Easy R8C/M16C/M32C/R32C Flash Programming

DJ Delorie

September 15, 2009

Note: This document is formatted with the free *Liberation* fonts[1] and the freely redistributable *Timing Diagram* font[2]. If you don't have those installed, this document may not render properly.

The home page of this document is http://people.redhat.com/dj/m32c/

1 https://www.redhat.com/promo/fonts/
2 http://www.pcserviceselectronics.co.uk/fonts/

Easy R8C/M16C/M32C/R32C Flash Programming

This document will provide a basic understanding of how the R8C/M16C/M32C/R32C family of microprocessors can be programmed with new flash contents. To make things easier, we'll refer to any of these chips as "the chip" unless we're talking about a specific one. We refer to the machine providing the programming signals as "the host" regardless of how these signals are generated (serial port, USB, something else).

The simplest method to program from a host is the asynchronous serial mode, often called "serial mode 2". This mode lets you use a standard serial port, such as is found on most PCs, and two jumpers. One jumper is for nRESET, and the other is for the MODE or CNVss pin (different chips call them different things, we'll use MODE in this document). In addition, each chip may have other pins which must be pulled high or low for this mode; check the chip's hardware manual for details. Usually a high-valued resistor is sufficient for these pins. The serial pins on the chip are TTL logic, *not* RS-232 logic. You'll need a converter to connect to a standard serial port, or use a TTL-level USB-to-serial cable like FTDI's TTL-232R (5 volts) or TTL-232R-3V3 (3.3 volts). You can also connect the pins to the UART pins of some other MCU chip such as another M32C. You just need two serial pins (Tx and Rx), and two GPIO pins (nRESET and MODE).

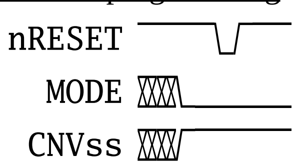
For example, in one of my projects, I have five R8C chips connected to an XScale chip which can program them as well as communicate with them. The five R8C RX lines are all tied together, and a multiplexer chooses which of the five TX lines the XScale listens to. The MODE lines are all tied together, and five XScale GPIO independently reset the R8Cs. The XScale can thus reset only the relevant R8C chip and program it; the others will ignore their MODE and RX lines.

Each chip has two program memories. The main memory is the user flash memory that holds your application, which is what we want to reprogram. There is a second smaller memory in each chip that contains a small bootloader program. When you want to program a new application, you tell the chip to boot from the second memory, and the bootloader there helps you download new data into the main flash memory through a simple communications protocol.

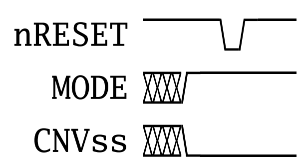
Reset

The chip decides which memory to boot from (user flash or bootloader) by checking the MODE pin at reset. If the MODE pin is high (it's normally pulled high with a resistor), the chip boots from user flash and runs your software. If it's low, the bootloader runs instead, and you can download new contents into the flash. For chips with CNVss, the functionality is the same but the logic is inverted:

Reset for programming



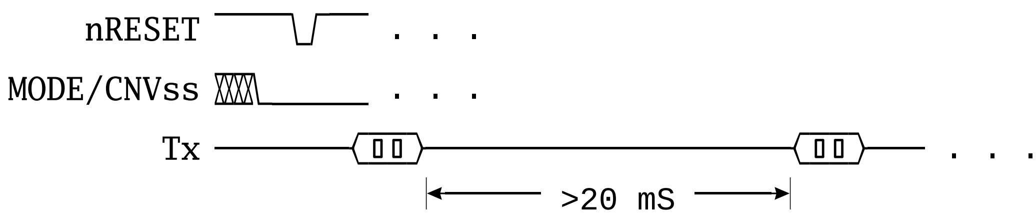
Reset to Run



You would normally leave the MODE pin unchanged except just before a reset. Fortunately, the timing is very loose on these - you can control these two pins with jumpers. On most eval boards, the pins are pulled high (or low, for CNVss) with resistors, so what you do is put the MODE jumper in place (which pulls MODE to ground), press the reset button, then run the host program which talks to the bootloader. When the programming is done, remove the MODE jumper, press the reset button, and your software runs.

