Libembroidery v0.1 Manual

Table of Contents

- Bibliography
- GNU Free Documentation License

What is libembroidery ?

libembroidery is the underlying library that is used by Embroidermodder 2 and is developed by the Embroidermodder team. It handles over 45 different embroidery specific formats as well as several non-embroidery specific vector formats.

It also includes a CLI called `embroider` that allows for better automation of changes to embroidery files and will be more up-to date than the Embroidermodder 2 GUI.

Finally there are some shell tools in `embtools.sh/embtools.bat` to help with calls to ImageMagick and other programs that can help produce new embroidery machine files.

Libembroidery is distributed under the permissive zlib licence, see the LICENSE file.

If you want to find a simple fix to contribute see the *To Do* section of the manual.

Build

libembroidery and EmbroiderModder 2 use CMake builds so if you are building the project to use as a library we recommend you run:

Linux or MacOS

```
git clone https://github.com/Embroidermodder/libembroidery
cd libembroidery
./embtools.sh build
```

Windows (or any system without Bash) (IN DEVELOPMENT)

```
git clone https://github.com/Embroidermodder/libembroidery
cd libembroidery
embtools.bat build
```

These builds both the static and shared versions of the library as well as the command line program `embroider`.

Usage

For basic use, we recommend you build as above then run:

```
./embroider
```

which will print out this advice on how to use these tools without digging straight into the rest of this manual.

```
EMBROIDER
    A command line program for machine embroidery.
    Copyright 2013-2021 The Embroidermodder Team
    Licensed under the terms of the zlib license.

    https://github.com/Embroidermodder/libembroidery
    https://embroidermodder.org

Usage: embroider [OPTIONS] fileToRead...

Conversion:
-t, -to         Convert all files given to the format specified
                by the arguments to the flag, for example:
                    $ embroider -t dst input.pes
                would convert \"input.pes\" to \"input.dst\"
                in the same directory the program runs in.

                The accepted input formats are (TO BE DETERMINED).
                The accepted output formats are (TO BE DETERMINED).

Output:
-h, -help       Print this message.
-f, -format     Print help on the formats that
                embroider can deal with.
-q, -quiet      Only print fatal errors.
-V, -verbose    Print everything that has reporting.
-v, -version    Print the version.

Graphics:
-c, -circle     Add a circle defined by the arguments
                given to the current pattern.
-e, -ellipse    Add a circle defined by the arguments
                given to the current pattern.
-l, -line       Add a line defined by the arguments
                given to the current pattern.
-P, -polyline   Add a polyline.
-p, -polygon    Add a polygon.
-s, -satin      Fill the current geometry with satin
                stitches according
                to the defined algorithm.
-S, -stitch     Add a stitch defined by the arguments
                given to the current pattern.
```

```
Quality Assurance:
    -test       Run the test suite.
```

For each of the flags described here we will go into greater detail in this manual.

Abstract

The *Embroidermodder 2* project is a collection of small software utilities for manipulating, converting and creating embroidery files in all major embroidery machine formats. The program *Embroidermodder 2* itself is a larger graphical user interface (GUI) which is at the heart of the project.

This manual, the website (`embroidermodder.org`), mobile embroidery format viewers and tools (`iMobileViewer`, `MobileViewer`), the core library of functions (`libembroidery`) and CLI (`embroider`) are all tools to make the standard user experience of working with an embroidery machine better without expensive software which is locked to specific manufacturers and formats. But ultimately we hope that the core *Embroidermodder 2* is a practical, ever-present tool in larger workshops, small cottage industry workshops and personal hobbyist's bedrooms.

Embroidermodder 2 is licensed under the zlib license and we aim to keep all of our tools open source and free of charge. If you would like to support the project check out our Open Collective group. If you would like to help, please join us on GitHub. This document is written as developer training as well helping new users (see the last sections) so this is the place to learn how to start changing the code.

The Embroidermodder Team

Credits for Embroidermodder 2, libembroidery and all other related code

Please note that this file in not in alphabetical order. If you have contributed and wish to be added to this list, create a new credit section and increment the number. Fill it in with your information and submit it to us.

