

IR protocol analyzer – user's manual

Overview

IR protocol analyzer is a universal application for automatic decoding dozens types of infrared remote control protocol packets. The application uses microphone input of a soundcard to capture infrared signal from a remote control. As a consequence, the hardware receiver is minimalistic and easy to build; just plug a phototransistor to input of your soundcard, that's all hardware you need.

Application processes IR signal from a remote control and compares it with its own database of known protocols. When a match is found, packet is decoded and its characteristic is displayed to user (including protocol name, description, decoded data and graph with timing).

Protocol definitions are stored in separate XML file. This XML file can be easily modified and new protocol definitions can be added by user. When an unknown packet is captured, user may still display its graph with timing details. Then, on the basis of the graph and timing, he may create new protocol description.

All captured data can be saved to a file and restored later. Moreover, packet graph can be exported to a bitmap image.

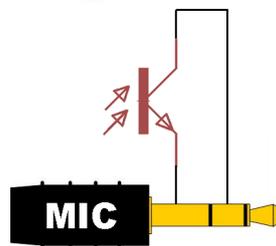
Getting started

You may want test the application without any hardware receiver first. For such case I made some testing data for you. You can find them in 'samples' directory. To open sample data, launch IR protocol analyzer and then press Ctrl+L (or choose **File** -> **L**oad samples). File open dialog will appear; select the 'samples' directory, choose one of *.map files and click 'Open'.

Now you can see former captured packets in the Samples list. Click a sample to view its packet details and graph. That's enough for now; refer to the *User interface* chapter for more information.

Building a simple IR hardware receiver

OK, you have just tested the application with some sample data and now you are certainly agog for analyzing your own remote controls. :) Let's build a receiver to utilize all features of IR protocol analyzer! Construction of a receiver is the simplest as it can be. All you need is a 3.5mm jack connector, a phototransistor (any type, it doesn't matter as long it's sensitive to infrared light – sensitivity near 940nm is ideal), a soldering gun and a piece of wire. Connect components like on the picture below:



The receiver is passive – that means there is basically no chance to hurt your soundcard. However, be aware that you use the receiver ON YOUR OWN RISK. I'm not responsible for any damages done to your computer.

Capturing samples from a remote control

Plug your brand-new receiver into a microphone input of your soundcard. Launch IR protocol analyzer and click the 'Recording' button. IR protocol analyzer starts sampling sound data from microphone (but because there is the IR receiver instead of a microphone, it actually captures infrared signal). While signal recording is active, put the phototransistor directly in front of the transmitting IR LED of your remote control and press a button on the remote. Remote control will send a packet and new samples will appear in the 'Samples' list. Finally, click the 'Recording' button again to stop capturing data.

User interface

The screenshot shows the IR protocol analyzer interface. The main window is titled "IR protocol analyzer" and contains a "Graph" area displaying a digital waveform. The waveform is labeled with "startbit", "command", and "address". Below the graph, there are controls for "zoom", a "Recording" button (which is currently pressed), and a "Clear samples" button. The bottom section of the interface is divided into two parts: "Packet details" and "Samples".

Packet details:

Sony SIRC
<http://www.sbprojects.com/knowledge/ir/sirc.htm>
 The SIRC protocol uses a pulse width encoding of the bits. The pulse representing a logical "1" is a 1.2ms long burst of the 40kHz carrier, while the burst width for a logical "0" is 0.6ms long. All bursts are separated by a 0.6ms long space interval. The recommended carrier duty-cycle is 1/4 or 1/3.
 With this protocol the LSB is transmitted first. The start burst is always 2.4ms wide, followed by a standard space of 0.6ms. Apart

data name	dec	hex	bin
command	101	0x65	1100101
address	1	0x1	1

Samples:

