Embroidermodder 2.0.0-alpha

The Embroidermodder Team

February 13, 2023

Contents

The Embroidermodder Team

The Embroidermodder Team consists of:

- **Jonathan Greig** `redteam316` https://github.com/redteam316 Core Developer, Artwork, Documentation, Designs, Commands
- **Josh Varga** `JoshVarga` https://github.com/JoshVarga Core Developer
- **Jens Diemer** `jedie` https://github.com/jedie Documentation
- **Kim Howard** `turbokim` https://github.com/turbokim BugFixes
- **Martin Schneider** `craftoid` https://github.com/craftoid Documentation
- **Edward Greig** `Metallicow` https://github.com/Metallicow Artwork, BugFixes, Commands *"It is a sin to wear the band's shirt on concert night, Unless you buy it @t the show."*
- **Sonia Entzinger** Translation
- **SushiTee** `SushiTee` https://github.com/SushiTee BugFixes
- **Vathonie Lufh** `x2nie` https://github.com/x2nie BugFixes, Bindings
- **Nina Paley** Designs
- **Theodore Gray** Designs
- **Jens-Wolfhard Schicke-Uffmann** `Drahflow` BugFixes *Emmett Lauren Garlitz - Some Little Sandy Rd, Elkview, West by GOD Virginia* ['Oll Em'] *I have a nice cherry chess-top(Glass). "But remember, I NEVER played on it."*
- **Robin Swift** `robin-swift` https://github.com/robin-swift Core Developer, Documentation

The up-to date version of this list is maintained as part of the source code in the file `CREDITS.md`.

Chapter 1

Introduction

(UNDER MAJOR RESTRUCTURING, PLEASE WAIT FOR VERSION 2)
http://www.libembroidery.org
Embroidermodder is a free machine embroidery application. The newest version, Embroidermodder 2 can:

- edit and create embroidery designs
- estimate the amount of thread and machine time needed to stitch a design
- convert embroidery files to a variety of formats
- upscale or downscale designs
- run on Windows, Mac and Linux

Embroidermodder 2 is very much a work in progress since we're doing a ground up rewrite to an interface in C using the GUI toolkit SDL2. The reasoning for this is detailed in the issues tab.

For a more in-depth look at what we are developing read our [website]https://www.libembroidery.org which includes these docs as well as the up-to date printer-friendly versions. These discuss recent changes, plans and has user and developer guides for all the Embroidermodder projects.

To see what we're focussing on right now, see the [Open Collective News]https://opencollective.com/embroidermodder.

The current printer-friendly version of the manual: https://www.libembroidery.org/embroidermodder_2.0.0-alpha_manual.pdf.

1.1 License

The source code is under the terms of the zlib license: see `LICENSE.md` in the source code directory.

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".

1.2 The Embroidermodder Project and Team

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.

The tools and associated documents are:

- This manual.
- The website (['www.libembroidery.org']https://www.libembroidery.org)), which is maintained [here]https://github.com/Embroidermodder/embroidermodder.github.io.
- Mobile embroidery format viewers and tools ('EmbroideryMobile').
- The core library of functions ('libembroidery') and its manual.

- The Python version of the library of functions ('libembroidery-python') which is part of [libembroidery]https://github.com/Embroidermodder/libembroidery.
- The CLI ('embroider') which is part of [libembroidery]https://github.com/Embroidermodder/libembroidery.
- Specs for an open hardware embroidery machine called Embroiderbot (not started yet) which is part of [libembroidery]https://github.com/Embroidermodder/libembroidery and its manual.
- The GUI ('embroidermodder'), this repository.

They 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](https://opencollective.com/embroid 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 is the collection of people who've submitted patches, artwork and documentation to our three projects. The team was established by Jonathan Greig and Josh Varga. The full list is actively maintained below.

1.3 Credits for Embroidermodder 2, libembroidery and all other related code

If you have contributed and wish to be added to this list, alter the [README on Embroidermodder github page]https://github.com/Embroidermodder/Embroidermodder and we'll copy it to the libembroidery source code since that is credited to "The Embroidermodder Team".

1.4 Embroidermodder 1

The Embroidermodder Team is also inspired by the original Embroidermodder that was built by Mark Pontius and the same Josh Varga on SourceForge which unfortunately appears to have died from linkrot. We may create a distribution on here to be the official "legacy" Embroidermodder code but likely in a seperate repository because it's GNU GPL v3 and this code is written to be zlib (that is, permissive licensed) all the way down.

One reason why this is useful is that the rewrite by Jonathan Greig, John Varga and Robin Swift for Embroidermodder 2 should have no regressions: no features present in v1 should be missing in v2.

Chapter 2

Features

Embroidermodder 2 has many advanced features that enable you to create awesome designs quicker, tweak existing designs to perfection, and can be fully customized to fit your workflow.

A summary of these features:

- Cross Platform
- Realistic rendering
- Various grid types and auto-adjusting rulers
- Many measurement tools
- Add text to any design
- Supports many formats
- Batch Conversion
- Scripting API

2.1 Cross Platform

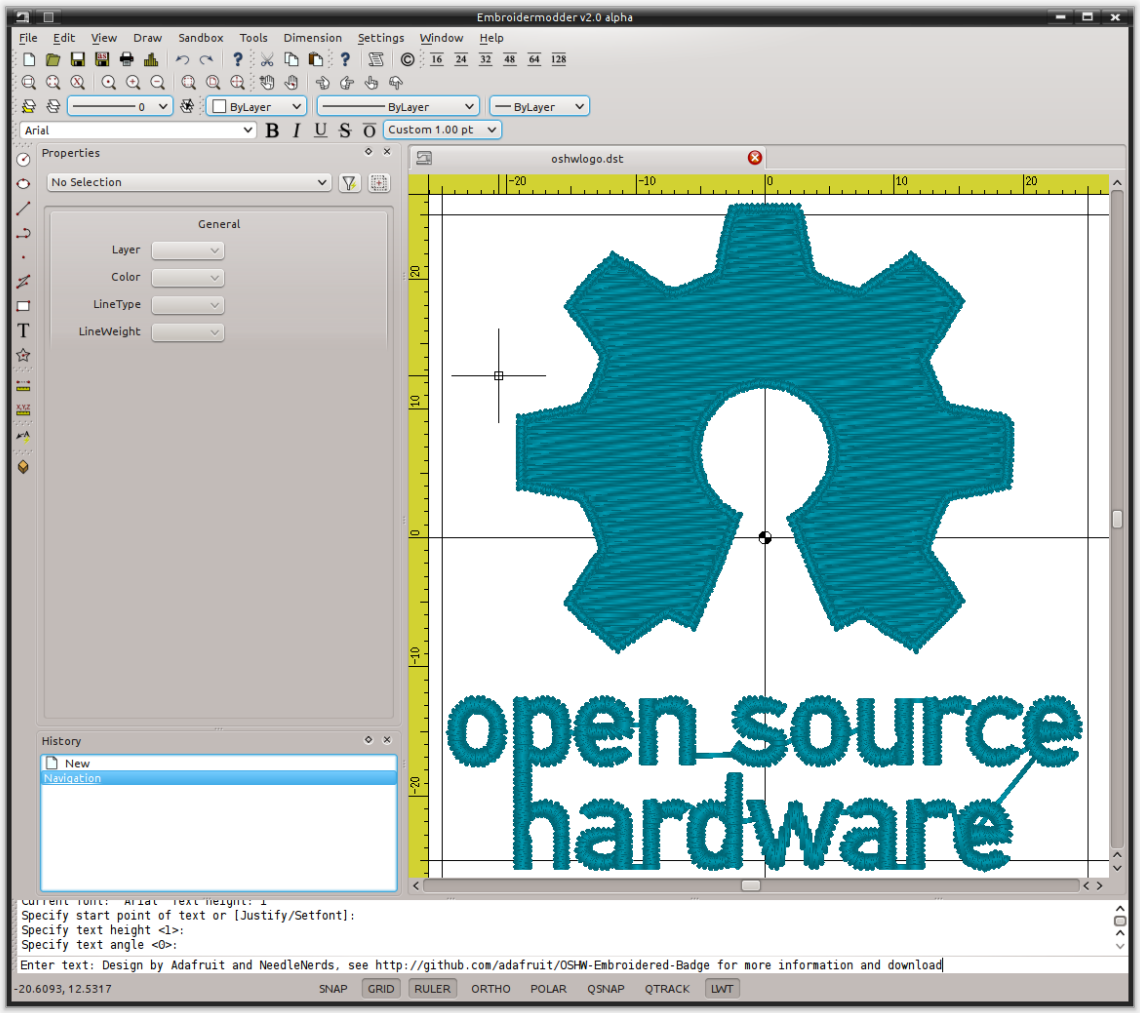
If you use multiple operating systems, it's important to choose software that works on all of them.

Embroidermodder 2 runs on Windows, Linux and Mac OS X. Let's not forget the Raspberry Pi (http://www.raspberrypi.org).

