Glossary and terms

Table of contents

Date	Description
2013-12-01	Initial version.
2013-12-20	More stuff added.
2014-02-02	Even more stuff added.
2014-09-18	Yet more stuff added.
2016-01-08	Substantial improvements and extensions.
2016-04-29	Some more entries.
2017-03-19	Some more entries.
2019-08-03	Minor updates, triggered by IrpTransmogrifier.
2019-12-25	Yet another batch of minor updates.
2022-05-13	Some updates in the context of IrScrutinizer 2.4.0.

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.

Substantial program/program packages are written capitalized, (not uppercase), in some case in CamelCase, and in Roman typeface, like a proper noun. "Small" programs are written as `code`. For example, we write "Lirc", not "LIRC", or `lirc`; and `irsend`.

AMX Beacon
A daemon program (or protocol) implemented in some networked components. It is used for periodically announce their existence, and some of their properties.

AnalysIR
Commercial infrared analyzer and decoder program for Windows. By its makers characterized as "... the leading tool available for analysing, decoding and reverse engineering infrared remote control protocols". Last release appears to be from July 2016. Web site.

Analyzer
We use the word "analyzer" for a program that takes one or more IR signals/ sequences as input, and from them computes one (or several alternative) IRP-Forms (with parameter values) that is compatible with the data, i.e. that could have generated the given data. IrpTransmogrifier contains an implementation, used in the current version of IrScrutinizer.

ANTLR

"A powerful parser generator for reading, processing, executing, or translating structured text or binary files. ... From a grammar, ANTLR generates a parser that can build and walk parse trees." Website, see also documentation site.

Arduino

A single-board micro-controller, based upon a 8-bit Atmel AVR micro-controller, 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, see for example this project. The main components of the Arduino eco systems are:

- The Arduino IDE (Integrated Development Environment) (which is an open source Java IDE program containing an editor and a code preprocessor. It invokes the tool chain),
- A set of includes and libraries for the processor, mainly GNU software,
- A tool chain (compiler, linker, "binutils") for the processor, mainly GNU software (installed with the IDE),
- A bootloader,
- A number of hardware boards based on "open hardware".

Arduino library manager

One part of the Arduino ecosystem. Consists of a number of user-contributed, not officially supported libraries for Arduino. From the Arduino IDE, selecting "Tools" -> "Manage libraries" a GUI frontend to these libraries is opened, allowing to install and update any of these libraries.

Bracketed raw format

Simple text format for formatting of a raw IR signal, introduced by IrpTransmogrifier. Intro-, repeat, and (optionally) ending sequences are enclosed by brackets ("[]", sometimes called "square brackets" or "square parenthesis"). Signs ("+" for flashes, "-" for gaps), as well as interleaved commands or semicolons, are accepted, and ignored (even if they are not in correct order). Optionally, the beginning sequence can be preceeded by `Freq=`*frequency_in_Hz*`Hz` giving the modulation frequency. If not explicitly given, a default value (dependent on the reading program) is assumed. Example (an RC5-signal): `Freq=36000Hz[][+889,-889,+1778,-889, +889,-889,+889,-889,+889,-889,+889,-889,+889,-889, +889,-889,+889,-889,+889,-889,+889,-889,+889,-889, +889,-90886][]`.

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. See (Non-demodulating) IR receiver. Cf. receiving IR signals.

CCF (text signal format)

Sometimes used as a synonym for the Pronto Hex format. Not to be confused with the ccf file format of the Pronto Classic remotes!

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 dirty signal, numerically "close" duration values are lumped into one single value. Often combined with a repeat finder.

CML

Proprietary binary format by RTI for their IR database files. Has been reverse engineered; IrScrutinizer can import it. One well known CML file is the Mega List Database, which is a huge database of IR codes maintaned by Glackowitz from the Remote Central Forums.

Command

Here, an IR signal with a name, like "Play".

Consumer IR (CIR)

Consumer IR deals with IR control of various devices, often audio or video. Wikipedia article. Not to be confused with IRDA. Typically uses wave lengths of 940–950nm.

CVS (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 2.45. License: public domain.Binaries for Windows, Linux, and Mac, source code at SourceForge. Arduino version (also available in the Arduino library manager). Now superseded by IrpTransmogrifier.

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. Receiver chips are typically marked TSMP*XXYY*, where *XX* (two or three digits) denotes a vendor specific type, and *YY* the modulation frequency in kHz. (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.

`/dev/lirc`

Using Linux, a device node for a connected, supported IR device. Despite the name, it is a part of the Linux kernel, not Lirc, thus available also if Lirc is not installed. Supported by IrScrutinizer, using the DevSlashLirc library.

Device Type

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

DevSlashLirc

Library for object oriented access to `/dev/lirc` -hardware using Java or C++ (currently). Source repository.