Here is a summary of the values used:

- Name: The full name of the contributor starting with first name.

- GitHub: The GitHub account name of the contributor (in parentheses).
- CoreDeveloper: This is reserved for long term contributors.
- Documentation: If you have contributed changes to README files or help files, set this to true.
- Artwork: If you have contributed artwork or related changes, set this to true
- BugFixes: If you have contributed bug fixes or added new features, set this to true.
- Translation: If you have provided language translations, set this to true.
- Designs: If you have contributed an embroidery design sample, set this to true.
- Bindings: If you have contributed programming language bindings for libembroidery, set this to true.
- Commands: If you have contributed a command for Embroidermodder 2, set this to true.

Credits

1. Jonathan Greig (`redteam316`) CoreDeveloper, Artwork, Documentation, Designs, Commands
2. Josh Varga (`JoshVarga`) CoreDeveloper
3. Jens Diemer (`jedie`) Documentation
4. Kim Howard (`turbokim`) BugFixes
5. Martin Schneider (`craftoid`) Documentation
6. Edward Greig (`Metallicow`) Artwork, BugFixes, Commands
7. Sonia Entzinger Translation
8. SushiTee (`SushiTee`) BugFixes
9. Vathonie Lufh (`x2nie`) BugFixes, Bindings
10. Nina Paley Designs
11. Theodore Gray Designs
12. Jens-Wolfhard Schicke-Uffmann (`Drahflow`) BugFixes
13. Emmett Lauren Garlitz - Some Little Sandy Rd, Elkview, West by GOD Virginia (`Oll Em`)
14. Robin Swift (`robin-swift`) CoreDeveloper, Documentation

Introduction

libembroidery aims to be a low-level library for reading, writing, and programmatically manipulating digital embroidery files, available to in many different programming languages.

Currently, libembroidery is the underlying library that is used by Embroidermodder 2 and is developed by the Embroidermodder team. It handles over 45 different embroidery specific formats as well as several non-embroidery specific vector formats.

It also includes a CLI called `embroider` that allows for better automation of

changes to embroidery files and will be more up-to date than the Embroidermodder 2 GUI.

Build

libembroidery and EmbroiderModder 2 use CMake builds so if you are building the project to use as a library we recommend:

Generic System with Bash

```
git clone https://github.com/Embroidermodder/libembroidery
cd libembroidery
embtools.sh build
```

Windows

```
git clone https://github.com/Embroidermodder/libembroidery
cd libembroidery
embtools.bat build
```

This builds both the static and shared versions of the library as well as the command line program `embroider`.

License

Coding Standards

A basic set of guidelines to use when submitting code.

Naming Conventions

- Name variables and functions intelligently to minimize the need for comments.
- It should be immediately obvious what information it represents.
- Short names such as x and y are fine when referring to coordinates.
- Short names such as i and j are fine when doing loops.
- Variable names should be "camelCase", starting with a lowercase word followed by uppercase word(s).
- C++ Class Names should be "CamelCase", using all uppercase word(s).
- C Functions that attempt to simulate namespacing, should be "nameSpace_camelCase".
- All files and directories shall be lowercase and contain no spaces.

- Tabs should not be used when indenting. Setup your IDE or text editor to use 4 spaces.

Braces

For functions: please put each brace on a new line.

```
void function_definition(int argument)
{

}
```

For control statements: please put the first brace on the same line.

```
if (condition) {

}
```

Use exceptions sparingly.

Do not use ternary operator (?:) in place of if/else.

Do not repeat a variable name that already occurs in an outer scope.

Comments

When writing code, sometimes there are items that we know can be improved, incomplete or need special clarification. In these cases, use the types of comments shown below. They are pretty standard and are highlighted by many editors to make reviewing code easier. We also use shell scripts to parse the code to find all of these occurrences so someone wanting to go on a bug hunt will be able to easily see which areas of the code need more love.

libembroidery is written in C and adheres to C89 standards. This means that any C99 or C++ comments will show up as errors when compiling with gcc. In any C code, you must use:

```
/* C Style Comments */
/* TODO: This code clearly needs more work or further review. */
/* BUG: This code is definitely wrong. It needs fixed. */
/* HACK: This code shouldn't be written this way or I don't feel right about it. There may a
/* WARNING: Think twice (or more times) before changing this code. I put this here for a goo
/* NOTE: This comment is much more important than lesser comments. */
```

We advise you also use these style of comments in C++ code in Embroidermodder 2 as well for consistency.