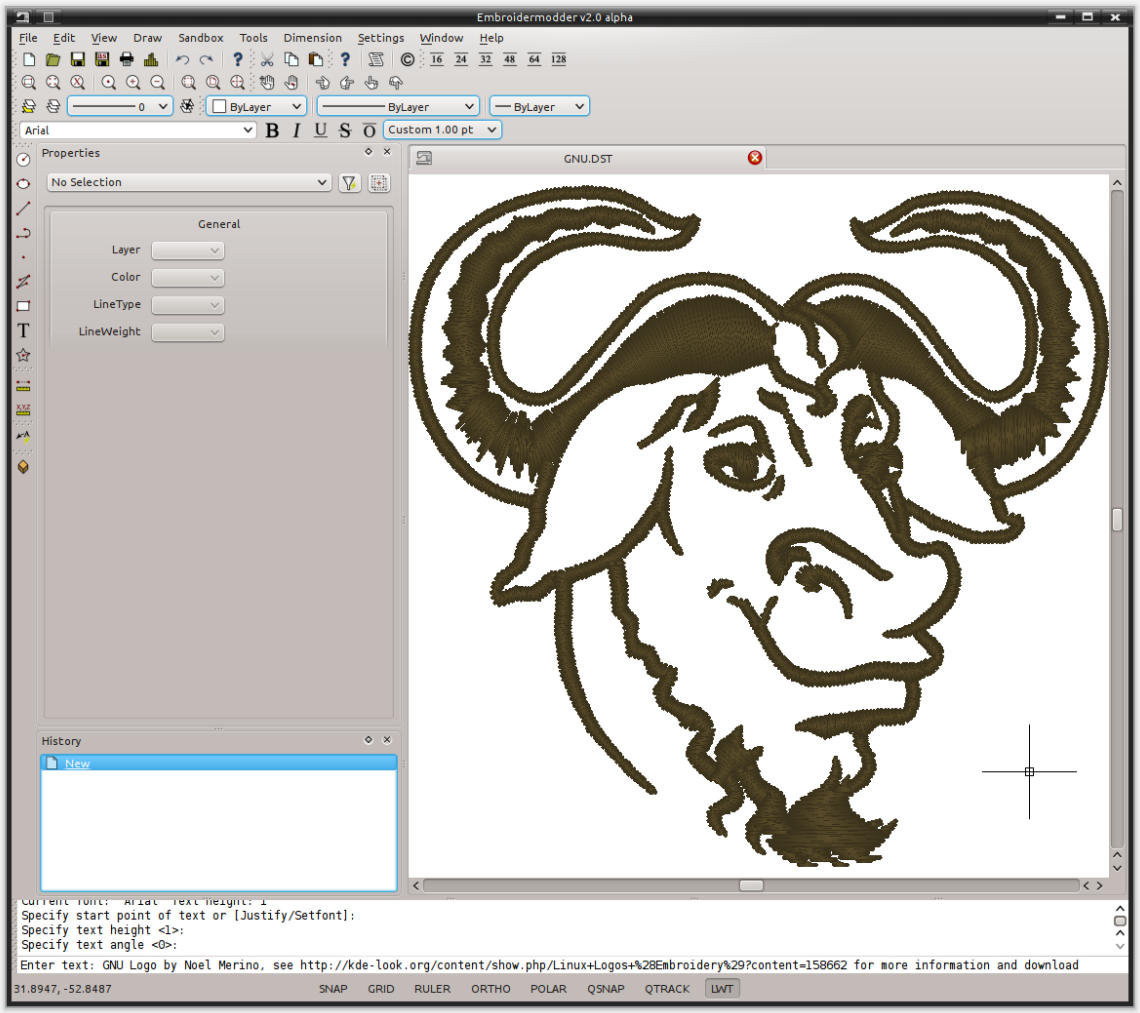
2.2 Realistic Rendering

It is important to be able to visualize what a design will look like when stitched and our pseudo "3D" realistic rendering helps achieve this.

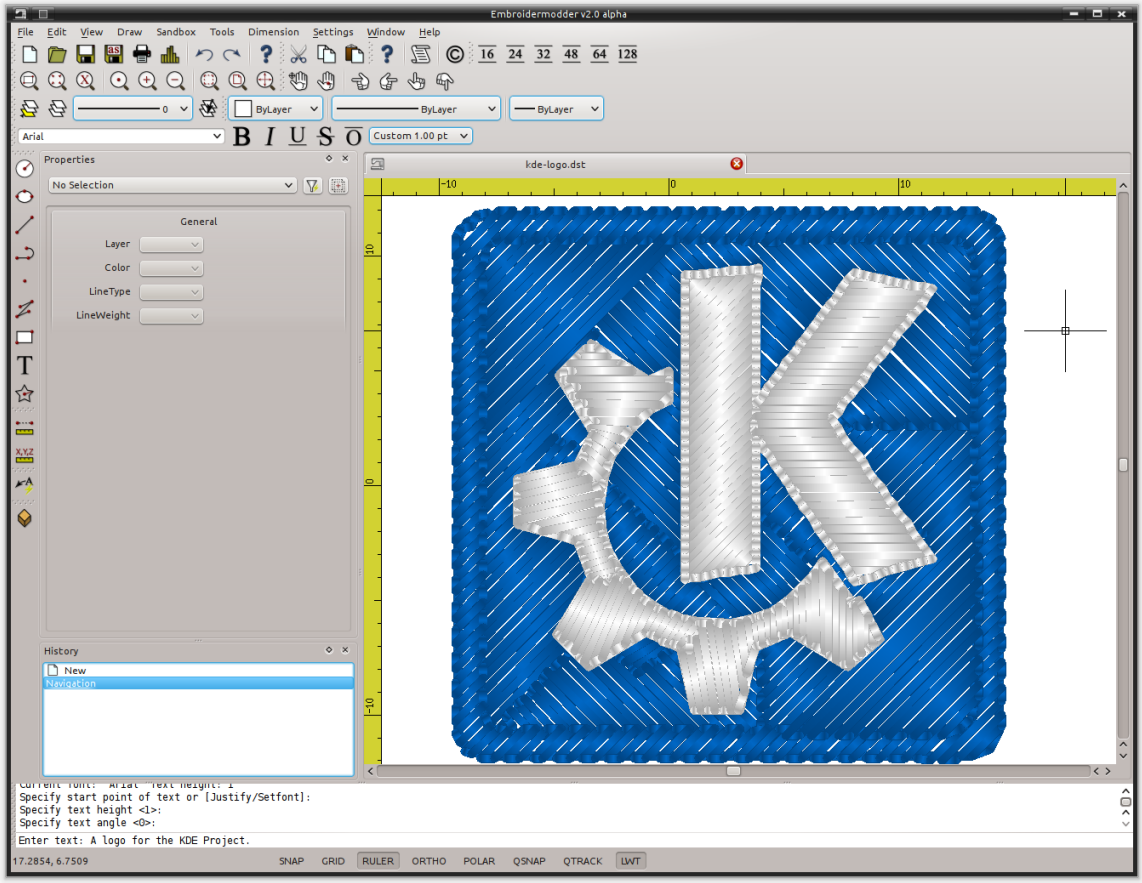
Realistic rendering sample #1:



Realistic rendering sample #2:



Realistic rendering sample #3:

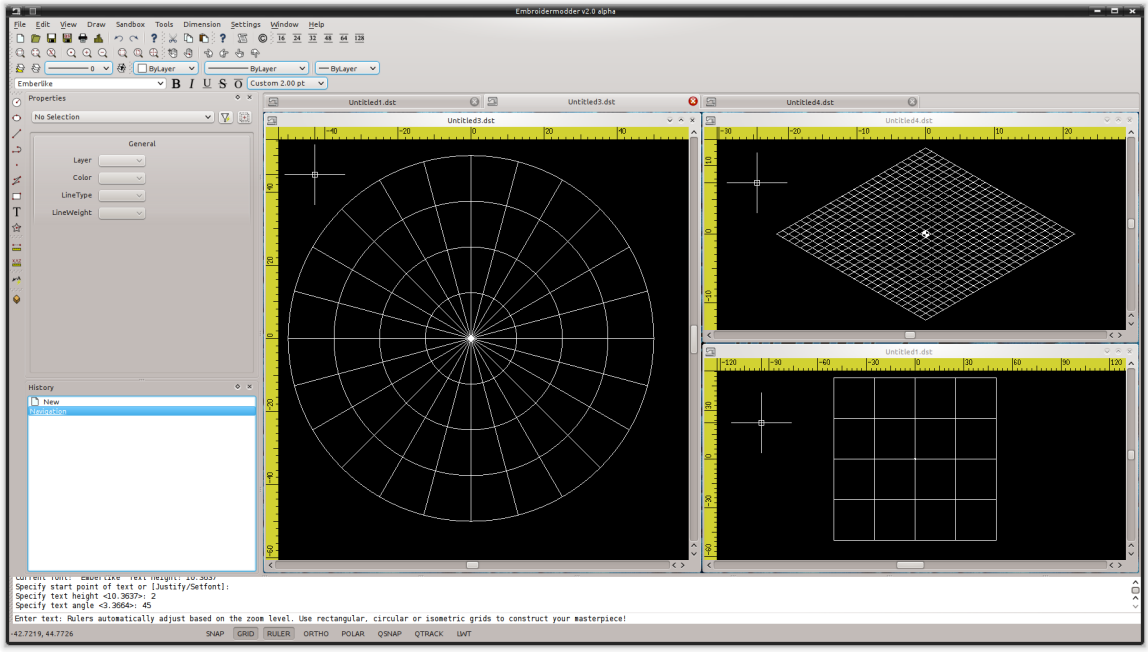


Various grid types and auto-adjusting rulers

Making use of the automatically adjusting ruler in conjunction with the grid will ensure your design is properly sized and fits within your embroidery hoop area.

Use rectangular, circular or isometric grids to construct your masterpiece!

Multiple grids and rulers in action:

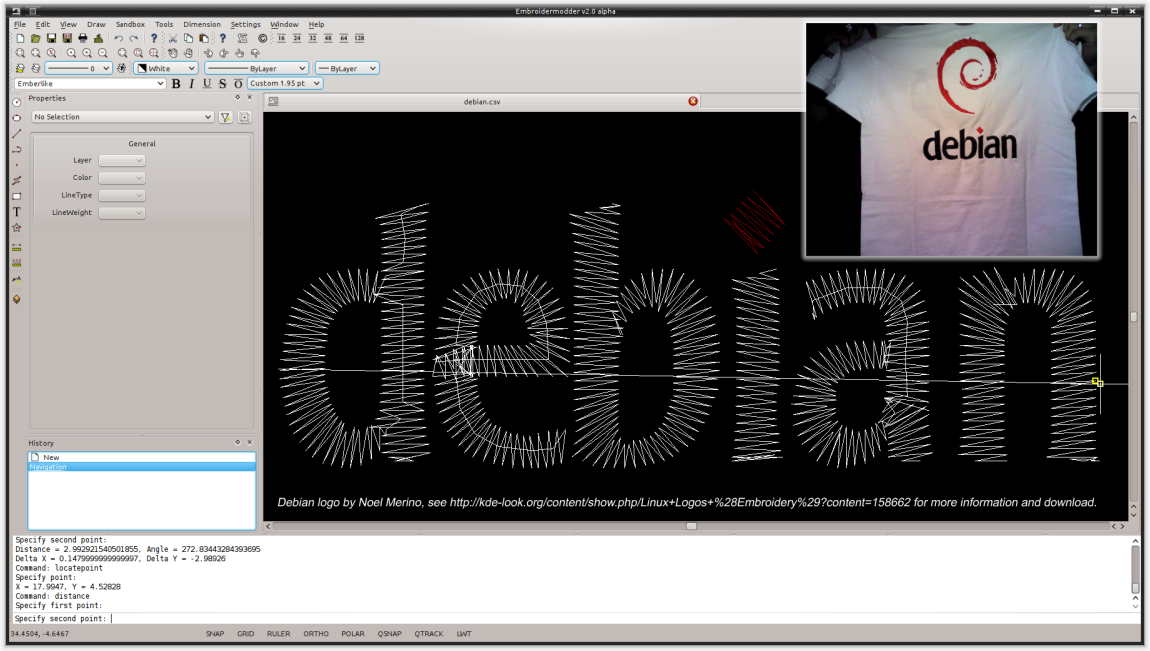


2.3 Many measurement tools

Taking measurements is a critical part of creating great designs. Whether you are designing mission critical embroidered space suits for NASA or some other far out design for your next meet-up, you will have precise

measurement tools at your command to make it happen. You can locate individual points or find distances between any 2 points anywhere in the design!

