CHIP-8

Package manual
version 0.2.0
August 2016

Dani Rodríguez

This is the user manual for version 0.2.0 of CHIP-8 software package. To check for updates, please visit http://github.com/danirod/chip8.

Table of Contents

1 Introduction

Welcome to *chip8*, my implementation of an emulator compatible with the CHIP-8 virtual machine. This software manual will bring you information about the emulator, how to run programs on the CHIP-8, how to use the emulator and some information for those interested on the insides of the CHIP-8 virtual machine and the format used by the ROMs that can be run from CHIP-8.

1.1 Main features

This implementation for the CHIP-8 virtual machine has, at this moment, the following features:

- Compatible with CHIP-8 and SUPERCHIP specifications.
- Modular, lightweight and fast.
- Multiplatform. Not only runs on GNU/Linux but also can be compiled on Microsoft® Windows® and Apple® MacOS® X.
- Virtually compatible with any machine by using the SDL2 library.

At this stage the emulator has not been finished. There are a few features that have not been implemented yet. There is a particular interest in adding the following features:

- Support for the MEGACHIP specification.
- User defined settings.
- Development tools for writing ROMs.

1.2 Reporting bugs

This tool has not been finished yet, and therefore there might be issues that haven't been reported or addressed yet. If during the execution of a program you find some bug that would like to report for having it fixed, you can report bugs to our issue tracker, at the canonical repository for our project. The URL is `https://www.github.com/danirod/chip8`.

Please, provide as much information as you can, including the operating system you are using, the version of the emulation software and if you can tell us which ROM were you playing, it would be better.

1.3 Some history

CHIP-8 is an interpreted programming language and the specification for a virtual machine. As is, CHIP-8 is not a cmoputer, and there are no computers in the wild using this architecture, because it has limited resources. Instead there are computers with their own hardware, that have a virtual machine that allows the computer to decode and execute CHIP-8 programs.

One of the first computers in add support for CHIP-8 as the COSMAC VIP, sold by RCA in the late 70s. This computer had a processor made too by RCA, with a speed of about 1.77 MHz and 2 kB of RAM memory. Inside the ROM, the operating system had the CHIP-8 interpreter that allowed the system to run games using the CHIP-8 language.

However, there are more machines that have added support for CHIP-8, specially during the late 80s and early 90s, with the expansion of small computers such as graphical calculators. In fact, during this period and thanks to the first Internet communities that came around that time, many hobbyists could extend the original CHIP-8 specification, by making some changes that would make them run better on their machines, and sometimes even improving the original CHIP-8 system, with features such as a bigger screen, more memory and even color.

Thanks to being easy to understand, CHIP-8 is one of the greatest platforms for those people who want to start working in the world of emulators development, learning an easy architecture before starting bigger projects.

2 Emulator usage

2.1 Installing the emulator

The main procedure for installing the emulator is to download the source code distribution, compiling it and installing it. Of course, there are aditional ways for obtaining the emulator, such as getting it from a binary distribution, for example, a *.deb* file for Debian based systems or an *.exe* application for Microsoft® Windows®.

This emulator only depends on SDL2, because it is the multimedia library used for displaying the emulator output, reading keys from the keyboard and making the speaker buzz. After the dependencies are met, this CHIP-8 emulator can be installed by downloading the software release, extracting it somewhere, and then running `./configure && make` to compile it.

Once compiled, it can be installed using `make install`. By default on a standard GNU/Linux system they would be installed into `/usr/local/bin`, `/usr/local/share/doc/chip8`, etc. On Apple® MacOS® X systems, these locations will be used as well. On Microsoft® Windows®, the location might actually change depending on how you build the software.

If you are interested in checking that the software actually works as intended you can use the test suite provided with the source distribution. In order to run the tests you'll need to have *check* installed on your computer. Install the dependencies, and then after running `./configure && make` on the emulator package, run `make check` to run the test suite. After the execution, you will get a report on the test status.

2.2 Running the emulator

After the emulator has been downloaded it can be executed using the command `chip8 file`, where *file* is the path to a binary file that has the encoded data for a program that can run on the CHIP-8 virtual machine.