Wrapper Functions

.

Formats

Toyota Embroidery Format (.100)

- Stitch Only Format.
- Uses an external color file.

☒ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

The stitch encoding is in 4 byte chunks.

Toyota Embroidery Format (.10o)

- Stitch Only Format.
- Uses an external color file.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

The stitch encoding is in 3 byte chunks.

Bernina Embroidery Format (.art)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

We don't know much about this format. TODO: Find a source. ## Bitmap Cache Embroidery Format (.bmc)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

We don't know much about this format. TODO: Find a source.

Bits and Volts Embroidery Format (.bro)

- Stitch Only Format.
- Uses an external color file.

☒ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read

☐ Well Tested Write

The header is 256 bytes. There's a series of unknown variables in the header.

The stitch list uses a variable length encoding which is 2 bytes for any stitch

Melco Embroidery Format (.cnd)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

We don't know much about this format. TODO: Find a source.

Embroidery Thread Color Format (.col)

- Stitch Only Format.

☒ Basic Read Support
☒ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Singer Embroidery Format (.csd)

- Stitch Only Format.

☒ Basic Read Support
☒ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Comma Separated Values (.csv)}

- Stitch Only Format.

☒ Basic Read Support
☒ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Barudan Embroidery Format (.dat)}

- Stitch Only Format.

☒ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Melco Embroidery Format (.dem)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Barudan Embroidery Format (.dsb)}

- Stitch Only Format.

☒ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Tajima Embroidery Format (.dst)}

- Stitch Only Format.

☒ Basic Read Support
☒ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

.DST (Tajima) embroidery file read/write routines Format comments are thanks to tspilman@dalcoathletic.com who's notes appeared at http://www.wotsit.org under Tajima Format.

Header The header seems to contain information about the design. Seems to be ASCII text delimited by 0x0D (carriage returns). This must be in the file for most new software or hardware to consider it a good file! This is much more important than I originally believed. The header is 125 bytes in length and padded out by 0x20 to 512 bytes total. All entries in the header seem to be 2 ASCII characters followed by a colon, then it's value trailed by a carriage return.

C memory	*Description*
`char LA[16+1];`	First is the 'LA' entry, which is the design name with no path or extension information. The blank is 16 characters in total, but the name must not be longer that 8 characters and padded out with spaces (0x20).

C memory	*Description*
`char ST[7+1];`	Next is the stitch count ST, this is a 7 digit number padded by leading zeros. This is the total stitch count including color changes, jumps, nups, and special records.
`char CO[3+1];`	Next, is CO or colors, a 3 digit number padded by leading zeros. This is the number of color change records in the file.
`char POSX[5+1];`	Next is +X or the positive X extent in centimeters, a 5 digit non-decimal number padded by leading zeros.
`char NEGX[5+1];`	Following is the -X or the negative X extent in millimeters, a 5 digit non-decimal number padded by leading zeros.
`char POSY[5+1];`	Again, the +Y extents.
`char NEGY[5+1];`	Again, the -Y extents.
`char AX[6+1]; char AY[6+1];`	AX and AY should express the relative coordinates of the last point from the start point in 0.1 mm. If the start and last points are the same, the coordinates are (0,0).
`char MX[6+1]; char MY[6+1];`	MX and MY should express coordinates of the last point of the previous file for a multi-volume design. A multi-volume design means a design consisted of two or more files. This was used for huge designs that can not be stored in a single paper tape roll. It is not used so much (almost never) nowadays.
`char PD[9+1];`	PD is also storing some information for multi-volume design.

Uses 3 byte per stitch encoding with the format as follows:

The 3 byte encoding for the dxf format.

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Byte 0	y+1	y-1	y+9	y-9	x-9	x+9	x-1	x+1
Byte 1	y+3	y-3	y+27	y-27	x-27	x+27	x-3	x+3

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Byte 2	jump	color change	y+81	y-81	x-81	x+81	set	set

T01 and Tap appear to use Tajima Ternary.

