

Glossary and terms

Table of contents

1 Glossary.....	2
2 Appendix. Semantics of the Pronto CCF format.....	9

Date	Description
2013-12-01	Initial version.
2013-12-20	More stuff added.
2014-02-02	Even more stuff added.

Table 1: Revision history

1 Glossary

Here we explain and define some of the used terms. In most cases (but not all!), this correspond to established usage in the Internet, e.g. in the JP1 forum. It should also be pointed out that in some cases, in particular when comparing programs by others to my own, the assessment should be considered as subjective.

AMX Beacon

A daemon program implemented in some networked components. It is used for periodically announce their existence, and some of their properties.

Analyzer

The part of the [ExchangeIR](#) library that analyzes a [raw IR Signal](#). If successful, it returns an [IRP form](#), together with parameter values. The returned IRP form may or may not correspond to a known protocol.

Arduino

A single-board microcontroller, based upon a 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. [Official site](#). Can be used for implementing sending and/or capturing of infrared signals with additional hardware attached to its IO-pins.

Capturing (sometimes "learning")

The process of recording an unknown physical [IR sequence](#) as a sequence of on- and off-durations, preferably together with a measure of its [modulation frequency](#). Cf. [receiving IR signals](#).

CCF (text signal format)

IR signal format. Not to be confused with the [ccf file format](#) of the Pronto Classic remotes! Often called "hex", "Pronto", or "Pronto hex". Consist of a sequence of four-digit hexadecimal numbers. For the interpretation, see [the Appendix](#). It is a very popular format, e.g. for textual the exchange in Internet forums.

ccf file format

Not to be confused with the [CCF text signal format](#)! Configuration file format for the Pronto Classic. File extension is `ccf`. Can be edited by the [ProntoEdit](#) program. Has been completely reverse engineered, and the open-source program [Tonto](#) is able both to interactively edit them, as well as non-interactively through an API. IrScrutinizer can import and export ccf files, using the said API.

Cleansed signal

Given a captured [IR sequence](#), using the [repeat finder](#), all occurrences of the repeat sequence but one are removed, and numerically "close" duration values are lumped into one single value.

Command

Here, an [IR signal](#) with a name, like "Play".

Consumer IR (CIR)

Consumer IR deals with IR control of various devices. [Wikipedia article](#). Not to be confused with [IRDA](#). Typically uses wave lengths of 930-950nm.

CSV (comma separated values)

Primitive data base format, one record consisting of one line, the entries separated from one another by a comma (,) (or sometimes another character). One possible file extension `csv`. Can be read directly by spreadsheet programs.

Decode (noun)

Given an [IrSequence](#) or an [IrSignal](#), a *decode* is a [protocol](#), together with parameter values, that are compatible with the given IrSequence/IrSignal, i.e. could have generated the original signal. Note that the determination is governed by numerical uncertainties, so that small deviations from the perfect signal are accepted. Furthermore, one signal/sequence may have none, one, or more valid decodes.

DecodeIR

Library for the [decoding](#) of IrSequences. Originally written by John Fine, extended by others; used by many widely spread programs as a shared library, often using [JNI](#). The current version is 2.44. [Binaries for Windows, Linux, and Mac](#), [source code](#).

Demodulating IR Receiver

An integrated circuit that receives a [modulated IR signal](#) and recovers the original signal with the modulation removed. The modulation frequency of the signal must "match" the frequency of the demodulator. [Data sheet](#) for a typical product. Not suited for [capturing](#) of unknown signals, since it removes the modulation frequency without identifying it.

Device Number

See [protocol parameters](#). Denoted by D in [IRP protocols](#).

Device Type

Class of components, like TV, VCR, Satellite receiver, etc.

Duty Cycle

The percentage of the time the the modulation pulse is on. Typically slightly less than 50%.

Ending sequence

See [IrSignal](#).

ExchangeIR

Library for IR signal analysis and exchange by Graham Dixon. Licensed under the [GPL3 license](#). For the present project, the interesting parts are the [Analyzer](#), the [repeat finder](#), and functions for the [UEI learned format](#). These parts has been translated to Java by myself: [API documentation](#), [source](#).

Flash

Period of time when the IR light is "on", or flashed with the selected [modulation frequency](#). See [IrSequence](#).

Function Number

See [protocol parameters](#). In IRP protocols, denoted by F. A synonym is [OBC](#).

Gap (or "Pause")

Period of time when the IR light is off. See [IrSequence](#).

Generating, sometimes called rendering

The process of evaluating an [IrProtocol](#) for a particular parameter value, rendering an [IrSignal](#). Commonly used rendering programs/engines are the older [MakeHex](#), and the more modern and capable [IrpMaster](#).