As an example, to run the ROM `WIPEOFF.BIN` using the emulator, the following command can be used:

```
$ chip8 WIPEOFF.BIN
```

assuming the game is on the current working directory. Some games can make use of the buzzer that the CHIP-8 computer has. The buzzer will sound through the speakers. If you want to avoid this behaviour you can run the game muted using the `--mute` option:

```
$ chip8 --mute WIPEOFF.BIN
```

There are actually two kinds of ROMs: binary ROMs and hexadecimal ROMs. A binary ROM only has binary data and it is the most lightweight and fast way for running a ROM because the data can be placed in memory and executed instantaneously. This data can come from a dump for an actual CHIP-8 game or a game made for the CHIP-8 platform in any other way

However, to make easier to people to create their own ROMs, there is an aditional way for running these ROMs and is using hexadecimal files. These files are plain text files only having the characters `0-9`, `A-F` and `a-f` encoded as either US-ASCII, UTF-8, Windows-1252 or a similar human readable text-enconding and can be modified using any kind of text editor that supports plain text. This format is slower to load since the file has to be converted to binary but it makes playing around faster.

2.3 Using the emulator

Once the emulator opens, it automatically starts executing the ROM contents. It is possible to control the emulator using the emulated keyboard. The CHIP-8 specification sets a 16 keys keyboard using a 4x4 table having the following layout:

```
[1] [2] [3] [C]
[4] [5] [6] [D]
[7] [8] [9] [E]
[A] [0] [B] [F]
```

Using the current settings, that aren't modifiable by the user yet –although that is hoped for future releases, the assigned keys for the keyboard are mapped to the following keys on a traditional keyboard:

```
[1] [2] [3] [4]
[Q] [W] [E] [R]
[A] [S] [D] [F]
[Z] [X] [C] [V]
```

Which means that if you want to send the *5* key to your game you should press the *W* key in the emulator. Pressing *F* will trigger *E* key and similar.

During the execution of a CHIP-8 ROM, some games may play sounds on the buzzer. As explained in Section 2.2 [Running the emulator], page 3, this behaviour can be changed by using the option `--mute` when running the game.

2.4 Stopping the emulation

At this moment there is no way for pausing, resuming, resetting or stopping the emulation in this CHIP-8 emulator. The only way for stopping it is by exiting the emulator, something that you can do by pressing the Close button on the emulator window. This depends on your window manager or operating system.

3 Emulated systems

This CHIP-8 emulator is compatible with other kinds of systems. CHIP-8 is based in the origina lspecification written for the first computers using CHIP-8 virtual machines inside, such as the COSMAC VIP. However, through the years, hobbyists interested in CHIP-8 emulation have developed emulators for other kinds of machines such as calculators or PCs, and they have extended the original specification for adding support to new features if the host hardware allowed so.

In particular, this emulator is at the moment compatible with CHIP-8 and SUPERCHIP ROMs.

3.1 CHIP-8 platform

This is the original platform for the CHIP-8 virtual machine. It is a simple architecture that has the following hardware resources.

- 4 kB of RAM memory.
- A register bank made of 16 registers, each one being 8 bits. These registers are refered to as `V0`, `V1`, ..., `V9`, `VA`, `VB`, `VC`, `VD`, `VE` and `VF`.
- A stack that can hold 16 16-bit values.
- A stack pointer for pointing to the next free location in the stack. As an example, if the stack pointer holds the value 5, that means that the next value pushed to the stack will be in the position 5. Pushing a value to the stack will increase the value of the stack pointer by 1 to make it point to the next free location. Popping something will decrease the value by 1, and then whatever is pointed by it, will be transfered to the target destination.
- A special register named I that is used by some operations such as painting to the screen.

The CHIP-8 screen is a rasterized 64 x 32 pixels display. It is monochrome and up to 2048 pixels can be represented. Usually these pixels will be black and white, but there is no official statement on this, which means that is possible for some systems to use different colors, such as green over black.

CHIP-8 also has a 16 keys keyboard, as indicated in Section 2.3 [Using the emulator], page 4. It also has a buzzer that can play sounds on a particular and single frequency.

Talking about the interpreter, the CHIP-8 specification sets 34 instructions that can manipulate the data in memory, registers and stack, and interact with other IO devices such as sending images to the screen, reading keys from the keyboard or making the speaker buzz.

3.2 SUPERCHIP platform

SUPERCHIP is an extension over the original CHIP-8 specification that appeared in the early 90s in order to run CHIP-8 games in some graphical calculators. Thanks to the new hardware capabilities of the machine, this specification adds the following features to CHIP-8:

- A new graphical mode. By default the emulator runs in compatibility mode in order to play old games that do not support the new mode, but a new mode can be enabled that has a 128x64 monochrome display, with up to 4 times more pixels to display.
- Instructions for scrolling the screen into different directions, making faster some operations that rely on this, such as letting the user see more parts of a big screen.
- Interaction with the underlying operating system that runs the SUPERCHIP emulator. For the first time, it is possible to stop the emulation without having to close the emulator. Plus, it is possible to receive input from the operating system and send output to the operating system using a new register bank called *R*, whose values can be provided before running the game, and requested after finishing the game.