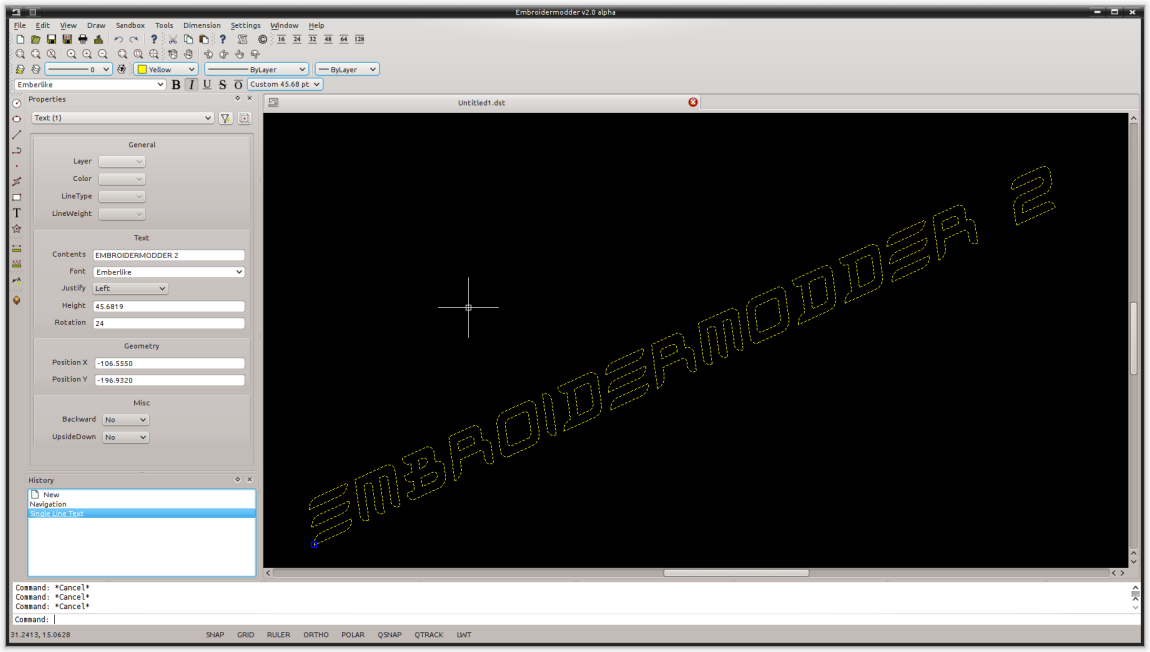
Take quick and accurate measurements:



2.4 Add text to any design

Need to make company apparel for all of your employees with individual names on them? No sweat. Just simply add text to your existing design or create one from scratch, quickly and easily. Didn't get it the right size or made a typo? No problem. Just select the text and update it with the property editor.

Add text and adjust its properties quickly:



2.5 Supports many formats

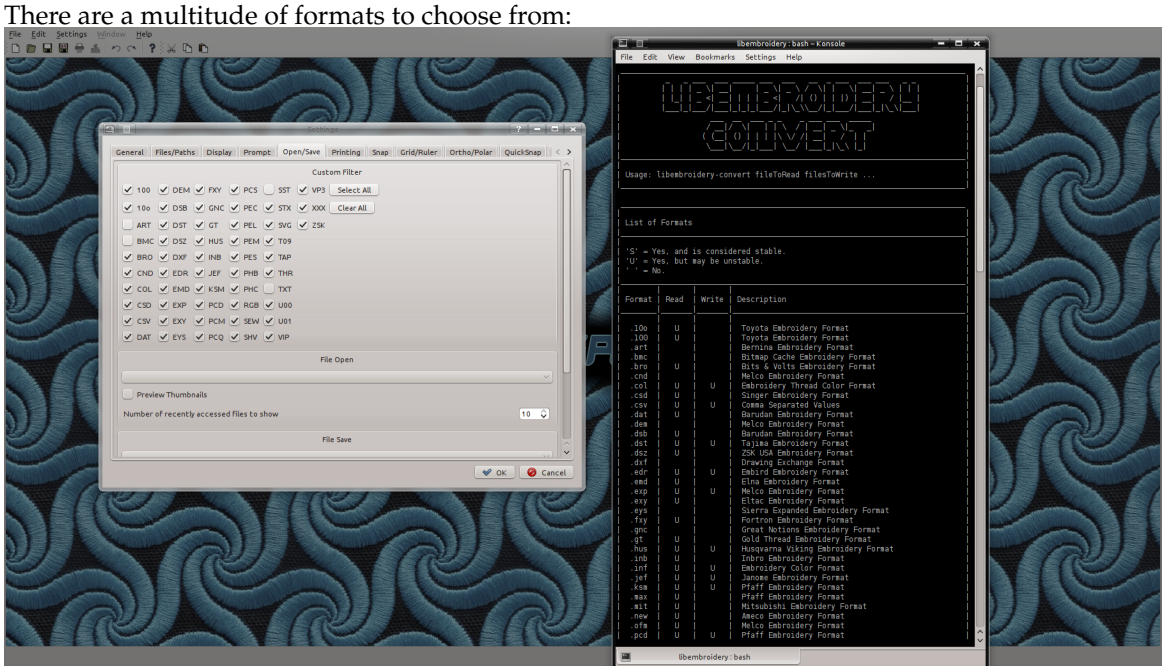
Embroidery machines all accept different formats. There are so many formats available that it can sometimes be confusing whether a design will work with your machine.

Embroidermodder 2 supports a wide variety of embroidery formats as well as several vector formats, such as SVG and DXF. This allows you to worry less about which designs you can use.

2.6 Batch Conversion

Need to send a client several different formats? Just use libembroidery-convert, our command line utility which supports batch file conversion.

There are a multitude of formats to choose from:



2.7 Scripting API

If you've got programming skills and there is a feature that isn't currently available that you absolutely cannot live without, you have the capability to create your own custom commands for Embroidermodder 2. We provide an QtScript API which exposes various application functionality so that it is possible to extend the application without requiring a new release. If you have created a command that you think is worth including in the next release, just ¡a href="contact.html"¿contact us¡/a¿ and we will review it for functionality, bugs, and finally inclusion.

An Embroidermodder 2 command excerpt:

```
var strList = str.split(",");
if(isNaN(strList[0]) || isNaN(strList[1]))
{
    setPromptPrefix("Point or option keyword required.");
    appendPromptHistory();
    setPromptPrefix("Specify second axis end point or [Rotation]: ");
}
else
{
    global.x3 = Number(strList[0]);
    global.y3 = Number(strList[1]);
    global.height = perpendicularDistance(global.x3, global.y3, global.x1, global.y1, global.x2, global.y2)*2.0;
    setRubberPoint("ELLIPSE_AXIS2_POINT2", global.x3, global.y3);
    vulcanize();
    endCommand();
}
```

2.8 Build and Install

Assuming you already have the SDL2 libraries you can proceed to using the fast build, which assumes you want to build and test locally.

The fast build should be:

```
bash build.sh
```

or, on Windows:

```
.\build.bat
```

Then run using the 'run.bat' or 'run.sh' scripts in the build/ directory.

Otherwise, follow the instructions below.

If you plan to install the dev version to your system (we recommend you wait for the official installers and beta release first) then use the CMake build instead.

2.8.1 Install on Desktop

We recommend that if you want to install the development version you use the CMake build. Like this:

```
git submodule init
git submodule update

mkdir build
cd build
cmake ..
cmake --build .
sudo cmake --install .
```

These lines are written into the file:

```
./build_install.sh
```

On Windows use the next section.

Chapter 3

Design

Embroidermodder 2 was written in C++/Qt5 and it was far too complex. We had issues with people not able to build from source because the Qt5 libraries were so ungainly. So I decided to do a rewrite in C/SDL2 (originally FreeGLUT, but that was a mistake) with data stored as YAML. This means linking 4-7 libraries depending on your system which are all well supported and widely available.

This is going well, although it's slow progress as I'm trying to keep track of the design while also doing a ground up rewrite. I don't want to throw away good ideas. Since I also write code for libembroidery my time is divided. Overview of the UI rewrite

(Problems to be solved in brackets.)

It's not much to look at because I'm trying to avoid using an external widgets system, which in turn means writing things like toolbars and menubars over. If you want to get the design the actuator is the heart of it.

Without Qt5 we need a way of assigning signals with actions, so this is what I've got: the user interacts with a UI element, this sends an integer to the actuator that does the thing using the current state of the mainwindow struct of which we expect there to be exactly one instance. The action is taken out by a jump table that calls the right function (most of which are missing in action and not connected up properly). It also logs the number, along with key parts of the main struct in the undo history (an unsolved problem because we need to decide how much data to copy over per action). This means undo, redo and repeat actions can refer to this data.