Where the stitch type is determined as:

- Normal Stitch `00000011 0x03`
- Jump Stitch `10000011 0x83`
- Stop/Change Color `11000011 0xC3`
- End Design `11110011 0xF3`

Inclusive or'ed with the last byte.

Note that:

1. The max stitch length is the largest sum of `1+3+9+27+81=121` where the unit length is 0.1mm so 12.1mm.
2. The coordinate system is right handed.

ZSK USA Embroidery Format (.dsz)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Drawing Exchange Format (.dxf)}

- Graphics format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Embird Embroidery Format (.edr)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Elna Embroidery Format (.emd)}

- Stitch Only Format.

☐ Basic Read Support

☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Melco Embroidery Format (.exp)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Eltac Embroidery Format (.exy)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Sierra Expanded Embroidery Format (.eys)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Smoothie G-Code Embroidery Format (.fxy)?

Fortron Embroidery Format (.fxy)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Great Notions Embroidery Format (.gnc)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Gold Thread Embroidery Format (.gt)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Husqvarna Viking Embroidery Format (.hus)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Inbro Embroidery Format (.inb)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Embroidery Color Format (.inf)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Janome Embroidery Format (.jef)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Pfaff professional Design format (.ksm)

Back

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read

☐ Well Tested Write

Pfaff Embroidery Format (.max)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Mitsubishi Embroidery Format (.mit)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Ameco Embroidery Format (.new)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Melco Embroidery Format (.ofm)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Pfaff PCD File Format

Back

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

The format uses a signed 3 byte-length number type.

See the description here5 for the overview of the format.

For an example of the format see11.

Pfaff Embroidery Format (.pcm)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Pfaff Embroidery Format (.pcq)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Pfaff Embroidery Format (.pcs)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write ## Brother Embroidery Format (.pec)

Back

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Brother Embroidery Format (.pel)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Brother Embroidery Format (.pem)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Brother Embroidery Format (.pes)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

```
struct PEShdr
{
char PESId[8];
DWORD PECOffset;
};
```

Brother Embroidery Format (.phb)

Back

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Brother Embroidery Format (.phc)}

Back

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

AutoCAD Embroidery Format (.plt)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

RGB Embroidery Format (.rgb)}

- Color only format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read

☐ Well Tested Write

Janome Embroidery Format (.sew)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Husqvarna Viking Embroidery Format (.shv)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Sunstar Embroidery Format (.sst)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Data Stitch Embroidery Format (.stx)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Scalable Vector Graphics (.svg)}

- Graphics format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Pfaff Embroidery Format (.t01)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support

☐ Well Tested Read
☐ Well Tested Write

Pfaff Embroidery Format (.t09)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Happy Embroidery Format (.tap)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

ThredWorks Embroidery Format (.thr)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Text File (.txt)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Barudan Embroidery Format (.u00)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Barudan Embroidery Format (.u01)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Pfaff Embroidery Format (.vip)

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Pfaff Embroidery Format (.vp3)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

Singer Embroidery Format (.xxx)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

ZSK USA Embroidery Format (.zsk)}

- Stitch Only Format.

☐ Basic Read Support
☐ Basic Write Support
☐ Well Tested Read
☐ Well Tested Write

On Embedded Systems

The library is designed to support embedded environments, so it can be used in CNC applications.

Compatible Boards

We recommend using an Arduino Mega 2560 or another board with equal or greater specs. That being said, we have had success using an Arduino Uno

R3 but this will likely require further optimization and other improvements to ensure continued compatibility with the Uno. See below for more information.

Arduino Considerations

There are two main concerns here: Flash Storage & SRAM.

libembroidery continually outgrows the 32KB of Flash storage on the Arduino Uno and every time this occurs, a decision has to be made as to what capabilities should be included or omitted. While reading files is the main focus on arduino, writing files may also play a bigger role in the future. Long term, it would be most practical to handle the inclusion or omission of any feature via a single configuration header file that the user can modify to suit their needs.