There are 10 new instructions that cna be used in the programs. SUPERCHIP is still compatible with the original instruction set because these new opcodes are mapped to opcodes that would trigger on a crash on regular CHIP-8 emulators. A classic ROM can still work because the original instruction set is, of course, provided.

3.3 MEGACHIP platform

This mode is not implemented yet in this emulator, although their features makes this platform very appealing.

MEGACHIP is a new format born after the improvement of personal computers. Old limitations have no place on regular devices we have today, and therefore we can have better multimedia capabilities for running games.

MEGACHIP adds support for the following features, that should be designed to be backwards compatible with CHIP-8 and SUPERCHIP formats:

- 256x192 raster resolution.
- For the first time, multiple colors using an indexed 256 color palette.
- Up to 32 MB of RAM.
- 8 bit digital sound.

There are 11 new instructions that are of course mapped to opcodes that neither CHIP-8 nor SUPERCHIP uses. Therefore, it is possible to run old CHIP-8 and SUPERCHIP games without side effects.

4 Compatible ROM formats

In order to run the programs, it is required to have a ROM containing the instructions for the program. These instructions are executed by the virtual machine when the emulator starts.

This CHIP-8 distribution comes with a set of public domain ROMs that can be executed in order to test the features of the emulator. These ROMs are in the `examples/` directory for the software distribution package, and once installed they are placed in `/usr/local/share/doc/chip8/roms`, although the exact location may change depending on where is the package installed in.

The user can also create new ROMs using software tools or manually writing machine code into a binary file or an hexadecimal file. These files have to be provided as parameters when starting the emulator, as explained in Section 2.2 [Running the emulator], page 3.

However, there are two ways of storing the program instructions inside CHIP-8 ROMs.

4.1 Binary ROM files

A binary ROM file is a file where every byte is a byte that must be placed in the RAM memory for the emulator when the game starts. In other words, the actual opcodes are encoded. Every 2 bytes in the ROM file can be translated to an actual instruction that the machine must run. (Or to some data such as an sprite).

This is the fastest way for starting ROMs because no transformation is needed. As the emulator starts, the contents of the file are loaded in RAM memory if they fit, and the game is started. However, they need special software, such as hexadecimal editors, in order to see the contents, because they are binary and cannot be opened with regular text editors.

As an example, let's suppose that we have a binary ROM with the following contents, that we might have got using an hexadecimal file tool such as `hexdump`:

```
$ hexdump examples/PONG
0000 6a 02 6b 0c 6c 3f 6d 0c a2 ea da b6 dc d6 6e 00
0010 22 d4 66 03 68 02 60 60 f0 15 f0 07 30 00 12 1a
```

(more lines have been omitted for brevity)

When this ROM is loaded into memory, the first instruction that will be executed will be `6A02`, as these are the first two bytes in the file. Next, the instruction `6B0C` will be executed, and then `6C3F`, and so.

Binary format is the most preferred way when working with dumps coming from external sources such as actual chips containing programs for old computers using the CHIP-8 format, because they represent the program as it was.

4.2 Hexadecimal ROM files

An hexadecimal ROM file has the hexadecimal characters that are required for running a ROM file, organized in a text file that can be edited by the user in order to build their own programs and games.

It is a file that makes use of a human readable character encoding such as US-ASCII, UTF-8 or Windows-1252 and that contains only characters in the ranges "*0*" to "*9*", "*A*" to

"*F*" and "*a*" to "*f*. These ranges would equal to the US-ASCII codes `0x30-0x39`, `0x41-0x46` and `0x61-0x66`.

The contents for an hexadecimal ROM file can be inspected using regular text editors, as an example:

```
$ cat pong.hex
6A026B0C6C3F6D0CA2EADAB6DCD66E0022D46603
68026060F015F0073000121AC717770869FFA2F0
D671A2EADAB6DCD66001E0A17BFE6004E0A17B02
601F8B02DAB6600CE0A17DFE600DE0A17D02601F
8D02DCD6A2F0D67186848794603F8602611F8712
46021278463F1282471F69FF47006901D671122A
```

When an hexadecimal ROM file is given to the emulator, it must be processed. The hexadecimal ROM is converted to binary using the following method:

- It ignores any whitespace or line break.
- It reads the characters 2 by 2.
- For every pair of characters that should represent hexadecimal characters (meaning that they should match the regular expresion `([0-9A-Fa-f]{2})`, the following operation is made: `(L, R) -> (L << 4) | R`, where L is the left hexadecimal character once converted to binary and R is the right hexadecimal character once converted to binary. This conversion is made by transforming every character to the 4-bit digit they represent (`0x0` to `0xF`).