Girr (Generic IR Remote)

A general [XML](#)-based exchange formats for IR Signals. Really a container format that can contain any of the [CCF](#), [raw format](#), [protocol/parameter](#) format, as well as other text formats like [Global Caché sendir](#) and [UEI learned format](#). For a full description, see [the full documentation](#).

Global Caché

<http://www.globalcache.com/>

GPL3 license

The current version of the [GNU General Public License](#). Used by my software projects (with some exceptions), and many so-called open-source software projects. The basic idea is the licensee is allowed to use, enhance etc. the software (also in a commercial product and context), but is not allowed to turn it, or a derived product, into non-free software.

IR (Infrared light)

According to [Wikipedia](#), infrared light are light (electromagnetic radiation) of wavelength between 700nm and 1mm. For control of consumer electronics (CIR), according to [Wikipedia](#), wavelengths around 870 nm and 930-950 nm (latter preferred), in comparison to IrDA (850-900nm) are used. Almost always generated by an [IR LED](#).

IrDA

[IrDA](#) is a method for data exchange between PCs and portable devices. It is no longer to be considered as state-of-the-art, and has been almost completely replaced by Bluetooth and WiFi. Many devices with IrDA hardware is still around, but it is, with few exceptions, unsuitable for [consumer IR](#).

IR LED (light emitting diode)

Semiconductor component capable of sending light with the desired IR wavelength. A typical representative is the [Osram SFH 4512](#).

(IR) Protocol

An algorithm for turning a number of parameters into an [IR signal](#). It defines the necessary parameters and their allowed values. In almost all protocols, the most frequently changing parameter is called "F" (function number). Almost all protocols have a "device number" called "D". Many protocols have a "sub-device" number,

called "S". A few protocols have a [toggle](#) parameter, in general called "T", and being [persistent](#). A protocol may also have other parameters, with "arbitrary" names.

IrMaster

A program for generating, analyzing, importing, and exporting of infrared signals. Now discontinued, replaced by [IrScrutinizer](#).

IrpMaster

A program and API library for [rendering IRP protocols](#) version 2. See [its documentation](#). Comes with a powerful (but slightly hard to use) command line interface. For GUI usage, see [IrMaster](#) and [IrScrutinizer](#). Note that the word "IrpMaster" sometimes refers to the command line program, sometimes to the rendering engine contained in IrMaster and IrScrutinizer.

IRP Notation

Compact, slightly cryptical, notation for defining an [IrProtocol](#). [Specification](#).

IrScope

Program that accompanies the [IrWidget](#), also by Kevin Timmerman. Originally a support to the IrWidget, was further developed (in particular through Graham Dixon) to a fairly general and capable IR analyzing program, supporting also [DecodeIR](#) and [ExchangeIR](#). The program was a major inspiration source for [IrScrutinizer](#).

IrScrutinizer

IrScrutinizer is a powerful program for capturing, generating, analyzing, importing, and exporting of infrared signals.

IR Sequence

Sequence of time durations, in general in expressed microseconds, together with a [modulation frequency](#). The even numbered entries normally denote times when the IR light is on (modulated), called ["flashes"](#), the other denote off-periods ["gaps"](#).

IR Signal

Consists of three [IR sequences](#), called

1. *start sequence* (or "intro", or "beginning sequence"), sent exactly once at the beginning of the transmission of the IR signal,
2. *repeat sequence*, sent "while the button is held down", i.e. zero or more times during the transmission of the IR signal (although some protocols may require at least one copy to be transmitted),
3. *ending sequence*, sent exactly once at the end of the transmission of the IR signal, "when the button has been released". Only present in a few protocols.

Any of these can be empty, but not both the intro and the repeat.

IrToy

An interesting "open hardware" project by Dangerous Prototypes, see the [product page](#). Consists of a microprocessor PIC18F2550, a [demodulating IR-receiver](#), a [non-demodulating IR-receiver](#), an [IR-LED](#), and a USB-connector. Thus, is usable both for learning, including frequency measurements, receiving demodulated R-signals, and sending IR signals. Supported by IrScrutinizer.

IrTrans

A [series of IR products](#) from the firm with the same name. IrScrutinizer and IrMaster supports the Ethernet models (preferably with the "IR data base"), for sending only.

IrWidget

A very interesting "open hardware" project by Kevin Timmerman. [Project page](#). That page presents many different versions, but the most spread version (until recently made [commercially available](#) by Tommy Tyler) consists of a micro processor PIC12F629, a non-demodulating sensor, and a USB serial FTDI interface. Supported by Kevin's [IrScope](#), as well as IrScrutinizer.