SRAM is in extremely limited supply and it will deplete quickly so any dynamic allocation should occur early during the setup phase of the sketch and sparingly or not at all later in the sketch. To help minimize SRAM consumption on Arduino and ensure libembroidery can be used in any way the sketch creator desires, it is required that any sketch using libembroidery must implement event handlers. See the ino-event source and header files for more information.

There is also an excellent article by Bill Earl on the Adafruit Learning System which covers these topics in more depth: http://learn.adafruit.com/memories-of-an-arduino?view=all.

Space

Since a stitch takes 3 bytes of storage and many patterns use more than 10k stitches, we can't assume that the pattern will fit in memory. Therefore we will need to buffer the current pattern on and off storage in small chunks. By the same reasoning, we can't load all of one struct beore looping so we will need functions similar to binaryReadInt16 for each struct.

This means the EmbArray approach won't work since we need to load each element and dynamic memory management is unnecessary because the arrays lie in storage.

TODO: Replace EmbArray functions with embPattern load functions.

Tables

All thread tables and large text blocks are too big to compile directly into the source code. Instead we can package the library with a data packet that is compiled from an assembly program in raw format so the specific padding can be controlled.

In the user section above we will make it clear that this file needs to be loaded on the pattern USB/SD card or the program won't function.

TODO: Start file with a list of offsets to data with a corresponding table to load into with macro constants for each label needed.

Current Pattern Memory Management

It will be simpler to make one file per EmbArray so we keep an EmbFile* and a length, so no malloc call is necessary. So there needs to be a consistent tmpfile naming scheme.

TODO: For each pattern generate a random string of hexadecimal and append it to the filenames like `stitchList_A16F.dat`. Need to check for a file which indicates that this string has been used already.

Special Notes

Due to historical reasons and to remain compatible with the Arduino 1.0 IDE, this folder must be called "utility' '. Refer to the arduino build process for more info: https://arduino.github.io/arduino-cli/0.19/sketch-build-process/

libembroidery relies on the Arduino SD library for reading files. See the ino-file source and header files for more information.

Experimental Projects

Anything contained within the experimental/ folder is unstable and unsupported. Items in here may eventually be moved elsewhere or may not. If you have an interesting concept or prototype that you would like us to add, contact us and we may add it here.

Dependencies

To build additional language bindings for libembroidery from source you will need at least:

Ubuntu repository packages:

```
sudo apt-get install swig python-dev
```

Mac

```
brew install swig
```

An example for building for use in C#

```
swig -csharp -o ./csharp/binding/swig_wrap.c -I../ swig.i
```

The Assembly Split

One problem to the problem of supporting both systems with abundant memory (such as a 2010s or later desktop) and with scarce memory (such as embedded systems) is that they don't share the same assembly language. To deal with this:

there will be two equivalent softwares which are hand engineered to be similar but one will be in C and the other in the assembly dialects we support.

All assembly will be intended for embedded systems only, since a slightly smaller set of features will be supported. However, we will write a `x86` version since that can be tested.

That way the work that has been done to simplify the C code can be applied to the assembly versions.

Utility Functions

Avoiding the use of libc

Avoiding the use of libm

The cos and sin functions are calculated using

The arctan2 function in the source is calculated using Euler's series for the inverse tangent[**?**]:

```
\begin{equation}
\tan^{-1}(x) = \sum_{n=0}^{\infty} \frac{2^{2n}(n!)^{2}}{(2n+1)!} \frac{x^{2n+1}}{(1+x^{2})^
\end{equation}
```

Embroidermodder Website

https://embroidermodder.org

This is the main website and documentation hub for the Embroidermodder project.

For a list of what we're currently working on please see to-do.md.

For specifics about the embedded systems development see embedded.md.

Build

To build the documentation run `make`. This should run no problem on a normal Unix-like environment assuming pandoc is available.

- Pandoc creates the content of the page by converting the markdown to html.
- Pandoc also creates the printer-friendly documentation from the same markdown.
- Markdown acts as a go-between because it is easy to alter directly in the GH editor.