Although it has not been implemented yet, the following operations are planned:

- Ignore also tabs, in order to ignore any kind of whitespace.
- Allow comments in hexadecimal ROM files to make it easier to people to know what code snippets should do.

4.3 Corrupt ROM files

This emulator has not been tested against fuzz testing, which means that it is not possible to know at this moment what should happen and how should the emulator behave when ROMs having corrupt instructions are execute.

Fuzz testing is made by giving a random input to a program. As an example, randomly generating about 100 to 200 bytes of instructions, either in binary or hexadecimal way, and loading the contents in order to check what happens.

Even though it has not been tested, the expected behaviour should be:

- Glitches on the screen.
- Weird behaviour of the program.
- Keyboard delays where they shouldn't be.
- Sound, if available, playing through the speaker.

Anyway, it is not expected that a corrupt ROM could harm the host machine in anyway, becuase none of the operations execcuted by the program interact with the underlying operating system in a way that could have side effects. On the other side, all the operations that work with memory buffers make enough tests to minimize the risk of a buffer overflow error.

Despite all of that, this is something that hasn't been further tested. As stated by the license terms of this sfotware, available in the COPYING file, the emulator is provided AS IS, with no extra warranties.

Appendix A Hacking the emulator

By downloading the software package including source code, any user can change how the emulator works to make it behave in a different way for their personal interest, as stated in the license terms for the software package.

Some hints are given on the source code to make it easier to advanced people to modify the program. They should be specially useful for people that wants to contribute to the project by making their contributions public by patching the upstream source code via a Pull Request.

A.1 Basic source code guidelines

This program is made using the C programming language. Although any port to a different programming language is good to have as a side project, this is the official language for this project. Therefore, any patch that attempts to change that is not welcome. The project is good using C; I don't want to change to C++, Python or whatever funky language is the trendiest at this moment.

This program uses GNU Autotools for the build tool. Although GNU Autotools is not loved by everyone, this is the official tool for this project. So, any attempt to change the build tool to CMake, Gradle or regular Makefiles is not welcome.

A.2 Style guides

Although this is not an official GNU project, I make use of most of their style guidelines. Any contributor that wants to make public changes to the project must keep the following rules in mind.

- Line width is limited to 80 characters.
- Indent uses 4 spaces. No tabs, no 8 spaces, no 2 spaces.
- Functions should be defined by keeping the datatype in a different line than the function name and parameters. Check the source code for an example.
- Parameters and variables should have a easy to understand name. Some short names are allowed when by context is clear, such as using `P`, `X`, `Y` or `K` for variables related to an opcode, because these are the names given to the opcode structures. Long names when they are not required is not pretty.
- Comments should be added when neccesary. Don't comment every line, but don't add long chunks of code with no clear explanation of what they do.
- Long functions should be split in multiple small functions to make the development easier, plus to make testing easier whenever there are functions that should be tested.

A.3 Opcodes table

In order to implement the opcodes, the following structure has been made.

There is a common definition for every opcode function in the code, declared as a type named `opcode_table_t`, located as a private *typedef* in the file `cpu.c` at the *lib8* subproject.

After that, 16 functions are declared, every one compatible with that typedef. Their names are `nibble_0`, `nibble_1`, `nibble_2`... These functions have been declared as private functions using the same parameters:

```
static void
nibble_0(struct machine_t* cpu, word opcode);

static void
nibble_1(struct machine_t* cpu, word opcode);

static void
nibble_2(struct machine_t* cpu, word opcode);
```

Every one of these functions executes opcodes with a P value matching the one in the function name. As an example, the opcode `6104` would be executed by `nibble_6`, since $\mathbf{P(6104) = 6}$.

Then, there is an array pointing to every opcode function named `nibbles[]`. These functions are sorted by P value, so that `nibbles[6]` will return a pointer to the function `nibble_6`, and `nibbles[0xB]` will return a pointer to the function `nibble_B`.

Thanks to this indirection, the processing logic for an opcode is easier since no big switch is required on code. Using this indirection, it is more straightforward to know which function should process each opcode. Although, some functions such as `nibble_6` will make use of switch anyway since there are multiple opcodes sharing the same P value.

Appendix B GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

   The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

    "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

    "CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

    "Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

    An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

    The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.