Auto-Baud

In mode 2, the first thing you do is teach the bootloader what baud rate to use, since the bootloader doesn't know how fast the chip's clock is. Wait 3mS or so for the chip to come out of reset, then send 16 NUL bytes spaced 20 mS or more (I use 40 mS) apart. These bytes should be sent at 9600 baud, 8 bits, no parity, one stop bit. Different chips require different delays between bytes, but this should be long enough to work with them all. In this diagram we show the sequence - set MODE, reset, start sending NUL bytes. Note that all values shown for the Tx and Rx lines are in hex, so ⟨65⟩ is 0x65 (decimal 101), not decimal 65.



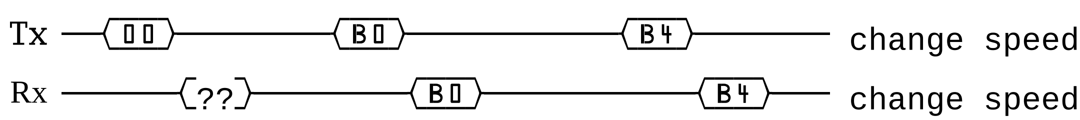
After sending the last byte, wait 2mS and then read any data that might have been sent back to the host. Sometimes there's a glitch at reset that looks like a byte, and some chips transmit a byte at the end of synchronization. By reading these optional bytes and clearing your receive buffer, you ensure you're in sync with the bootloader.

Baud Rate

The next step is to verify the autobaud rate, and set the desired baud rate. The initial synchronization is done at 9600 baud, but you can ask the chip to use a faster baud rate if your hardware (host and chip) support it. Baud changes are done with single byte commands, and the chip responds by sending you the same byte back:

9600	B0
19200	B1
38400	B2
57600	B3
115200	B4

The request is sent out with the old baud rate, and the chip responds with the **old** baud rate before changing it. Send the baud rate command, check the response, and if it's correct then change the baud rate at your end. So, after the last NUL byte is sent, read any pending bytes and send a request for 9600 baud to verify communications, then request your new baud rate and verify it:

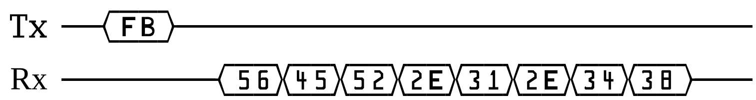


Note that not all baud rates will work; which ones work depend on what speeds your chips are running at. For some speeds, the chip might not be able to generate a baud clock accurately enough. Try them all until you find one that works reliably.

Bootloader Version

The next thing to do is ask the bootloader for its version number. It's not that useful from a program's point of view, but you can use it to make sure the baud rate is reliable. The request is a single byte

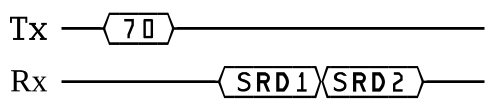
(0xFB), the response is eight bytes, intended to be human readable:



In the above example, the version string is "`VER.1.48`". The chips I've seen all follow this pattern (with various digits, of course), so my utility checks for each character to be the right type (specific letters, any digits, etc).

Status

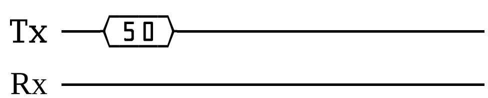
The next command to know is how to check the status of the bootloader. We use this command at various points to make sure the previous operation completed successfully. The request is a single byte (0x70), and response is two bytes:



There are only a couple of bits that we care about. Bit 7 of SRD1 is set when the bootloader is ready for another command. Bit 5 is set when an erase command fails. Bit 4 is set when a program command fails. Bits 3 and 2 of SRD2 tell you if you've "unlocked" the flash by providing the correct unlock key:

`xxxx 00xx`	No key provided yet
`xxxx 01xx`	Wrong key provided
`xxxx 11xx`	Correct key provided

Throughout this manual, when we say "check the status" we mean to run this command and check the status bits that are returned. You can also clear the status bits:

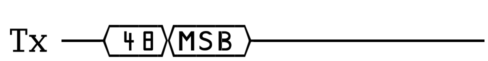


You should clear the status before erasing or programming the flash, as the error bits for those are cumulative.