Time	Protocol	Data	Length	Gap
15:06:52	Sony SIRC	command=101; address=1;	20ms	25,3ms
15:06:52	Sony SIRC	command=101; address=1;	20ms	25,3ms
15:08:55	NEC	address=18243; command=201; lcommand=54;	67,1ms	13s
15:08:55	NEC (repeat code)		11,9ms	41ms
15:08:55	NEC (repeat code)		11,8ms	97,5ms
15:08:55	NEC (repeat code)		11,9ms	97,5ms
15:09:26	NEC	address=65025; command=9; lcommand=246;	68,2ms	29,3s
15:10:37	Panasonic	byte0=2; byte1=32; byte2=176; byte3=0; byte4=17; byte5=161;	56,2ms	18,4s
15:10:37	Panasonic	byte0=2; byte1=32; byte2=176; byte3=0; byte4=17; byte5=161;	56,1ms	74,7ms
15:10:37	Panasonic	byte0=2; byte1=32; byte2=176; byte3=0; byte4=17; byte5=161;	56,1ms	74,7ms
15:10:44	Panasonic	byte0=2; byte1=32; byte2=160; byte3=0; byte4=201; byte5=105;	57,7ms	6,1s
15:10:44	Panasonic	byte0=2; byte1=32; byte2=160; byte3=0; byte4=201; byte5=105;	57,7ms	74,1ms
15:10:44	Panasonic	byte0=2; byte1=32; byte2=160; byte3=0; byte4=201; byte5=105;	57,7ms	74,1ms
15:10:59	RC-5 extended	commandBit6=0; togglebit=1; address=8; command=23;	24,1ms	15,1s
15:10:59	RC-5 extended	commandBit6=0; togglebit=1; address=8; command=23;	24,1ms	89,9ms
15:10:59	RC-5 extended	commandBit6=0; togglebit=1; address=8; command=23;	24ms	89,9ms

Recording stopped

The 'Recording' button

This two-state button starts or stops recording. On startup, recording is stopped and button is up. Clicking this button will start capturing data from your soundcard. While application is capturing data, 'Recording' button is in pressed state. To stop recording, click the button again. The button returns to up state and recording is stopped.

Samples

Samples list contain captured or loaded data. You can see a number of basic information about each sample. Those are:

- Capture time
- Protocol name
- Packet data
- Packet length – time duration from start of first mark to end of last mark (burst)
- Gap – time interval between end of previous sample and start of this sample

Click a sample to see its details and graph.

Graph

The graph shows captured raw signal from your soundcard. Each pulse is marked with its timing, time intervals are in μs . Furthermore, if the sample was successfully decoded, the packet structure is displayed and packet segments (delimiters and bit groups) are tagged with their names. Use the scale bar to zoom in and out.

Packet details

Detailed description of selected packet is displayed, including information about the protocol and list of decoded data (in decimal, hexadecimal and binary notation).

Save and Load samples

In the 'File' menu you may 'Save' (Ctrl+S) or 'Load' (Ctrl+L) captured samples. Saving samples will store all raw data that you have recorded (i.e. all samples in the 'Samples' list) to a file. Actually two files are created; *.map and *.dat file. The *.map file contains necessary sample descriptions (*capture time, sample length and gap*). The *.dat file contains chunks of data recorded from your soundcard. When opening sample files, both *.map and *.dat files must be present in the same directory.

IR protocol analyzer doesn't save any data that can be reconstructed from the raw data. That means that anytime a sample file is opened, raw data are scanned and decoded again, using a protocol definitions file loaded on application startup. By the way, that's great opportunity to test how changes in the 'protocols.xml' file affect protocol recognition. Feel free to try that.

Export image

Application allows you to export currently displayed graph to a bitmap image (*.bmp). Open the 'File' menu and choose 'Export graph' (Ctrl+E).

Protocol definition file

On startup, IR protocol analyzer loads file 'protocols.xml' located in the same directory as the executable file. This XML file contains protocol descriptions. Its structure was designed with a respect to user, who may want to extend the protocol description database by himself. If you are interested in adding new protocol definitions, see 'protocols.xml' for more details...

Troubleshooting

Problem: 'When I start recording, regardless whether the receiver is connected or not, samples list fills with many samples of unknown protocol. Their graphs look like an audio wave.'

IR protocol analyzer probably samples sound from your music player or other application.

Solution: Mute all sound sources.