3.1 To Do

3.1.1 For 2.0.0-alpha1

- WIP - Statistics from 1.0, needs histogram
- WIP - Saving DST/PES/JEF (varga)
- WIP - Saving CSV/SVG (rt) + CSV read/write UNKNOWN interpreted as COLOR bug

3.1.2 For 2.0.0-alpha2

- TODO - Notify user of data loss if not saving to an object format.
- TODO - Import Raster Image
- TODO - SNAP/ORTHO/POLAR
- TODO - Layer Manager + LayerSwitcher DockWidget
- TODO - Reading DXF

3.1.3 For 2.0.0-alpha3

- TODO - Writing DXF
- DONE - Up and Down keys cycle thru commands in the command prompt
- TODO - Amount of Thread & Machine Time Estimation (also allow customizable times for setup, color changes, manually trimming jump threads, etc...that way a realistic total time can be estimated)

- TODO - Otto Theme Icons - whatsthis icon doesn't scale well, needs redone
- TODO - embroidermodder2.ico 16 x 16 looks horrible

3.1.4 For 2.0.0-alpha4

- WIP - CAD Command: Arc (rt)
- TODO - automate changelog and write to a javascript file for the docs: git log –pretty=tformat:'¡a href="https://github.com/Embroidermodder/Embroidermodder/commit/%H"¿

3.1.5 For 2.0.0-beta1

- TODO - Custom Filter Bug - doesn't save changes in some cases
- TODO - Cannot open file with # in name when opening multiple files (works fine when opening the single file)
- TODO - Closing Settings Dialog with the X in the window saves settings rather than discards them
- WIP - Advanced Printing
- TODO - Filling Algorithms (varga)
- TODO - Otto Theme Icons - beta (rt) - Units, Render, Selectors

3.1.6 For 2.0.0-rc1

- TODO - QDoc Comments
- TODO - Review KDE4 Thumbnailer
- TODO - Documentation for libembroidery & formats
- TODO - HTML Help files
- TODO - Update language translations
- TODO - CAD Command review: line
- TODO - CAD Command review: circle
- TODO - CAD Command review: rectangle
- TODO - CAD Command review: polygon
- TODO - CAD Command review: polyline
- TODO - CAD Command review: point
- TODO - CAD Command review: ellipse
- TODO - CAD Command review: arc
- TODO - CAD Command review: distance
- TODO - CAD Command review: locatepoint
- TODO - CAD Command review: move
- TODO - CAD Command review: rgb
- TODO - CAD Command review: rotate
- TODO - CAD Command review: scale
- TODO - CAD Command review: singlelinetext
- TODO - CAD Command review: star
- TODO - Clean up all compiler warning messages, right now theres plenty :P

3.1.7 For 2.0 release

- TODO - tar.gz archive
- TODO - zip archive
- TODO - Debian Package (rt)
- TODO - NSIS Installer (rt)
- TODO - Mac Bundle?
- TODO - press release

3.1.8 For 2.x/Ideas

- TODO - libembroidery.mk for MXE project (refer to qt submodule packages for qmake based building. Also refer to plibc.mk for example of how write an update macro for github.)
- TODO - libembroidery safeguard for all writers - check if the last stitch is an END stitch. If not, add an end stitch in the writer and modify the header data if necessary.
- TODO - Cut/Copy - Allow Post-selection
- TODO - CAD Command: Array
- TODO - CAD Command: Offset
- TODO - CAD Command: Extend
- TODO - CAD Command: Trim
- TODO - CAD Command: BreakAtPoint
- TODO - CAD Command: Break2Points
- TODO - CAD Command: Fillet
- TODO - CAD Command: Chamfer
- TODO - CAD Command: Split
- TODO - CAD Command: Area
- TODO - CAD Command: Time
- TODO - CAD Command: PickAdd
- TODO - CAD Command: Product
- TODO - CAD Command: Program
- TODO - CAD Command: ZoomFactor
- TODO - CAD Command: GripHot
- TODO - CAD Command: GripColor & GripCool
- TODO - CAD Command: GripSize
- TODO - CAD Command: Highlight
- TODO - CAD Command: Units
- TODO - CAD Command: Grid
- TODO - CAD Command: Find
- TODO - CAD Command: Divide
- TODO - CAD Command: ZoomWindow (Move out of view.cpp)
- TODO - Command: Web (Generates Spiderweb patterns)

- TODO - Command: Guilloche (Generates Guilloche patterns)
- TODO - Command: Celtic Knots
- TODO - Command: Knotted Wreath
- TODO - Lego Mindstorms NXT/EV3 ports and/or commands.
- TODO - native function that flashes the command prompt to get users attention when using the prompt is required for a command.
- TODO - libembroidery-composer like app that combines multiple files into one.
- TODO - Settings Dialog, it would be nice to have it notify you when switching tabs that a setting has been changed. Adding an Apply button is what would make sense for this to happen.
- TODO - Keyboard Zooming/Panning
- TODO - G-Code format?
- TODO - 3D Raised Embroidery
- TODO - Gradient Filling Algorithms
- TODO - Stitching Simulation
- TODO - RPM packages?
- TODO - Reports?
- TODO - Record and Playback Commands
- TODO - Settings option for reversing zoom scrolling direction
- TODO - Qt GUI for libembroidery-convert
- TODO - EPS format? Look at using Ghostscript as an optional add-on to libembroidery...
- TODO - optional compile option for including LGPL/GPL libs etc... with warning to user about license requirements.
- TODO - Realistic Visualization - Bump Mapping/OpenGL/Gradients?
- TODO - Stippling Fill
- TODO - User Designed Custom Fill
- TODO - Honeycomb Fill
- TODO - Hilburt Curve Fill
- TODO - Sierpinski Triangle fill
- TODO - Circle Grid Fill
- TODO - Spiral Fill
- TODO - Offset Fill
- TODO - Brick Fill
- TODO - Trim jumps over a certain length.
- TODO - FAQ about setting high number of jumps for more controlled trimming.
- TODO - Minimum stitch length option. (Many machines also have this option too)
- TODO - Add 'Design Details' functionality to libembroidery-convert
- TODO - Add 'Batch convert many to one format' functionality to libembroidery-convert
- TODO - EmbroideryFLOSS - Color picker that displays catalog numbers and names.

3.2 Problems to be fixed before the Beta Release

1. Realistic Visualization - Bump Mapping/OpenGL/Gradients?
2. Get undo history widget back (BUG).
3. Mac Bundle, .tar.gz and .zip source archive.
4. NSIS installer for Windows, Debian package, RPM package
5. GUI frontend for embroider features that aren't supported by embroidermodder: flag selector from a table
6. Update all formats without color to check for edr or rgb files.
7. Setting for reverse scrolling direction (for zoom, vertical pan)
8. Keyboard zooming, panning
9. New embroidermodder2.ico 16x16 logo that looks good at that scale.
10. Saving dst, pes, jef.
11. Settings dialog: notify when the user is switching tabs that the setting has been changed, adding apply button is what would make sense for this to happen.
12. Update language translations.
13. Replace KDE4 thumbnailer.
14. Import raster image.
15. Statistics from 1.0, needs histogram.
16. SNAP/ORTHO/POLAR.
17. Cut/copy allow post-selection.
18. Layout into config.
19. Notify user of data loss if not saving to an object format.
20. Add which formats to work with to preferences.
21. Cannot open file with # in the name when opening multiple files but works with opening a single file.
22. Closing settings dialog with the X in the window saves settings rather than discarding them.
23. Otto theme icons: units, render, selectors, what's this icon doesn't scale.
24. Layer manager and Layer switcher dock widget.
25. Test that all formats read data in correct scale (format details should match other programs).
26. Custom filter bug – doesn't save changes in some cases.
27. Tools to find common problems in the source code and suggest fixes to the developers. For example, a translation miss: that is, for any language other than English a missing entry in the translation table should supply a clear warning to developers.
28. Converting Qt C++ version to native GUI C throughout.
29. OpenGL Rendering: "Real" rendering to see what the embroidery looks like, Icons and toolbars, Menu bar.
30. Libembroidery interfacing: get all classes to use the proper libembroidery types within them. So 'Ellipse' has 'EmbEllipse' as public data within it.
31. Move calculations of rotation and scaling into 'EmbVector' calls.
32. GUI frontend for embroider features that aren't supported by embroidermodder: flag selector from a table

33. Update all formats without color to check for edr or rgb files.
34. Setting for reverse scrolling direction (for zoom, vertical pan)
35. Keyboard zooming, panning
36. Better integrated help: I don't think the help should backend to a html file somewhere on the user's system. A better system would be a custom widget within the program that's searchable.
37. New embroidermodder2.ico 16x16 logo that looks good at that scale.
38. Settings dialog: notify when the user is switching tabs that the setting has been changed, adding apply button is what would make sense for this to happen.

3.3 Contributing

3.4 Version Control

Being an open source project, developers can grab the latest code at any time and attempt to build it themselves. We try our best to ensure that it will build smoothly at any time, although occasionally we do break the build. In these instances, please provide a patch, pull request which fixes the issue or open an issue and notify us of the problem, as we may not be aware of it and we can build fine.

Try to group commits based on what they are related to: features/bugs/comments/graphics/commands/etc...

See the coding style [here](coding-style)

3.4.1 Get the Development Build going

When we switch to releases we recommend using them, unless you're reporting a bug in which case you can check the development build for whether it has been patched. If this applies to you, the current development build is:

Linux (https://github.com/Embroidermodder/Embroidermodder/suites/8882922866/artifacts/406005099)

Mac OS (https://github.com/Embroidermodder/Embroidermodder/suites/8882922866/artifacts/406005101)

Windows (https://github.com/Embroidermodder/Embroidermodder/suites/8882922866/artifacts/406005102)

3.5 Problems to be fixed during Beta and before 2.0.0

1. Libembroidery 1.0.
2. Better integrated help: I don't think the help should backend to a html file somewhere on the user's system. A better system would be a custom widget within the program that's searchable.
3. EmbroideryFLOSS - Color picker that displays catalog numbers and names.
4. Custom filter bug – doesn't save changes in some cases.
5. Advanced printing.
6. Stitching simulation.

3.6 Problems to be fixed eventually

1. User designed custom fill.

Chapter 4

Libembroidery

4.1 To Do

4.1.1 For Arduino

- TODO - Fix emb-outline files
- TODO - Fix thread-color files
- TODO - Logging of Last Stitch Location to External USB Storage(commonly available and easily replaced) ...wait until TRE is available to avoid rework
- TODO - inotool.org - seems like the logical solution for Nightly/CI builds
- TODO - Smoothieboard experiments

4.1.2 Testing

- TODO - looping test that reads 10 times while running valgrind. See `embPattern_loadExternalColorFile()` Arduino leak note for more info.

4.2 DXF File Format

AutoDesk [2012]

4.3 Development

If you wish to develop with us you can chat via the contact email on the [website]https://libembroidery.org or in the issues tab on the [github page]https://github.com/Embroidermodder/Embroidermodder/issues. People have been polite and friendly in these conversations and I (Robin) have really enjoyed them. If we do have any arguments please note we have a [Code of Conduct] CODE_OF_CONDUCT.md so there is a consistent policy to enforce when dealing with these arguments.

The first thing you should try is building from source using the [build advice](build) above. Then read some of the [manual] https://libembroidery.org/embroidermodder_2.0_manual.pdf to get the general layout of the source code and what we are currently planning.

4.4 Testing

To find unfixed errors run the tests by launching from the command line with:

```
$ embroidermodder --test
```

then dig through the output. It's currently not worth reporting the errors, since there are so many but if you can fix anything reported here you can submit a PR.

4.5 Contributing

4.5.1 Funding

The easiest way to help is to fund development (see the Donate button above), since we can't afford to spend a lot of time developing and only have limited kit to test out libembroidery on.

4.5.2 Programming and Engineering

Should you want to get into the code itself:

- Low level C developers are be needed for the base library 'libembroidery'.
- Low level assembly programmers are needed for translating some of 'libembroidery' to 'EmbroiderBot'.
- Hardware Engineers to help design our own kitbashed embroidery machine 'EmbroiderBot', one of the original project aims in 2013.
- Scheme developers and C/SDL developers to help build the GUI.
- Scheme developers to help add designs for generating of custom stitch-filled emblems like the heart or dolphi. Note that this happens in Embroidermodder not libembroidery (which assumes that you already have a function available).