Ok, now that we know how to talk to the bootloader, we need to start the process of downloading the new application into the flash. We first have to unlock the chip. Then we erase the flash. Then, for each page (256 bytes) of memory that needs to be programmed, we download that page then read it back to verify it.

Address Sizes

A quick note on address sizes... The commands we send to the chip all have 24-bit address parameters. This is because the protocol was designed for the M16C family, which has a 20-bit address. However, with the introduction of the R32C, which has a 32-bit address, a new command was added to provide the upper 8 bits of the address separately. Throughout this manual, where you see a command that takes a 24-bit address, for the R32C, you prefix it with this sequence (0x48, not 0x4B) to provide the extra bits:



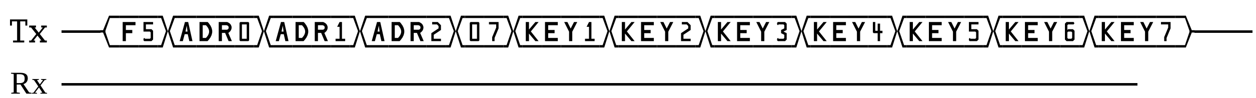
For example, to issue a fictitious command CMD with address 0x12345678, we'd do something like this:

Unlock

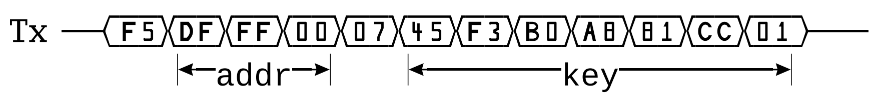
To unlock the chip, we need to know the flash locking code that was last programmed into it. Most development tools use either all zeros or all ones (0xff) for the default key, so if you don't know the right key you can try those. Each chip reserves seven bytes near the end (high addresses) of the flash, which vary by family:

Key Byte	R8C	M16C	M32C	R32C
1	0x00FFDF	0x0FFFDF	0xFFFFDF	0xFFFFFFE8
2	0x00FFE3	0x0FFFE3	0xFFFFE3	0xFFFFFFE9
3	0x00FFEB	0x0FFFEB	0xFFFFEB	0xFFFFFFEA
4	0x00FFEF	0x0FFFEF	0xFFFFEF	0xFFFFFFEB
5	0x00FFF3	0x0FFFF3	0xFFFFF3	0xFFFFFFEC
6	0x00FFF7	0x0FFFF7	0xFFFFF7	0xFFFFFFED
7	0x00FFFB	0x0FFFFB	0xFFFFB	0xFFFFFFEE

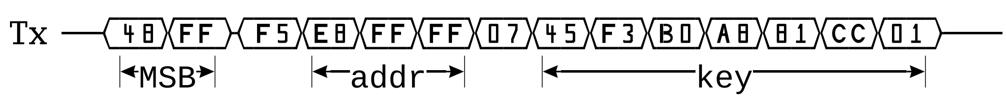
Always check the status after attempting to unlock the chip, to see if the key you provided is correct. The unlock request is one byte 0xF5 followed by the address of key byte one (from the above table, three bytes, least significant byte first), followed by the size of the key (0x07), followed by the seven key bytes (in the order indicated by the above table):



For example, to program an R8C/1B with a key of 45,F3,B0,A8,81,CC,01 we'd use these bytes to unlock it:



To program an R32C with a key of 45,F3,B0,A8,81,CC,01 we'd need to provide a 32-bit address:



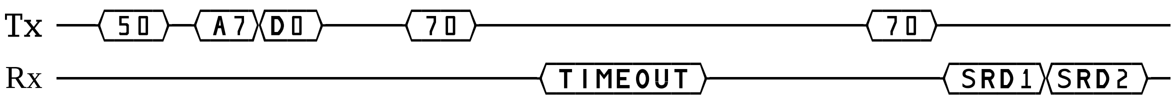
In my utilities, I try unlocking first all zeros, then all ones (0xff), stopping when I see the "correct key" code in the status response.

Ok, now for the actual programming part. There are three command you need to know how to do - erase, program, and verify. The programming sequence is to erase everything, then program the pages you need to program, reading each one back to verify it got copied correctly.