Java

Java Native Interface (JNI)

A technique for having a Java program calling a native shared library (DLL in Windows, "Shared object" (.so) in other operating systems). See the [Wikipedia article](#).

JP1

JP1 Remote

JSON

A standard for using human readable text to transfer structured data, as an alternative to XML. See the [Wikipedia article](#).

LIRC

An open source project for sending and receiving IR signals from Linux. [Official web site](#). First release in May 1996, current version is 0.9.0, released in March 2011. IrScrutinizer supports sending through a LIRC server, patched with the [CCF patch](#).

MakeHex

A predecessor to IrpMaster. Adheres to an earlier version ("Version 1") of the [IRP Notation](#). For the original C++ program by John Fine, neither a GUI nor a command line interface are present; the parameters are given to the program by editing the data base files. A Java translation (by myself) exists, which has a command line interface, [available here](#).

Mode2 (LIRC) format/program

Modulation frequency

During the "on" periods, the IR light is not just constantly on, but "flashed" on and off at a frequency called the modulation frequency, typically between 36kHz to 40kHz, in some cases higher (up to 56kHz), or much higher (455kHz, Bang & Olufsen equipment). This reduces noise sensitivity and power consumption, and also allows higher currents through the IR LED (that thus does not have to be able to survive the high current continuously). Also see [Duty cycle](#).

(Non-demodulating) IR receiver

IR receiver that outputs the received IR signal essentially as received, i.e. without removing a [modulation](#).

Original Button Code

Synonym for [function number](#).

Parametric IR signal

An Ir Signal given as a protocol and a parameter assignment to its parameters. Of course, a [renderer](#) may compute the numerical IR Sequences, but these are *considered* secondary, it is *defined* by its protocol and parameters values.

PCF

IR signal format, not to be confused with the [pcf file format](#) of the Pronto NG remotes! This is a proprietary and encrypted form of IR signals. As far as I am aware, it is presently not known how to decode this representation.

pcf file format

Like the [xcf format](#), this is a [ZIP](#) file containing an [XML](#) file with the real payload, and a number of icon files. Unfortunately, the enclosed IR signals are in the [PCF](#) format, thus possible to decrypt only by the ProntoEditNG program.

Persistent variable

A *persistent variable* in an [IR protocol](#) may, but need not, be given a value before [generating](#). If not, it retains its value from previous invocation, or, for the first invocation, has a default value.

Pronto Classic

Family of advanced touch-screen remote controls. Manufactured by Philips 19xx-19yy. TODO. Consists of the models TS1000, TSU2000, TSU6000, RC5000, RC5000i, RC5200, RC9200, RU890, RU940, RU970, USR5, RAV2K, RAV2KZ1. Configurable/programmable by a GUI program "ProntoEdit", as well as the open-source program [Tonto](#).

ProntoEdit

Windows program for programming the Pronto remotes. Exists in different versions for different Pronto series. From its owner Philips now discontinued, but available for download at [RemoteCentral](#).

Pronto frequency code

The second number in the CCF representation. For f in Hertz, this is the four-digit hexadecimal number given as $1000000/(f*0.241246)$. It can be conveniently computed by the Time/Frequency Calculator in IrScrutinizer, available under the Tools menu.

Pronto NG (New Generation)

Later generation of Pronto touch screen programmable remotes. Uses the [pcf format](#) as their configurations. Can be read by ProntoEditNG.

Pronto Professional (*.xcf configuration files)

Later generation of Pronto touch screen programmable remotes. Uses the [xcf format](#) as their configurations. Consists of the models TSU9800, TSU9600, TSW9500, TSU9400, TSU9300, TSU9200, TSU9500 (Philips) and RC9001 (Marantz).

Protocol Parameters

See [IR Protocol](#).

protocols.ini

properties (of an interactive program)

The part of the program's state saved between sessions for each user; saved to disk.

Raw IR sequence/signal

A raw Ir Sequence is a sequence of (in general) measured on-off durations. It may or may not have one or many [decodes](#), but these are *considered* to be secondary; its is *defined* by its numeric time durations. Often written with signs: a "+" indicates a [flash](#), a "-" indicates a [gap](#).

Receiving IR signals (deployment)

The use case of receiveing an a priori partially known (typically through its [protocol](#), in particular, the [modulation frequency](#)) signal, identifying it completely (typically its parameters [protocol parameters](#)), and possibly initiating an action. Cf. the other use case [capturing](#).

Repeat finder

Class in [ExchangeIr](#) (function in the original version) that from an [IR sequence](#) numerically tries to identify the intro-, repeat-, and ending sequence that has generated the original sequence.