Dirty Signal/Sequence

A physically measured signal or sequence containing random measurement errors, "dirt". Such a signal/sequence contains several numbers that are close, but not equal.

Duration

A duration is either a gap or a flash.

Duty Cycle

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

Ending sequence

See IrSignal.

Eventghost

"EventGhost is an advanced, easy to use and extensible automation tool for MS Windows. It can use different input devices like infrared or wireless remote controls to trigger macros, that on their part control a computer and its attached hardware." Licensed under GPL2. Home page.

ExchangeIR

Library for IR signal analysis and exchange by Graham Dixon. Licensed under the GPL3 license. Some interesting parts are an Analyzer, a repeat finder, and functions for the UEI learned format. These parts has been translated to Java by myself: source.

Executor

An embedded program fragment for the rendering and transmission of one or several protocols. One executor can manage several protocols; also, for one protocol there may be several alternative executors. An executer has its own parametrization, more-or-less similar to the parametrization of the protocol. Used in JP1 Remotes and RemoteMaster.

Flash (or "mark", "on-period")

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. In the JP1 community, often the synonym OBC is used.

Gap (or "space", "off-period", "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, IrpMaster (included in versions 1 of IrScrutinizer, and the more modern and capable IrpTransmogrifier (included in the current versions of IrScrutinizer).

Girr (Generic IR Remote)

A general XML-based exchange formats for IR Signals. Really a container format that can contain any of the Pronto Hex, raw format, protocol/parameter format, as well as other text formats like Global Caché sendir. For a full description, see the full documentation.

Global Caché

Manufacturer of IR sending hardware, in some cases with learning and/or receiving possibilities. Web site. Supported in IrScrutinizer.

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 commerical product and context), but is not allowed to turn it, or a derived product, into non-free software.

Iguana USB IR transceiver

A family of USB transmitter/receivers for IR signals. Supported by IrScrutinizer on Lirc (only) by using the `/dev/lirc` interface.

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-or-the-art, and has been almost completely replaced by Bluetooth and WiFi. Many devices with IrDA hardware are still around, but they are, possibly with a 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 4546.

(IR) Protocol

An algorithm for turning a number of parameter values into an IR signal. It defines the necessary parameters and their allowed values. By convention, 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 version 1. Now discontinued and obsolete; replaced by IrpTransmogrifier.

IRP Notation

Compact, slightly cryptical, notation for defining an IrProtocol. Specification.

IrpTransmogrifier

Library and command line program for the IRP notation protocols, with rendering, code generation, and recognition applications. Reference manual. Effectively replaces IrpMaster, DecodeIR", and ExchangeIR.

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. Reference manual.

irsend

Program that implements a simple client for the Lircd server daemon, for sending commands to a lircd server. Contained in the Lirc package. For a complete list or commands implemented, see its man page. IrScrutinizer contains a GUI version with (approximately) the same functionality, under `Tools -> Named Command Sender`. For a Python alternative, see Lirconian. For a Java alternative, see JavaLircClient.

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" or "marks", the other denote off-periods "gaps" or "spaces". They always start with a flash, and end with a gap. Sometimes the flashes are written with a leading "+"-, and the gaps with a leading "-"-sign. This has only a decorative purpose; all involved numbers are still positive.

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. A non-empty ending sequence is only meaningful with a non-empty repeat.

IrToy

An "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 up to and including 2.3.1, but not 2.4.0.

IrTrans

A series of IR products from the firm with the same name. IrScrutinizer and IrMaster support the Ethernet models (preferably with the "IR data base"), for sending only. Not supported in the current version 2.4.0.

IrWidget

An "open hardware" project by Kevin Timmerman. Project page. That page presents many different versions, but the most spread version (commercially available by Tommy Tyler; unknown if still active) consists of a micro processor PIC12F629, a non-demodulating sensor (QSE15x), and a USB serial FTDI interface. Supported by Kevin's IrScope, as well as IrScrutinizer.

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

Community for customizing JP1 Remotes. Forum.

JP1 Remote

Customer remotes manufactured by Universal Electronics Inc., manufactured by many different names, like One for all etc. Has been reverse engineered by the JP1 community. Can be programmed through a connector, called (after the PCB label) JP1. The main tool for this is RemoteMaster.

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 0.10.1, released in September 2017. Also packaged in all major Linux distributions. IrScrutinizer supports sending named commands to a Lirc server, and can also import and export files in `lircd.conf` format.

lircd

The main daemon of the Lirc project. Accepts commands on a Unix domain socket, alternatively on a TCP socket, by default 8765. Accepts commands to send IR signals in Remote/Command format only. To send commands to a running lircd, often a program called irsend is used.

lircd.conf

The main Lirc configuration file; a data base of remotes and its contained commands. Typically residing in `/etc/lirc/lircd.conf`. Also used to denote the file format. The files in the Lirc data base are in this format. Although its syntax and semantics is documentated, should not be considered a viable exchange format. Can be imported by IrScrutinizer — since it contains a substantial amount of Lircd, translated to Java.