4.5.3 Writing

We also need people familiar with the software and the general machine embroidery ecosystem to contribute to the [documentation](https://github.com/Embroidermodder/docs).

We need researchers to find references for the documentation: colour tables, machine specifications etc. The history is murky and often very poorly maintained so if you know anything from working in the industry that you can share: it'd be appreciated!

4.6 Design

These are key bits of reasoning behind why the software is built the way it is.

4.7 Translation of the user interface

In a given table the left column is the default symbol and the right string is the translation. If the translate function fails to find a translation it returns the default symbol.

So in US English it is an empty table, but in UK English only the dialectical differences are present.

Ideally, we should support at least the 6 languages spoken at the UN. Quoting www.un.org:

> *There are six official languages of the UN. These are Arabic, Chinese, English, French, Russian and Spanish.*

We're adding Hindi, on the grounds that it is one of the most commonly spoken languages and at least one of the Indian languages should be present.

Written Chinese is generally supported as two different symbol sets and we follow that convension.

English is supported as two dialects to ensure that the development team is aware of what those differences are. The code base is written by a mixture of US and UK native English speakers meaning that only the variable names are consistently one dialect: US English. As for documentation: it is whatever dialect the writer prefers (but they should maintain consistency within a text block like this one).

Finally, we have "default", which is the dominant language of the internals of the software. Practically, this is just US English, but in terms of programming history this is the "C locale".

4.8 Old action system notes

NO LONGER HOW ACTION SYSTEM WORKS, MOVE TO DOCS.

Action: the basic system to encode all user input.

This typedef gives structure to the data associated with each action which, in the code, is referred to by the action id (an int from the define table above).

4.9 DESCRIPTION OF STRUCT CONTENTS

4.9.1 label

What is called from Scheme to run the function. It is always in US English, lowercase, seperated with hyphens.
For example: new-file.

4.9.2 function

The function pointer, always starts with the prefix scm, in US English, lowercase, seperated with underscores.
The words should match those of the label otherwise.
For example: scm_new_file.

4.9.3 flags

The bit based flags all collected into a 32-bit integer.

bit(s)	description
0	User (0) or system (1) permissions.
1-3	The mode of input. — — 4-8 — The object classes that this action — — — can be applied to. — — 9-10 — What m

4.9.4 description

The string placed in the tooltip describing the action.

Chapter 5

Tutorials

5.1 Basic Features

5.1.1 Move a single stitch in an existing pattern

1. In the 'File' menu, click 'Open...'. When the open dialog appears find and select your file by double clicking the name of the file. Alternatively, left click the file once then click the 'Open' button.
2.
3. In the 'File' menu

TIP: For users who prefer

5.1.2 Convert one pattern to another format

1. In the 'File' menu, click 'Open...'.
2. The
3. In the dropdown menu within the save dialog select the

5.2 Advanced Features

5.3 Format Support

FORMAT	READ	WRITE	NOTES
10o	YES		read (need to fix external color loading) (maybe find out what ctrl — code flags of 0x10, 0x08, 0x04, and 0x02 mean)
100			none (4 byte codes) 61 00 10 09 (type, type2, x, y ?) x — y (signed char)
art —		none	
bro	YES		read (complete)(maybe figure out detail of header)
cnd			none
col			(color file no design) read(final) write(final)
csd — YES		read (complete)	
dat			read ()
'dem'			none (looks like just encrypted cnd)
dsb	YES		read (unknown how well) (stitch data looks same as 10o)
dst	YES		read (complete) / write(unknown)
dsz	YES		read (unknown)
dxf			read (Port to C. needs refactored)
edr			read (C version is broken) / write (complete)
emd			read (unknown)

exp	YES		read (unknown) / write(unknown)
exy	YES		read (need to fix external color loading)
fxy	YES		read (need to fix external color loading)
gnc			none
gt			read (need to fix external color loading)
hus	YES		read (unknown) / write (C version is broken)
inb	YES		read (buggy?)
jef	YES		write (need to fix the offsets when it is moving to another spot)
ksm	YES		read (unknown) / write (unknown)
pcd			
pcm			
pcq			read (Port to C)
pcs	BUGGY		read (buggy / colors are not correct / after reading, writing any other format is messed up)
pec			read / write (without embedded images, sometimes overlooks some stitches leaving a gap)
pel			none
pem			none
pes	YES		
phb			
phc			
'rgb'			
'sew'	YES		
'shv'			read (C version is broken)
'sst'			none
'svg'		YES	
'tap'	YES		read (unknown)
'u01'			
'vip'	YES		
'vp3'	YES		
'xxx'	YES		
'zsk'			read (complete)

Support for Singer FHE, CHE (Compucon) formats?

5.4 Embroidermodder Project Coding Standards

A basic set of guidelines to use when submitting code.

Code structure is mre important than style, so first we advise you read "Design" and experimenting before getting into the specifics of code style.

5.4.1 Where Code Goes

Anything that deals with the specifics of embroidery file formats, threads, rendering to images, embroidery machinery or command line interfaces should go in 'libembroidery' not here.

Should your idea pass this test:

1. A new kind of GUI structure it goes in 'src/ui.c'.
2. If it's something the user can do, make a section of the 'actuator' function (which lives in 'src/actuator.c') using the guide "The Actuator's Behaviour".
3. Potentially variable data that is global goes in 'src/data.c'.
4. If the data will not vary declare it as a compiler definition using the "Compiler definitions" section and put it in 'src/em2.h'.
5. All other C code goes in 'src/em2.c'.

5.4.2 Where Non-compiled Files Go

TODO: Like most user interfaces Embroidermodder is mostly data, so here we will have a list describing where each CSV goes.

5.4.3 Ways in which we break style on purpose

Most style guides advise you to keep functions short. We make a few pointed exceptions to this where the overall health and functionality of the source code should benefit.

The 'actuator' function will always be a mess and it should be: we're keeping the total source lines of code down by encoding all user action into a descrete sequence of strings that are all below `_STRING_LENGTH` in length. See the section on the actuator (TODO) describing why any other solution we could think here would mean more more code without a payoff in speed of execution or clarity.

5.4.4 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 Functions that attempt to simulate namespacing, should be `nameSpace_camelCase`.

All files and directories shall be lowercase and contain no spaces.

5.5 Code Style

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

5.5.1 Braces

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

```
void function_definition(int argument)
{
    /* code block */
}
```

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

```
if (condition) {
    /* code block */
}
```

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.

5.6 Version Control

Being an open source project, developers can grab the latest code at any time and attempt to build it themselves. We try our best to ensure that it will build smoothly at any time, although occasionally we do break the build. In these instances, please provide a patch, pull request which fixes the issue or open an issue and notify us of the problem, as we may not be aware of it and we can build fine.

Try to group commits based on what they are related to: features/bugs/comments/graphics/commands/etc...

5.7 Donations

Creating software that interfaces with hardware is costly. A summary of some of the costs involved:

1. Developer time for 2 core developers
2. Computer equipment and parts

3. Embroidery machinery
4. Various electronics for kitbashing Embroiderbot
5. Consumable materials (thread, fabric, stabilizer, etc...)

If you have found our software useful, please consider funding further development by donating to the project on Open Collective (https://opencollective.com/embroidermodder).

Chapter 6

Introduction

(UNDER MAJOR RESTRUCTURING, PLEASE WAIT FOR VERSION 2)

Embroidermodder is a free machine embroidery application. The newest version, Embroidermodder 2 can:

- edit and create embroidery designs
- estimate the amount of thread and machine time needed to stitch a design
- convert embroidery files to a variety of formats
- upscale or downscale designs
- run on Windows, Mac and Linux

For more information, see our website **?**.

Embroidermodder 2 is very much a work in progress since we're doing a ground up rewrite to an interface in Python using the GUI toolkit Tk. The reasoning for this is detailed in the issues tab.

For a more in-depth look at what we are developing read the developer notes (link to dev notes section). This discusses recent changes in a less formal way than a changelog (since this software is in development) and covers what we are about to try.

Documentation

The documentation is in the form of the website (included in the 'docs/' directory) and the printed docs in this file.

6.0.1 Development

If you wish to develop with us you can chat via the contact email on the [website]https://www.libembroidery.org or in the issues tab on the [github page]https://github.com/Embroidermodder/Embroidermodder/issues. People have been polite and friendly in these conversations and I (Robin) have really enjoyed them. If we do have any arguments please note we have a [Code of Conduct](CODE_OF_CONDUCT.md) so there is a consistent policy to enforce when dealing with these arguments.

The first thing you should try is building from source using the [build advice](link to build) above. Then read some of the [development notes](link to dev notes.md) to get the general layout of the source code and what we are currently planning.

6.0.2 Testing

To find unfixed errors run the tests by launching from the command line with:

```
$ embroidermodder --test
```

then dig through the output. It's currently not worth reporting the errors, since there are so many but if you can fix anything reported here you can submit a PR.

6.1 Code Optimisations and Simplifications

6.1.1 Geometry

The geometry is stored, processed and altered via libembroidery. See the Python specific part of the documentation for libembroidery for this. What the code in Embroidermodder does is make the GUI widgets to change and view this information graphically.

For example if we create a circle with radius 10mm and center at (20mm, 30mm) then fill it with stitches the commands would be

```
from libembroidery import Pattern, Circle, Vector, satin
circle = Circle(Vector(20, 30), 10)
pattern = Pattern()
pattern.add_circle(circle, fill=satin)
pattern.to_stitches()
```

but the user would do this through a series of GUI actions:

1. Create new file
2. Click add circle
3. Use the Settings dialog to alter the radius and center
4. Use the fill tool on circle
5. Select satin from the drop down menu

So EM2 does the job of bridging that gap.

6.1.2 Postscript Support

In order to safely support user contributed/shared data that can define, for example, double to double functions we need a consistent processor for these descriptions.

Embroidermodder backends to the postscript interpreter included in libembroidery to accomplish this.

For example the string:

```
5 2 t mul add
```

is equivalent to the expression:

```
2*t + 5
```

The benefit of not allowing this to simply be a Python expression is that it is safe against malicious use, or accidental misuse. The program can identify whether the output is of the appropriate form and give finitely many calculations before declaring the function to have run too long (stopping equations that hang).

To see examples of this see the 'assets/shapes/*.ps' files.

6.1.3 SVG Icons

To make the images easier to alter and restyle we could switch to svg icons. There's some code in the git history to help with this.