This way: 1. We write one set of documents for all projects. 2. The website can be simple and static, supporting machines that don't run javascript. 3. We control the styling of each version independently of our editing (Markdown) version 4. The printer-friendly documentation can have nicely rendered fonts and well placed figures.

Features

Bindings

Bindings for libembroidery are maintained for the languages we use internally in the project, for other languages we consider that the responsibility of other teams using the library.

So libembroidery is going to be supported on:

```
* x86 systems as assembly (to aid writing assembly for other systems)
* avr systems as assembly (for arduino)
* arm systems as assembly (for other embedded systems)
* C (by default)
* C++ (also by default)
* Java (for the Android application MobileViewer)
* Swift (for the iOS application iMobileViewer)
```

For C# we recommend directly calling the function directly using the DllImport feature:

```
    [DllImport("libembroidery.so", EntryPoint="readCsv")]
```

see this StackOverflow discussion for help.

For Python you can do the same using ctypes.

To Flag

Circle Flag

Ellipse Flag

Line Flag

Polyline Flag

Polygon Flag

Satin Flag

Stitch Flag

Test Suite

Threads

- DXF Color Table
- HUS Color Table
- JEF Color Table
- PCM Color Table
- PEC Color Table

Other Supported Thread Brands

The thread lists that aren't preprogrammed into formats but are indexed in the data file for the purpose of conversion or fitting to images/graphics.

- Arc Polyester
- Arc Rayon
- Coats and Clark Rayon
- Exquisite Polyester
- Fufu Polyester
- Fufu Rayon
- Hemingworth Polyester
- Isacord Polyester
- Isafil Rayon
- Marathon Polyester
- Marathon Rayon
- Madeira Polyester
- Madeira Rayon
- Metro Polyester
- Pantone
- Robison Anton Polyester
- Robison Anton Rayon
- Sigma Polyester

- Sulky Rayon
- ThreadArt Rayon
- ThreadArt Polyester
- ThreaDelight Polyester
- Z102 Isacord Polyester

Development

A *right-handed coordinate system* is one where up is positive and right is positive. Left-handed is up is positive, left is positive. Screens often use down is positive, right is positive, including the OpenGL standard so when switching between graphics formats and stitch formats we need to use a vertical flip (`embPattern_flip`).

`0x20` is the space symbol, so when padding either 0 or space is preferred and in the case of space use the literal ' '. Use the macros:

```
#define PAD_SPACE(n) \
    embFile_pad(currentFile, ' ', n)
#define PAD_ZERO(n) \
    embFile_pad(currentFile, 0, n)
```

to save writing functions that may not inline.

Design Philosophy and Coding Standards

1. No GUI code will be present in `libembroidery`.
2. The library will be written in pedantic ANSI C/C90, aiming for no compiler warnings under GCC for compatibility with the most possible platforms.
3. The command line program `embroider` targets 32-bit and 64 bit systems that comply to POSIX or current Windows/MacOS standards. Practically, this means we only call C99 standard library functions.
4. Don't use any of the C Standard Library. All interfacing should be done by linux system calls or their equivalents.
5. Never use dynamic memory allocation (malloc/free) all memory that would need those functions uses temporary files instead.
6. Share heavy memory usage between functions, for example use embBuffer for buffering headers rather than having a separate `char header[]` variables.

Image Fitting

A currently unsolved problem in development that warrants further research is the scenario where a user wants to feed embroider an image that can then be .

Finding fixes

To find jobs marked within the source code rather than the list above, use:

```
grep "TODO" *.c *.h
```

Testing

Build `embroider` then run:

```
./embroider -test &> report.txt
```

If any of the tests return a fail in the summary (see `tail report.txt`) then it would help the project to send us an issue attaching the file to your message along with a description of the system you ran the program on.

To Do

We currently need help with:

1. Thorough descriptions of each embroidery format.
2. Finding resources for each of the branded thread libraries (along with a full citation for documentation).
3. Finding resources for each geometric algorithm used (along with a full citation for documentation).
4. Completing the full `--full-test-suite` with no segfaults and at least a clear error message (for example "not implimented yet").
5. Identifying "best guesses" for filling in missing information when going from, say `.csv` to a late `.pes` version. What should the default be when the data doesn't clarify?
6. Improving the written documentation.
7. Funding, see the Sponsor button above. We can treat this as "work" and put far more hours in with broad support in small donations from people who want specific features.