Also, in my utilities I forcibly set the flash locking key in the image I'm downloading to be all zeros, in case the tools don't set them to something meaningful. Also, I set the R8C watchdog byte to 0xff. If you actually use these features, you can include command-line options to leave them alone or set them to specific values.

Erase All

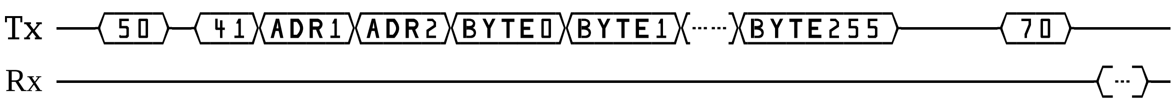
Erasing everything is easy. It's not as fast or as optimal as just erasing the blocks you need, but it's universal. Since this command takes a while to run, special precautions will be taken to properly clear and read the status.



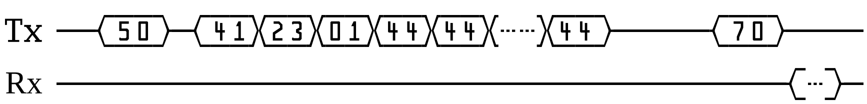
After you issue the format command (0xA7, 0xD0), set up a 1 second timeout on receive and start asking for status. If the status read times out, keep trying. I try 15 times. If you get data back but it doesn't make sense, or if it indicates that the erase isn't done, wait 500mS and flush any other bytes you may have received before trying again. This helps re-synchronize the status read, so that you're sure you're reading SRD1 first and not the old SRD2 first.

Program Page

A "page" of memory is 256 bytes, aligned on a 256 byte boundary. Since the address always ends in 0x00, only two bytes of address need be downloaded*. Send the middle byte first, followed by the high (MSB) byte, followed by 256 bytes of data:



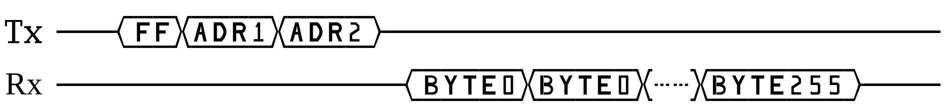
The same timeout/retry instructions apply here as for when erasing. So, for example, to write all fours to address 0x12300:



*For R32C, send the top 8 bits (bits 31-24) separately, with bits 23-16 then bits 15-8 in the command.

Page Read

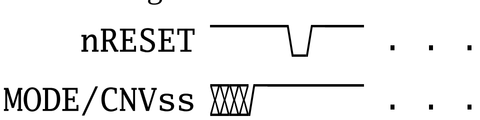
After you program each page, you should read it back to ensure it was programmed correctly. In my utilities, I retry the program/read/verify cycle three times before giving up (at which point, you need to erase the flash and start again). The sequence to read a page is similar to programming a page, in that you send the read page command (0xff), two of the three address bytes, then read 256 data bytes back:



Note that there is no need to clear or read the status, however it may be prudent to drain any bytes waiting on the Rx line to ensure you're in sync with the bootloader, before doing this command.

Done!

Once you've downloaded all the pages you need (don't forget the page that includes the reset vector), there's nothing left to do but reset in user mode and run your application!



Console

In my utilities, I have a command line option to maintain the connection to the serial port at this point, acting like a mini terminal emulator. At the very least, if your utility prints any received data to your screen, you can use the serial port on the chip as a "debug" stream. Even if you have to manually reset the chip, having this "console" open to the chip when you do, lets you see any messages that might be printed only once when your application starts (either through printf or some other function).

Conclusion

With the information in this document, you should be able to download your application into any of the R8C/M16C/M32C/R32C chips. However, keep in mind that you still need to read the chip-specific documentation (that chapter is usually called "Flash Memory Version" in the Renesas specs) and follow any requirements they indicate there.

Also, there are far more commands than this document covers, such as partial erasing, block locking, etc. To use those, you do need to know the specifics of the flash modules in each chip.

Renesas does publish a few specification documents about these commands, from which I've built up my utilities. Search the web for REJ05B0734, REJ05B0599, and MCU-M16C-95-0302.

GNU Free Documentation License

Version 1.2, November 2002

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.

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 "Acknowledgments", "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 "Acknowledgments" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgments 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 "Acknowledgments", 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 "Acknowledgments", "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 for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

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.