6.1.4 The Actions System

In order to simplify the development of a GUI that is flexible and easy to understand to new developers we have a custom action system that all user actions will go via an 'actuator' that takes a string argument. By using a string argument the undo history is just an array of strings.

The C 'action_hash_data' struct will contain: the icon used, the labels for the menus and tooltips and the function pointer for that action. There will be an accompanying argument for this function call, currently being drafted as 'action_call'. So when the user makes a function call it should contain information like the mouse position, whether special key is pressed etc.

6.1.5 Accessibility

Software can be more or less friendly to people with dylexia, partial sightedness, reduced mobility and those who don't speak English. Embroidermodder 2 has, in its design, the following features to help:

- icons for everything to reduce the amount of reading required
- the system font is configurable: if you have a dyslexia-friendly font you can load it
- the interface rescales to help with partial-sightedness

- the system language is configurable, unfortunately the docs will only be in English but we can try to supply lots of images of the interface to make it easier to understand as a second language
- buttons are remappable: XBox controllers are known for being good for people with reduced mobility so remapping the buttons to whatever setup you have should help

Note that most of these features will be released with version 2.1, which is planned for around early 2023.

6.1.6 Sample Files

Various sample embroidery design files can be found in the embroidermodder2/samples folder.

6.1.7 Shortcuts

A shortcut can be made up of zero or more modifier keys and at least one non-modifier key pressed at once.

To make this list quickly assessable, we can produce a list of hashes which are simply the flags ORed together.

The shortcuts are stored in the csv file "shortcuts.csv" as a 5-column table with the first 4 columns describing the key combination. This is loaded into the shortcuts 'TABLE'. Each tick the program checks the input state for this combination by first translating the key names into indices for the key state, then checking for whether all of them are set to true.

6.1.8 CAD command review

ID	name	arguments	description
0	newfile	none	Create a new EmbPattern with a new tab in the GUI.
1	openfile	filename string	Open an EmbPattern with the supplied filename 'fname'.
2	savefile	filename string	Save the current loaded EmbPattern to the supplied filname 'fname'.
3	scale	selected objects, 1 float	Scale all selected objects by the number supplied, without selection scales the entire design
4	circle	mouse co-ords	Adds a circle to the design based on the supplied numbers, converts to stitches on save for stitch only formats.
5	offset	mouse co-ords	Shifts the selected objects by the amount given by the mouse co-ordinates.
6	extend		
7	trim		
8	break_at_point		
9	break_2_points		
10	fillet		
11	star		
12	singlelinetext		
13	chamfer		
14	split		
15	area		
16	time		
17	pickadd		
16	zoomfactor		
17	product		
18	program		
19	zoomwindow		
20	divide		
21	find		
22	record		
23	playback		
24	rotate		
25	rgb		
26	move		

27 grid
28 griphot
29 gripcolor
30 gripcool
31 gripsize
32 highlight
33 units
34 locatepoint
35 distance
36 arc
37 ellipse
38 array
39 point
40 polyline
41 polygon
42 rectangle
43 line
44 arc (rt)
45 dolphin
46 heart

6.1.9 Removed Elements

So I've had a few pieces of web infrastructure fail me recently and I think it's worth noting. An issue that affects us is an issue that can effect people who use our software.

6.1.10 Qt and dependencies

Downloading and installing Qt has been a pain for some users (46Gb on possibly slow connections).

I'm switching to FreeGLUT 3 (which is a whole other conversation) which means we can ship it with the source code package meaning only a basic build environment is necessary to build it.

6.1.11 Social Platform

Github is giving me a server offline (500) error and is still giving a bad ping.

So... all the issues and project boards etc. being on Github is all well and good assuming that we have our own copies. But we don't if Github goes down or some other major player takes over the space and we have to move (again, since this started on SourceForge).

This file is a backup for that which is why I'm repeating myself between them.

6.1.12 Pandoc Documentation

The documentation is, well better in that it's housed in the main repository, but I'm not a fan of the "write once build many" approach as it means trying to weigh up how 3 versions are going to render.

Can we treat the website being a duplicate of the docs a non-starter? I'd be happier with tex/pdf only and (I know this is counter-intuitive) one per project.

6.1.13 OpenGL

OpenGL rendering within the application. This will allow for Realistic Visualization - Bump Mapping/OpenGL/-Gradients?

This should backend to a C renderer or something.

6.1.14 Configuration Data Ideas

Embroidermodder should boot from the command line regardless of whether it is or is not installed (this helps with testing and running on machines without root). Therefore, it can create an initiation file but it won't rely on its existence to boot: ' /.embroidermodder/config.json'.

- Switch colors to be stored as 6 digit hexcodes with a #.
- We've got close to a hand implemented ini read/write setup in 'settings.py'.

6.1.15 Distribution

When we release the new pip wheel we should also package:
* '.tar.gz' and '.zip' source archive. * Debian package * RPM package
Only do this once per minor version number.

6.1.16 Scripting Overhaul

Originally Embroidermodder had a terminal widget, this is why we removed it.

> **ROBIN**
>
> I think supporting scripting within Embroidermodder doesn't make sense.
>
> All features that use scripting can be part of libembroidery instead. Users who are capable of using scripting won't need it, they can alter their embroidery files in CSV format, or import pyembroidery to get access. It makes maintaining the code a lot more complicated, especially if we move away from Qt. Users who don't want the scripting feature will likely be confused by it, since we say that's what libembroidery, embroider and pyembroidery are for.
>
> How about a simpler "call user shell" feature? Similar to texmaker we just call system on a batch or shell script supplied by the user and it processes the file directly then the software reloads the file. Then we aren't parsing it directly.
>
> I don't want to change this without Josh's support because it's a fairly major change.
>
> **JOSH**
>
> I totally agree.
>
> I like the idea of scripting just so people that know how to code could write their own designs without needing to fully build the app. Scripting would be a very advanced feature that most users would be confused by. Libembroidery would be a good fit for advanced features.
>
> Now we are using Python (again, sort of) this would be a lot more natural, perhaps we could boot the software without blocking the shell so they can interact? TODO: Screenshot a working draft to demonstrate.

6.1.17 Perennial Jobs

1. Check for memory leaks
2. Clear compiler warnings on '-Wall -ansi -pedantic' for C.
3. Write new tests for new code.
4. Get Embroidermodder onto the current version of libembroidery.
5. PEP7 compliance.
6. Better documentation with more photos/screencaps.

6.1.18 Full Test Suite

(This needs a hook from Embroidermodder to embroider's full test suite.)

The flag '–full-test-suite' runs all the tests that have been written. Since this results in a lot of output the details are both to stdout and to a text file called 'test_matrix.txt'.

Patches that strictly improve the results in the 'test_matrix.txt' over the current version will likely be accepted and it'll be a good place to go digging for contributions. (Note: strictly improve means that the testing result for each test is as good a result, if not better. Sacrificing one critera for another would require some design work before we would consider it.)

6.1.19 Symbols

Symbols use the SVG path syntax.

In theory, we could combine the icons and symbols systems, since they could be rendered once and stored as icons in Qt. (Or as textures in FreeGLUT.)

Also we want to render the patterns themselves using SVG syntax, so it would save on repeated work overall.

Chapter 7

Tutorials

7.1 Basic Features

7.1.1 Move a single stitch in an existing pattern

1. In the 'File' menu, click 'Open...'. When the open dialog appears find and select your file by double clicking the name of the file. Alternatively, left click the file once then click the 'Open' button.
2.
3. In the 'File' menu

TIP: For users who prefer

7.1.2 Convert one pattern to another

1. In the 'File' menu, click 'Open...'.
2. The
3. In the dropdown menu within the save dialog select the

7.1.3 Advanced Features

7.1.4 Format Support

Support for Singer FHE, CHE (Compucon) formats?

7.2 Embroidermodder Project Coding Standards

A basic set of guidelines to use when submitting code.

7.2.1 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.

7.3 Code Style

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

7.3.1 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.

7.3.2 Version Control

Being an open source project, developers can grab the latest code at any time and attempt to build it themselves. We try our best to ensure that it will build smoothly at any time, although occasionally we do break the build. In these instances, please provide a patch, pull request which fixes the issue or open an issue and notify us of the problem, as we may not be aware of it and we can build fine.

Try to group commits based on what they are related to: features/bugs/comments/graphics/commands/etc...

7.3.3 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 and Embroidermodder are 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 better soluti
/* WARNING: Think twice (or more times) before changing this code. I put this here for a good reason. */
/* NOTE: This comment is much more important than lesser comments. */
```

7.4 Ideas

7.4.1 Why this document

I've been trying to make this document indirectly through the Github issues page and the website we're building but I think a straightforward, plain-text file needs to be the ultimate backup for this. Then I can have a printout while I'm working on the project.

7.4.2 googletests

gtests are non-essential, testing is for developers not users so we can choose our own framework. I think the in-built testing for libembroidery was good and I want to re-instate it.

7.4.3 Qt and dependencies

I'm switching to SDL2 (which is a whole other conversation) which means we can ship it with the source code package meaning only a basic build environment is necessary to build it.

7.4.4 Documentation

Can we treat the website being a duplicate of the docs a non-starter? I'd be happier with tex/pdf only and (I know this is counter-intuitive) one per project.

7.4.5 Social Platform

So... all the issues and project boards etc. being on Github is all well and good assuming that we have our own copies. But we don't if Github goes down or some other major player takes over the space and we have to move (again, since this started on SourceForge).

This file is a backup for that which is why I'm repeating myself between them.

7.4.6 Identify the meaning of these TODO items

- Saving CSV/SVG (rt) + CSV read/write UNKNOWN interpreted as COLOR bug #179
- Lego Mindstorms NXT/EV3 ports and/or commands

7.4.7 Progress Chart

The chart of successful from-to conversions (previously a separate issue) is something that should appear in the README.

7.4.8 Style

Rather than maintain our own standard for style, please defer to the Python's PEP 7 ([12](#12)) for C style. If it passes the linters for that we consider it well styled for a pull request.

As for other languages we have no house style other than whatever "major" styles exist, for example Java in Google style ([13](#13)) would be acceptable. We'll elect specific standards if it becomes an issue.

7.4.9 Standard

The criteria for a good Pull Request from an outside developer has these properties, from most to least important:

- No regressions on testing.
- Add a feature, bug fix or documentation that is already agreed on through GitHub issues or some other way with a core developer.
- No GUI specific code should be in libembroidery, that's for Embroidermodder.
- Pedantic/ansi C unless there's a good reason to use another language.
- Meet the style above (i.e. [PEP 7, Code Lay-out](https://peps.python.org/pep-0007/#code-lay-out)). We'll just fix the style if the code's good and it's not a lot of work.
- 'embroider' should be in POSIX style as a command line program.
- No dependancies that aren't "standard", i.e. use only the C Standard Library.

7.4.10 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 .

7.4.11 To Place

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 ' '.

7.4.12 To Do

We currently need help with:

- Thorough descriptions of each embroidery format.
- Finding resources for each of the branded thread libraries (along with a full citation for documentation).
- Finding resources for each geometric algorithm used (along with a full citation for documentation).
- Completing the full '–full-test-suite' with no segfaults and at least a clear error message (for example "not implemented yet").
- 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?
- Improving the written documentation.
- 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:

- Basic Features
- Code quality and user friendliness
- embroider CLI
- Documentation
- GUI
- electronics development

7.4.13 Basic features

- Incorporate '#if 0'ed parts of 'libembroidery.c'.
- Interpret how to write formats that have a read mode from the source code and vice versa.
- 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.
- Find more and better documentation of the structure of the headers for the formats we do understand.

7.4.14 Code quality and user friendliness

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

7.4.15 embroider CLI

- Make '-circle' flag to add a circle to the current pattern.
- Make '-rect' flag to add a rectangle to the current pattern.
- 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).
- Make '-ellipse' flag to add to ellipse to the current pattern.
- Make '-bezier' flag to add a bezier curve to the current pattern.

7.4.16 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:

- a circle with properties: thickness 2, satin 0.3,0.6
- a disc with properties:
- an arc with properties:

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

7.4.17 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.

7.4.18 GUI

1. Make EmbroideryMobile (Android) also backend to 'libembroidery' with a Java wrapper.
2. Make EmbroideryMobile (iOS) 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.

7.5 Electronics development

- Currently experimenting with Fritzing[8](8), upload netlists to embroiderbot when they can run simulations using the asm in 'libembroidery'.
- Create a common assembly for data that is the same across chipsets 'libembrodiery_data_internal.s'.
- 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.

7.6 Development

7.6.1 Contributing

If you're interested in getting involved, here's some guidance for new developers. Currently The Embroidermodder Team is all hobbyists with an interest in making embroidery machines more open and user friendly. If you'd like to support us in some other way you can donate to our Open Collective page (click the Donate button) so we can spend more time working on the project.

All code written for libembroidery should be ANSI C89 compliant if it is C. Using other languages should only be used where necessary to support bindings.

7.6.2 Debug

If you wish to help with development, run this debug script and send us the error log.