Beyond this the development targets are categories sorted into:

1. Basic Features
2. Code quality and user friendliness
3. embroider CLI
4. Documentation
5. GUI
6. electronics development

Basic features.

1. Incorporate `#if 0`ed parts of `libembroidery.c`.
2. Interpret how to write formats that have a read mode from the source code and vice versa.
3. Document the specifics of the file formats here for embroidery machine specific formats. Find websites and other sources that break down the binary formats we currently don't understand.

4. Find more and better documentation of the structure of the headers for the formats we do understand.

Code quality and user friendliness

1. Document all structs, macros and functions (will contribute directly on the web version).
2. Incorporate experimental code, improve support for language bindings.
3. Make stitch x, y into an EmbVector.

embroider CLI

1. Make -circle flag to add a circle to the current pattern.
2. Make -rect flag to add a rectangle to the current pattern.
3. Make -fill flag to set the current satin fill algorithm for the current geometry. (for example "-fill crosses -circle 11,13,10'' fills a circle with center 11mm, 13mm with radius 10mm with crosses).
4. Make -ellipse flag to add to ellipse to the current pattern.
5. Make -bezier flag to add a bezier curve to the current pattern.

Embroider pipeline Adjectives apply to every following noun so

```
embroider --satin 0.3,0.6 --thickness 2 --circle 10,20,5 \
    --border 3 --disc 30,40,10 --arc 30,50,10,60 output.pes
```

Creates:

1. a circle with properties: thickness 2, satin 0.3,0.6
2. a disc with properties:
3. an arc with properties:

in that order then writes them to the output file `output.pes`.

Documentation

1. Create csv data files for thread tables.
2. Convert tex to markdown, make tex an output of `build.bash`.
3. Run `sloccount` on `extern/` and `.` (and) so we know the current scale of the project, aim to get this number low. Report the total as part of the documentation.
4. Try to get as much of the source code that we maintain into C as possible so new developers don't need to learn multiple languages to have an effect. This bars the embedded parts of the code.

GUI

1. Make MobileViewer also backend to `libembroidery` with a Java wrapper.
2. Make iMobileViewer also backend to `libembroidery` with a Swift wrapper.

3. Share some of the MobileViewer and iMobileViewer layout with the main EM2. Perhaps combine those 3 into the Embroidermodder repository so there are 4 repositories total.
4. Convert layout data to JSON format and use cJSON for parsing.

Electronics development

1. Currently experimenting with Fritzingˆ(*8*), upload netlists to embroiderbot when they can run simulations using the asm in `libembroidery`.
2. Create a common assembly for data that is the same across chipsets `libembrodiery_data_internal.s`.
3. Make the defines part of `embroidery.h` all systems and the function list "c code only". That way we can share some development between assembly and C versions.

Bibliography

1. Rudolf *Technical Info* http://www.achatina.de/sewing/main/TECHNICL .HTM (Accessed 25 November 2021)
2. [**?**]
3. Edutech format description[**?**],
4. KDE Liberty Description\cite{libertyTajima
5. The Sewing Witch *PCD2FMT* http://www.sewingwitch.com/sewing/bin /pcd2fmt.html (Accessed November 2021)
6. http://steve-jansen.github.io/guides/windows-batch-scripting/part-7-functions.html
7. https://stackoverflow.com/questions/16286134/imagemagick-how-can-i-work-with-histogram-result
8. Fritzing https://github.com/fritzing/fritzing-app
9. Sahoo, P., Wilkins, C., and Yeager, J., "Threshold selection using Renyi's entropy", *Pattern Recognition*, vol. 30, no. 1, pp. 71–84, 1997. doi:10.1016/S0031-3203(96)00065-9.
10. http://www.fmwconcepts.com/imagemagick/sahoothresh/index.php (Accessed 12 Dec 2021)
11. FINDME

Appendix

GNU Free Documentation License

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 https://www.gnu.org/licenses/.

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.