor, etc.
- not only ARexx, but also Command and HotKey events; new name.
- a more modular approach to the codeset and code implementations.
- a device independant way of controlling audio/video devices, so that an application can transparently control any device.
- increased infrared transmission range, really a hardware thing.
- hardware for parallel port and maybe serial port or even power LED.
- IFF codeset file definition, fast asynchronous loading and saving.
- improved graphic user interface, aspect-ratio (hi/lowres) sensitive.
- context sensitive AmigaGuide online help support.
- maybe an extra Magic User Interface version. (MUI © by Stefan Stuntz).
- many more usage ideas and implementations.
- everything YOU suggest...

I hope to implement many of the above features in an complete rewrite of the programs, with a much more modular approach in an object oriented way.

1.20 Localizing InfraRexx...

Localizing InfraRexx...

If you want to see the InfraRexx messages in your own language, why don't you translate the messages yourself? I would be very glad if people sent me their localizations so that they can be included in future releases of InfraRexx. You need some knowledge of the localization procedure, but the following guidelines help you along.

Localization guidelines

The InfraRexx software uses the locale library and language catalogs to adapt to the user's locale preferences. Catalog description files (.cd), as well as empty catalog translations files (.ct) are included

in the InfraRexx archive. Use these to create a locale catalog of your language, using a catalog compiler such as CatComp.

A more easy way is to use CatEdit by Rafael D'Halleweyn, a complete catalog editor with a very good GUI for editing the catalog strings.

Follow these simple guidelines to generate a correct translation:

- know the features of the software.
- fill in the header of the translation file.
- translate EVERY sentence in the translation file.
- NOT change the order of the catalog strings.
- be sure that every sentence is CORRECTLY and CLEARLY translated.
- pay attention to the spaces around GUI strings (frames, sliders etc.).
- use a texteditor that does NOT wipe out the end-of-line spaces.
- adapt your translation so that the windows fit on a 'NTSC Hires Nonlaced' screen of regular size (640x200) using the 'Topaz 8' font.
- Add this line to the STRING_ABOUT message (translated of course).

```
\n\n<Language name> translation by <Your Name>
```

where <Language name> is the name of the language you translated to.
and <Your Name> is the name of the translator, yes, you!

If you think you made a nice translation, please send the translation file to one of the

authors

. Also state if you would like to translate future versions of the software, or not.

1.21 How to order InfraJoy...

How to order InfraJoy...

InfraJoy is the

hardware

part of InfraRexx and is needed by the

software

. Full plans and instructions are included with this archive to build the hardware yourself. However, the hardware can also be ordered via the

authors

by filling in the

registration/order form

.

Amiga Model

Please state the Amiga model the hardware will be used with, so that the receiver can be placed according to the joystick port direction.

Optimisation

Since InfraJoy v2, we are now able to, more or less, optimise the hardware for the electronics brand you use. This gives you a somewhat better response on the Amiga side. This is just a finetune; other brands should work as well, but may become somewhat less responsive on large distances.

Please allow two to six weeks before delivery of the InfraJoy hardware.

1.22 How to register to InfraRexx...

How to register to InfraRexx...

First of all, InfraRexx is freeware. You may use it to its full extend without paying a single dime, and without missing one single option.

If you think InfraRexx is good or if you built the hardware yourself and the stuff actually works ;-), please register the software.

To register, fill in the
registration/order form
and send it, along

with the

payment
, to one of the
authors
.

Registered users stay registered even if InfraRexx may become shareware in the future. The register fee is NLG 10.- (dutch guilders) or the equivalence of USD 10.- (US dollars) in any other currency. Note that if you

order the hardware
you are registered for free.

InfraRexx is no shareware, nor crippled in any way. I'm counting on you as a user to support the
future
of InfraRexx by registering.

1.23 How to pay for what you get...

How to pay for what you get...

Please send the money directly in cash or on an EuroCheck to one of the

authors
, along with the
registration/order form
.