```
#!/bin/bash

rm -fr libembroidery-debug

git clone http://github.com/embroidermodder/libembroidery libembroidery-debug
cd libembroidery-debug

cmake -DCMAKE_BUILD_TYPE=DEBUG .
cmake --build . --config=DEBUG

valgrind ./embroider --full-test-suite
```

While we will attempt to maintain good results from this script as part of normal development it should be the first point of failure on any system we haven't tested or format we understand less.

7.6.3 Binary download

We need a current 'embroider' command line program download, so people can update without building.

Chapter 8

Formats

8.1 Overview

8.2 Read/Write Support Levels

The table of read/write format support levels uses the status levels described here:

Status Label	Description
'rw-none'	Either the format produces no output, reporting an error. Or it produces a Tajima dst file as an alternative.
'rw-poor'	A file somewhat similar to our examples is produced. We don't know how well it runs on machines in practice as we don't have any user reports or personal tests.
'rw-basic'	Simple files in this format run well on machines that use this format.
'rw-standard'	Files with non-standard features work on machines and we have good documentation on the format.
'rw-reliable'	All known features don't cause crashes. Almost all work as expected.
'rw-complete'	All known features of the format work on machines that use this format. Translations from and to this format preserve all features present in both.

These can be split into 'r-basic w-none', for example, if they don't match.

So all formats can, in principle, have good read and good write support, because it's defined in relation to files that we have described the formats for.

8.2.1 Test Support Levels

Status Label	Description
test-none	No tests have been written to test the specifics of the format.
test-basic	Stitch Lists and/or colors have read/write tests.
test-thorough	All features of that format has at least one test.
test-fuzz	Can test the format for uses of features that we haven't thought of by feeding in nonsense that is designed to push possibly dangerous weaknesses to reveal themselves.
test-complete	Both thorough and fuzz testing is covered.

So all formats can, in principle, have complete testing support, because it's defined in relation to files that we have described the formats for.

8.2.2 Documentation Support Levels

Status Label	Description

— ‘doc-none‘ — We haven’t researched this beyond finding example files. — — ‘doc-basic‘ — We have a rough sketch of the size and contents of the header if there is one. We know the basic stitch encoding (if there is one), but not necessarily all stitch features. — — ‘doc-standard‘ — We know some good sources and/or have tested all the features that appear to exist. They mostly work the way we have described. — — ‘doc-good‘ — All features that were described somewhere have been covered here or we have thoroughly tested our ideas against other softwares and hardwares and they work as expected. — — ‘doc-complete‘ — There is a known official description and our description covers all the same features. —

Not all formats can have complete documentation because it’s based on what information is publically available. So the total score is reported in the table below based on what level we think is available.

8.2.3 Overall Support

Since the overall support level is the combination of these 4 factors, but rather than summing up their values it’s an issue of the minimum support of the 4.

Status Label	Description

— ‘read-only‘ — If write support is none and read support is not none. — — ‘write-only‘ — If read support is none and write support is not none. — — ‘unstable‘ — If both read and write support are not none but testing or documentation is none. — — ‘basic‘ — If all ratings are better than none. — — ‘reliable‘ — If all ratings are better than basic. — — ‘complete‘ — If all ratings could not reasonably be better (for example any improvements rely on information that we may never have access to). This is the only status that can be revoked, since if the format changes or new documentation is released it is no longer complete. — — ‘experimental‘ — For all other scenarios. —

8.2.4 Table of Format Support Levels

Overview of documentation support by format.

Format	Ratings	Score
Toyota Embroidery Format (.100)	rw-basic doc-none test-none	unstable
Toyota Embroidery Format (.10o)	rw-basic doc-none test-none	unstable
Bernina Embroidery Format (.art)	rw-none doc-none test-none	experimental
Bitmap Cache Embroidery Format (.bmc)	r-basic w-none doc-none test-none	unstable
Bits and Volts Embroidery Format (.bro)	rw-none doc-none test-none	experimental
Melco Embroidery Format (.cnd)	rw-none doc-none test-none	experimental
Embroidery Thread Color Format (.col)	rw-basic doc-none test-none	‘experimental‘
Singer Embroidery Format (.csd)	rw-none doc-none test-none	experimental
Comma Separated Values (.csv)	rw-none doc-none test-none	experimental
Barudan Embroidery Format (.dat)	rw-none doc-none test-none	experimental
Melco Embroidery Format (.dem)	rw-none doc-none test-none	experimental
Barudan Embroidery Format (.dsb)	rw-none doc-none test-none	experimental
Tajima Embroidery Format (.dst)	rw-none doc-none test-none	experimental
ZSK USA Embroidery Format (.dsz)	rw-none doc-none test-none	experimental
Drawing Exchange Format (.dxf)	rw-none doc-none test-none	experimental
Embird Embroidery Format (.edr)	rw-none doc-none test-none	experimental
Elna Embroidery Format (.emd)	rw-none doc-none test-none	experimental
Melco Embroidery Format (.exp)	rw-none doc-none test-none	experimental
Eltac Embroidery Format (.exy)	rw-none doc-none test-none	experimental
Sierra Expanded Embroidery Format (.eys)	rw-none doc-none test-none	experimental
Fortron Embroidery Format (.fxy)	rw-none doc-none test-none	experimental
Smoothie G-Code Embroidery Format (.gc)	rw-none doc-none test-none	experimental
Great Notions Embroidery Format (.gnc)	rw-none doc-none test-none	experimental
Gold Thread Embroidery Format (.gt)	rw-none doc-none test-none	experimental
Husqvarna Viking Embroidery Format (.hus)	rw-none doc-none test-none	experimental
Inbro Embroidery Format (.inb)	rw-none doc-none test-none	experimental

Embroidery Color Format (.inf)	rw-none doc-none test-none	experimental
Janome Embroidery Format (.jef)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.ksm)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.max)	rw-none doc-none test-none	experimental
Mitsubishi Embroidery Format (.mit)	rw-none doc-none test-none	experimental
Ameco Embroidery Format (.new)	rw-none doc-none test-none	experimental
Melco Embroidery Format (.ofm)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.pcd)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.pcm)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.pcq)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.pcs)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.pec)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.pel)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.pem)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.pes)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.phb)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.phc)	rw-none doc-none test-none	experimental
AutoCAD Embroidery Format (.plt)	rw-none doc-none test-none	experimental
RGB Embroidery Format (.rgb)	rw-none doc-none test-none	experimental
Janome Embroidery Format (.sew)	rw-none doc-none test-none	experimental
Husqvarna Viking Embroidery Format (.shv)	rw-none doc-none test-none	experimental
Sunstar Embroidery Format (.sst)	rw-none doc-none test-none	experimental
Data Stitch Embroidery Format (.stx)	rw-none doc-none test-none	experimental
Scalable Vector Graphics (.svg)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.t01)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.t09)	rw-none doc-none test-none	experimental
Happy Embroidery Format (.tap)	rw-none doc-none test-none	experimental
ThredWorks Embroidery Format (.thr)	rw-none doc-none test-none	experimental
Text File (.txt)	rw-none doc-none test-none	experimental
Barudan Embroidery Format (.u00)	rw-none doc-none test-none	experimental
Barudan Embroidery Format (.u01)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.vip)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.vp3)	rw-none doc-none test-none	experimental
Singer Embroidery Format (.xxx)	rw-none doc-none test-none	experimental
ZSK USA Embroidery Format (.zsk)	rw-none doc-none test-none	experimental

8.2.5 Toyota Embroidery Format (.100)

The Toyota 100 format is a stitch-only format that uses an external color file.
The stitch encoding is in 4 byte chunks.

8.2.6 Toyota Embroidery Format (.10o)

The Toyota 10o format is a stitch-only format that uses an external color file.
The stitch encoding is in 3 byte chunks.

8.2.7 Bernina Embroidery Format (.art)

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

8.2.8 Bitmap Cache Embroidery Format (.bmc)

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

8.2.9 Bits and Volts Embroidery Format (.bro)

The Bits and Volts bro format is a stitch-only format that uses an external color file.
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

8.3 Melco Embroidery Format (.cnd)

The Melco cnd format is a stitch-only format.

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

8.4 Embroidery Thread Color Format (.col)

An external color file format for formats that do not record their own colors.

It is a human-readable format that has a header that is a single line containing only the number of threads in decimal followed by the windows line break `\r\n`.

Then the rest of the file is a comma seperated value list of all threads with 4 values per line: the index of the thread then the red, green and blue channels of the color in that order.

8.4.1 Example

If we had a pattern called "example" with four colors: black, red, magenta and cyan in that order then the file is (with the white space written out):

example.col