Repeat sequence

See [IrSignal](#).

Remote

A collection of commands with unique names.

RemoteMaster

rmdu file

scrutinize

"To examine in detail with careful or critical attention."

sendir (Global Caché) format

Start sequence

See [IrSignal](#).

Sub device Number

See [protocol parameters](#). In IRP protocols, denoted by S.

TVS (tab separated values)

Like [CSV](#) but using a tab character (ASCII character 9). File extension `.tsv`, or other.

Toggle

[Persistent variable](#) in an [IrProtocol](#), in general alternating between 0 and 1, between different invocations. I.e., if the first invocation has the toggle value 0, all even invocations will have the value 1 of the toggle, all even the value 0, independent of the number of repeat sequences. Also see [protocol parameters](#).

Tonto

An open source re-implementation of [ProntoEdit](#) for the [Pronto Classic](#), as well as an Java API library for reading and manipulating [CCF files](#). Author is Stuart Allen. [Home site](#).

UEI learned format

wave file format

An [IR sequence](#) rendered with halfed [modulation frequency](#), as a sequences of equidistant samples (in general with sample frequency 44.1kHz or 48kHz) considered as an audio signal. It is supposed to be "playbacked" through an audio system

connected to a pair of IR LEDs connected in anti-parallel, which will again double the carrier frequency. IrScrutinizer supports both the generation of wave files, as well as its import and analysis.

WinLIRC

According to its [its web site](#), it "...is the Windows equivalent of [LIRC](#)". It is not a port of LIRC, nor does it share any code with LIRC.

xcf configuration file

Configuration file format for the [Pronto Professional](#) line of remotes. Consists of a [ZIP](#) file containing one configuration file in XML-format, as well as a number of supplementary icon images. The XML file is very easy to understand (for programmers!), and can contain IR signals in different formats, like [CCF format](#) (usable!) and [PCF format](#) (encrypted, thus not usable).

XML

XML Schema

XML Schema (also called XSD, for "Xml Schema Language") is an XML language for describing the syntax of XML documents. See the [Wikipedia article](#).

XSLT (Extensible Stylesheet Language Transformations)

XSLT is an XML language for transforming XML documents into other XML documents, HTML-pages, or plain text. See the [Wikipedia article](#). The programs here use only XSLT version 1.0.

2 Appendix. Semantics of the Pronto CCF format.

An IR signal in Pronto CCF form consists of a number of 4-digit hexadecimal numbers. For example:

```
0000 006C 0022 0002 015B 00AD 0016 0041 0016 0016 0016 0016
0016 0016 0016 0016 0016 0016 0016 0016 0016 0016 0016 0016
0016 0041 0016 0016 0016 0016 0016 0016 0016 0016 0016 0016
0016 0016 0016 0041 0016 0041 0016 0016 0016 0016 0016 0016
0016 0016 0016 0016 0016 0016 0016 0016 0016 0016 0016 0041
0016 0041 0016 0041 0016 0041 0016 0041 0016 0041 0016 06FB
015B 0057 0016 0E6C
```

The first number, here 0000, denotes the type of the signal. 0000 denotes a [raw IR signal](#) with [modulation](#), while 0100 denotes a non-modulated raw IR signal. There are also a small number of other allowed values, denoting signals in [protocol/parameter form](#), notably 5000 for RC5-protocols, 6000 for RC6-protocols, and 900A for NEC1-protocols.

The second number, here 006C, denotes a frequency code. For the frequency f in Hertz, this is the number $1000000/(f*0.241246)$ expressed as a four-digit hexadecimal number. In the example, 006C corresponds to $1000000/(0x006c * 0.241246) = 38381$ Hertz. (It can be conveniently computed by the Time/Frequency Calculator in [IrScrutinizer](#), available under the Tools menu.)

The third and the forth number denote the number of *pairs* (= twice the number of durations) in the [start-](#) and the [repeat sequence](#) respectively. In the example, there are $0x0022 = 34$ starting pairs, and 2 repeat pairs.

Next the start- and the repeat-sequences follow; their length being given by the third and the forth number, as per above. The numbers therein are all time durations, the ones with odd numbers on-periods, the other ones off-periods. These are all expressed as multiples of the period time; the inverse value of the frequency given through the second number. For this reason, "frequency" must be a sensible number also for the non-modulated case, denoted by the first number being 0100. In the example, the fifth number 0x015B denotes an on-period of $0x015B * periodtime = 347/f = 347/38381 = 0.009041$ seconds.

In particular, all sequences start with an on-period and end with an off-period.

In the Pronto representation, there is no way to express an ending sequence.