For worldwide users, cash money is the most easy way to get your money across. If you would like to pay using an internation cheque anyway, there is an extra of USD 15 to cover cashing costs.

For European users, both an Eurocheque (in NLG currency!) or cash money are fine.

1.24 InfraRexx Registration/Order Form

InfraRexx Registration/Order Form

Edit -> Use these buttons to fill in, and print out, the InfraRexx
 Print -> Registration and Order Form, as shown below.

READ THIS FIRST: (Read
 Payment
 and
 Ordering the hardware
 for details)

InfraRexx 1.6b Registration and Order Form

User Information

Name:
 ~~~~~  
 Address:  
 ~~~~~  
 ~~~~~  
 Country:  
 ~~~~~  
 Internet/FidoNet E-Mail:
 ~~~~~  
 ( ) Yes, please e-mail new InfraRexx releases as uuencoded mail.  
 ~~~~~ For those that don't have access to FTP or an up-to-date BBS.

InfraRexx Registration

() Yes, I want to register to InfraRexx. I have included the
 ~~~~~ registration fee (NLG 15 or USD 10) with this form.  
 ( ) I get free registration because I order the hardware.

#### Ordering the InfraJoy hardware

( ) Yes, I would like to order the complete InfraJoy v4 hardware.  
 ~~~~~ I've included...

Worldwide users

- () USD 28 in cash money.
- () The equivalent of USD 30 in my local currency, cash money.
- () USD 43 on an international cheque.

European users

() NLG 40 in cash money.
 ~~~~~ ( ) NLG 40 on an Eurocheque.



( ) The equivalent of NLG 43 in my local currency, cash money.

~~~~~What is the Amiga configuration the hardware is used with?

~~~~~  
 ~~~~~-----  
 ~~~~~( ) Optimise receiver for this brand (i.e. Sony, Philips etc.):

~~~~~-----

I have read the copyrights and disclaimer notes and agree,

place, date: _____ sign: _____
 ~~~~~  
 -----

Please allow two to six weeks for delivery of the InfraJoy hardware.

## 1.25 Obtaining Updates...

### Obtaining Updates...

#### Internet users

New InfraRexx releases will be uploaded to Aminet, a worldwide network of FTP sites for freely distributable Amiga software. "ftp.netnet.net" is one.

It can be found in the 'util/rexx' subdirectory of the Aminet tree ('/pub/aminet') under the name 'InfraRexxX\_X.lha' (X\_X representing the release version).

At the time of new releases, an announcement will be posted to the Usenet newsgroup called 'comp.sys.amiga.announce'.

#### World Wide Webbers

For those who surf the super information highway, we are setting up pages for InfraRexx users. Online documentation, latest releases, codesets and so will be accessible via those pages. Like most WWW pages, these are still under construction :-). See  
     authors  
     for the URLs.

#### Unregistered users

You can check your most favorite public domain source, such as your local bulletin board, CD-ROM collections or magazine disks. Should register also.

#### Registered snailmail users

Registered users will receive one update via snailmail, if they can't reach new versions through computer networks or local BBS systems. Snailmail

users will receive one update by means of a mailed 3.5" DD floppy (FFS).

Fred Fish Submissions

Major releases will be submitted to the makers of the Fred Fish series.

## 1.26 Reaching the authors...

Reaching the authors...

If you have any bugs, ideas or if you want to  
register  
or

order the hardware  
, write to one (or both) of the following addresses:

Leon Woestenberg  
Langenhof 62  
NL-5071 TP Udenhout  
The Netherlands

Internet Email: [leon@stack.unc.tue.nl](mailto:leon@stack.unc.tue.nl)  
World Wide Web: <http://www.stack.unc.tue.nl/~leon/>

Jeroen Steenblik  
Retiestraat 13  
NL-5628 PC Eindhoven  
The Netherlands

Internet Email: [jeroens@stack.unc.tue.nl](mailto:jeroens@stack.unc.tue.nl)  
World Wide Web: <http://www.stack.unc.tue.nl/~jeroens/>

---