```
4\r\n
0,0,0,0\r\n
1,255,0,0\r\n
2,0,255,0\r\n
3,0,0,255\r\n
```

8.5 Singer Embroidery Format (.csd)

Stitch Only Format

8.6 Comma Separated Values (.csv)

Comma Seperated Values files aren't a universal system, here we aim to offer a broad support. The dialect is detected based on the opening lines, as each manufacturer should label their CSV files there.

8.6.1 Embroidermodder 2.0 CSV Dialect

Our own version has the identifier comment line:

Control Symbol	Type	Description
#	COMMENT	
>	VARIABLE	To store records of a pattern's width, height etc. This means that data stored in the header of say a .dst file is preserved.
$	THREAD	
*	STITCH	
*	JUMP	
*	COLOR	To change a color: used for trim as well
*	END	To end a pattern.
*	UNKNOWN	For any feature that we can't identify.

8.6.2 EmBird CSV Dialect

8.7 Barudan Embroidery Format (.dat)

Stitch Only Format

8.8 Melco Embroidery Format (.dem)

Stitch Only Format

8.9 Barudan Embroidery Format (.dsb)

- Stitch Only Format.

X Basic Read Support

o Basic Write Support

o Well Tested Read

o Well Tested Write

8.10 Tajima Embroidery Format (.dst)

- Stitch Only Format.

X Basic Read Support

X 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.

Other references: Community, acatina.

8.10.1 Header

The header contains general information about the design. It is in lines of ASCII, so if you open a DST file as a text file, it's the only part that's easy to read. The line ending symbol is `0x0D`. The header is necessary for the file to be read by most softwares and hardwares.

The header is 125 bytes of data followed by padding spaces to make it 512 bytes in total.

The lines are as follows.

Label	Size	Description	Example
`"LA:"`	17	The design name with no path or extension. The space reserved is 16 characters, but the name must not be longer than 8 and be padded to 16 with spaces (0x20).	`"LA:Star "`
`ST:`	8	The stitch count. An integer in the format `%07d`, that is: a 7 digit number padded by leading zeros. This is the total accross all possible stitch flags.	
`CO:`	4	The number of color changes (not to be confused with thread count, an all black design we would have the record **000**). An integer in the format `%03d`, that is: a 3 digit number padded by leading zeros.	
`+X:`	6	The extent of the pattern in the postitive x direction in millimeters. An integer in the format `%05d`, that is: a 5 digit number padded by leading zeros.	
`-X:`	6	The extent of the pattern in the negative x direction in millimeters. An integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.	
`+Y:`	6	The extent of the pattern in the postitive y direction in millimeters. An integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.	

`-Y:`	6	The extent of the pattern in the negative y direction in millimeters. An integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.
`AX:`	7	The difference of the end from the start in the x direction in 0.1mm, the first char should be the sign, followed by an integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.
`AY:`	7	The difference of the end from the start in the y direction in 0.1mm, the first char should be the sign, followed by an integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.
`MX:`	7	The x co-ordinate of the last point in the previous file should the design span multiple files. Like AX, it is the sign, followed by a 5 digit integer. If we have a one file design set it to zero.
`MY:`	7	The y co-ordinate of the last point in the previous file should the design span multiple files. Like AY, it is the sign, followed by a 5 digit integer. If we have a one file design set it to zero.
`PD:`	10	Information about multivolume designs.

8.10.2 Stitch Data

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

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
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 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. The coordinate system is right handed.

8.11 ZSK USA Embroidery Format (.dsz)

The ZSK USA dsz format is stitch-only.

8.12 Drawing Exchange Format (.dxf)

Graphics format.

8.13 Embird Embroidery Format (.edr)

Stitch Only Format

8.14 Elna Embroidery Format (.emd)

Stitch Only Format.

8.15 Melco Embroidery Format (.exp)

Stitch Only Format.

8.16 Eltac Embroidery Format (.exy)

Stitch Only Format.

8.17 Sierra Expanded Embroidery Format (.eys)

Stitch Only Format.
Smoothie G-Code Embroidery Format (.fxy)?

8.18 Fortron Embroidery Format (.fxy)

Stitch Only Format.

8.19 Great Notions Embroidery Format (.gnc)

Stitch Only Format.

8.20 Gold Thread Embroidery Format (.gt)

Stitch Only Format.

8.21 Husqvarna Viking Embroidery Format (.hus)

Stitch Only Format.

8.22 Inbro Embroidery Format (.inb)

Stitch Only Format.

8.23 Embroidery Color Format (.inf)

Stitch Only Format.

8.24 Janome Embroidery Format (.jef)

Stitch Only Format.

8.25 Pfaff professional Design format (.ksm)

Stitch Only Format.

8.26 Pfaff Embroidery Format (.max)

Stitch Only Format.

8.27 Mitsubishi Embroidery Format (.mit)

Stitch Only Format.

8.28 Ameco Embroidery Format (.new)

Stitch Only Format.

8.29 Melco Embroidery Format (.ofm)

Stitch Only Format.

8.30 Pfaff PCD File Format (.pcd)

Stitch Only Format.
The format uses a signed 3 byte-length number type.
See the description here ([5](5)) for the overview of the format.
For an example of the format see ([11](11)).

8.31 Pfaff Embroidery Format (.pcm)

The Pfaff pcm format is stitch-only.

8.32 Pfaff Embroidery Format (.pcq)

The Pfaff pcq format is stitch-only.

8.33 Pfaff Embroidery Format (.pcs)

The Pfaff pcs format is stitch-only.

8.34 Brother Embroidery Format (.pec)

The Brother pec format is stitch-only.

8.35 Brother Embroidery Format (.pel)

The Brother pel format is stitch-only.

8.36 Brother Embroidery Format (.pem)

The Brother pem format is stitch-only.

8.37 Brother Embroidery Format (.pes)

The Brother pes format is stitch-only.

8.38 Brother Embroidery Format (.phb)

The Brother phb format is stitch-only.

8.39 Brother Embroidery Format (.phc)

The Brother phc format is stitch-only.

8.40 AutoCAD Embroidery Format (.plt)

The AutoCAD plt format is stitch-only.

8.41 RGB Color File (.rgb)

The RGB format is a color-only format to act as an external color file for other formats.

8.42 Janome Embroidery Format (.sew)

The Janome sew format is stitch-only.

8.43 Husqvarna Viking Embroidery Format (.shv)

The Husqvarna Viking shv format is stitch-only.

8.44 Sunstar Embroidery Format (.sst)

The Sunstar sst format is stitch-only.

8.45 Data Stitch Embroidery Format (.stx)

The Data Stitch stx format is stitch-only.

8.46 Scalable Vector Graphics (.svg)

The scalable vector graphics (SVG) format is a graphics format maintained by ...

8.47 Pfaff Embroidery Format (.t01)

The Pfaff t01 format is stitch-only.

8.47.1 Pfaff Embroidery Format (.t09)

The Pfaff t09 format is stitch-only.

8.48 Happy Embroidery Format (.tap)

The Happy tap format is stitch-only.

8.49 ThredWorks Embroidery Format (.thr)

The ThreadWorks thr format is stitch-only.

8.50 Text File (.txt)

The txt format is stitch-only and isn't associated with a specific company.

8.51 Barudan Embroidery Format (.u00)

The Barudan u00 format is stitch-only.

8.52 Barudan Embroidery Format (.u01)

The Barudan u01 format is stitch-only.

8.53 Pfaff Embroidery Format (.vip)

The Pfaff vip format is stitch-only.

8.54 Pfaff Embroidery Format (.vp3)

The Pfaff vp3 format is stitch-only.

8.55 Singer Embroidery Format (.xxx)

The Singer xxx format is stitch-only.

8.56 ZSK USA Embroidery Format (.zsk)

The ZSK USA zsk format is stitch-only.

8.57 On Embedded Systems

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

8.58 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.

8.59 Arduino Considerations

There are two main concerns here: Flash Storage and 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.

8.60 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.

8.61 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.

8.62 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.

8.63 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.

8.64 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 software 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.

8.65 Features

8.66 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:

- 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: https://stackoverflow.com/questions/11425202/is-it-possible-to-c

For Python you can do the same using ctypes: https://www.geeksforgeeks.org/how-to-call-a-c-function-in-pyth

8.67 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

Chapter 9

Actuator

9.1 ARC

9.2 CIRCLE

9.3 OPEN

Chapter 10

Libembroidery API

10.1 EmbVector

The basic vector type for libembroidery is:

```
typedef struct EmbVector_ {
    float x;
    float y;
} EmbVector;
```

Equivalent to:

$$\begin{pmatrix} a_x \\ a_y \end{pmatrix} \tag{10.1}$$

10.1.1 embVector_add(EmbVector a, EmbVector b) → EmbVector

Takes two arguments, both EmbVectors and adds them together, returning the result.
Equivalent to:

$$\begin{pmatrix} c_x \\ c_y \end{pmatrix} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \end{pmatrix} \tag{10.2}$$

10.1.2 embVector_subtract(EmbVector a, EmbVector b) → EmbVector

Takes two arguments, both EmbVectors and adds them together, returning the result.
Equivalent to:

$$\begin{pmatrix} c_x \\ c_y \end{pmatrix} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} - \begin{pmatrix} b_x \\ b_y \end{pmatrix} \tag{10.3}$$

10.2 EmbCircle

```
typedef struct EmbVector_ {
    EmbVector center;
    float radius;
    EmbColor color;
} EmbVector;
```

10.3 EmbEllipse

```
typedef struct EmbEllipse_ {
    float x;
    float y;
} EmbEllipse;
```

10.4 EmbRect

```
typedef struct EmbRect_ {
    float top;
    float bottom;
    float left;
    float right;
} EmbRect;
```

10.4.1 embRect_width

10.4.2 embRect_height

10.5 EmbArray

```
    typedef struct EmbArray_ {

    } EmbArray;
```

10.5.1 embArray_addCircle

10.5.2 embArray_addRect

Bibliography

acatina. Technical info. URL http://www.achatina.de/sewing/main/TECHNICL.HTM.

Inc. AutoDesk. *DXF Reference*. AutoDesk, Inc., 2012. URL https://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_enu.pdf.

KDE Community. Projects/liberty/file formats/tajima ternary - kde community wiki. URL https://community.kde.org/Projects/Liberty/File_Formats/Tajima_Ternary.

GNU Free Documentation License

Version 1.3, 3 November 2008

<https://fsf.org/>

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 © 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.

Index