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

1. A simple test format consisting of interleaved "on- (`pulse`) and off-durations (`space`).
2. A simple program contained in Lirc, printing the mode2 text format of received IR sequences to standard output.
3. In Lirc, notation for drivers/plugins capable of generating/evaluating data containing timing information.

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. Preferred sensor component is a chip with pre-amplifier like TSMP58000, at least for "moderate" modulation frequencies (< 60kHz).

Original Button Code

Acronym for "Original Button Code", a synonym for function number used in the JP1 community.

Parametric IR signal

An Ir Signal given as a protocol and an assignment to its parameters. The opposite is a raw signal. Of course, a renderer may compute the raw, 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 Hex

IR signal format. Often called "CCF", "hex", or "Pronto". 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.

Pronto Classic

Legendary Family of advanced touch-screen remote controls. Manufactured by Philips between 1998 and 2004. 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 Pronto hex 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 its Tools menu.

Pronto NG (New Generaton)

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). Discontinued in 2010.

Protocol Parameters

See IR Protocol.

protocols.ini

Data base file for RemoteMaster. Despite the name, it does not descibe protocols in our sense, but rather executors, their properties and parameterization.

properties (of an interactive program)

The part of the program's state saved between sessions for each user; saved to disk, or, sometimes with Windows, in the Windows registry.

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. A text format expressing the durations in microseconds is called "raw format", even if the signal as such is parametric.

Receiving IR signals (deployment)

The use case of receiving 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

A repeat finder is a program that, when fed with an IrSequence, tries to identify a repeating subsequence in it, and returns an IrSignal containing intro-, repeat-, and ending sequence, compatible with the given input data. The library ExchangeIr contains a repeat finder, which was used in IrScrutinizer versions up to and including version 1.2. IrpTransmogrifier also contains a repeat finder, which is used in the current version of IrScrutinizer.

Repeat sequence

See IrSignal.

Remote

1. A collection of commands with unique names.
2. A piece of hardware with buttons etc.

Remote/Command format

Given appropriate data base entries, the name of a remote (or device) together with the name of a command, can identify a command uniquely.

RemoteMaster

Powerful open-source tool for the programming of JP1 remotes. see Manual. Download current released version. Sources. Recent version can import and export Girr files.

RMDU file

Configuration file for "device update" for RemoteMaster, describing the configuration of a (universal) remote for one particular device. Contains parametrized commands, unfortunately not in the protocol/parameter form, but parametrized by an executor and *its* parameters.

RMIR file

Configuration file for a complete remote configuration for RemoteMaster, describing the configuration of a (universal) remote, in general containing one or several device updates. and *its* parameters.

scrutinize (verb)

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

sendir (Global Caché) format

Text format used by Global Caché devices for expressing an IR signal, together with some additional information (number of sends, transmitter). IrScrutinizer can translate to and from this format.

Start sequence

See IrSignal.

StringTemplate

"A Java template engine ... for generating source code, web pages, emails, or any other formatted text output. StringTemplate is particularly good at code generators, multiple site skins, and internationalization / localization." Website, see also Github documentation.

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. Now discontinued. Author is Stuart Allen. Sources at Github. The API library is used in IrScrutinizer to read from, and export to, CCF files.

Transmitter

Some IR senders have more than one sending channel, called transmitter, allowing for example to control different equipment independently, even if they are using the same commands. These are called transmitters. Note that by definition, every IR sender has at least one transmitter, but only in the case of multiple transmitters, a selection is meaningful.

Transmogrify (verb)

"To change or alter greatly and often with grotesque or humorous effect"

UEI learned format

Proprietary internal format for a single IR signal from Universal Electronics Inc. By convention formatted as a sequence of two-digit hexadecimal numbers. Has been reverse engineered in ExchangeIR. Supported in previous versions of IrScrutinizer, but not in the current. (Rationale.)

wave file format

An IR sequence rendered with halfed modulation frequency, as a sequences of equdistant 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. A common mis-conception is that a stereo signal is used for this. 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. Appears currently not to be maintained; last release was 0.9.0i released in May 2014. Statements on Lirc in these pages are not necessarily true for WinLirc.

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

Here, a data base file in certain, marked up, text file format.

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 HEX (CCF) format.

> Note:
>
> There are a few very old guides to the format circulating on the internet. These were written in the previous century, as the subject was not very well understood. Although likely very valuable at the time they were written, however, now they are basicall completely unsuitable. Please do not read, and in particular, do not recommend to others. This appendix contains all needed to know — at least in 2016.

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* (= half 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 as the second number. For this reason, "frequency" must be a non-zero 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 = 9.041 microseconds.

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.