

# Keep Dreaming Project

Project specific documentation

- [Overview](#)
- [Contributing to the Keep Dreaming Project](#)
- [G1 Bus Breakout](#)
- [openGDROM PCB](#)
- [Maple Bus Breakout](#)
- [Phabricator](#)
- [InvenTree](#)
- [BookStack](#)
- [Arcanist](#)

# Overview

The Keep Dreaming Project aims to provide the Dreamcast community with an opensource ecosystem for collaborating on electronics and software projects along with tools for gathering and preserving information. The project consists of custom circuit boards for hardware research/development and a hosted software stack for collaboration and preserving information.

- 1 If you're interested in contributing checkout the [how to contribute](#) page, also checkout the [roadmap](#) and [tasks](#) to see the projects status and recent activity.

## Electronics

I (Zeigren) am designing various development and breakout boards that interface with the Dreamcast and its peripherals. I'm primarily responsible for this part of the project.

## Development Boards

- ✔ Goal: Have a common set of opensource development boards that anyone can make and use for Dreamcast projects.

This makes collaboration easier since we can all use the same tools and software, sort of like Arduino. Code is easy to share and can be expected to work the same on each development board.

Having hardware designed for working with the Dreamcast will also make development easier. No need to fiddle with a mess of wires and chase down connection problems, everything will plug directly into the Dreamcast.

### G1 Bus

The G1 Bus is what the Dreamcast GD-ROM drive is connected to. I've already designed a development board for the G1 Bus in the form of the openGDROM, which I'm currently redesigning.

- ✔ Goal: Create a opensource GD-ROM drive emulator, like the GDEMU or USB-GDROM.



## G2 Bus

The G2 Bus is what the Dreamcast's modem is connected to. The G2 Bus development board is still in the planning stage. The current plan is to base its design off of the G1 Bus development board.

- ✓ Goal: Create a wireless network adapter with the G2 Bus development board, but I may try to design it in such a way that it could be used to create a LAN adapter.

## Maple Bus

The Maple Bus is what controllers and other peripherals are connected to.

- ✓ Goal: Create a wireless controller using the Maple Bus development boards.

The wireless controller is in the planning/proof of concept stage. This consists of two parts, the Dreamcast side and the controller side. I have a pretty good idea of what the design is going to look like but there's still testing to do.

## Serial Port

This is still in the planning stage. This is generally the simplest of them all on the electronics side. Right now I'm just trying to figure out a good, cheap, and reliable way to connect to the serial port. Since we currently can't buy connectors for it (maybe in the future though!).

- ✓ Goal: Make a USB to Serial Port adapter.

You can still buy Serial Port to SD Card adapters, but that is something I would consider designing as well.

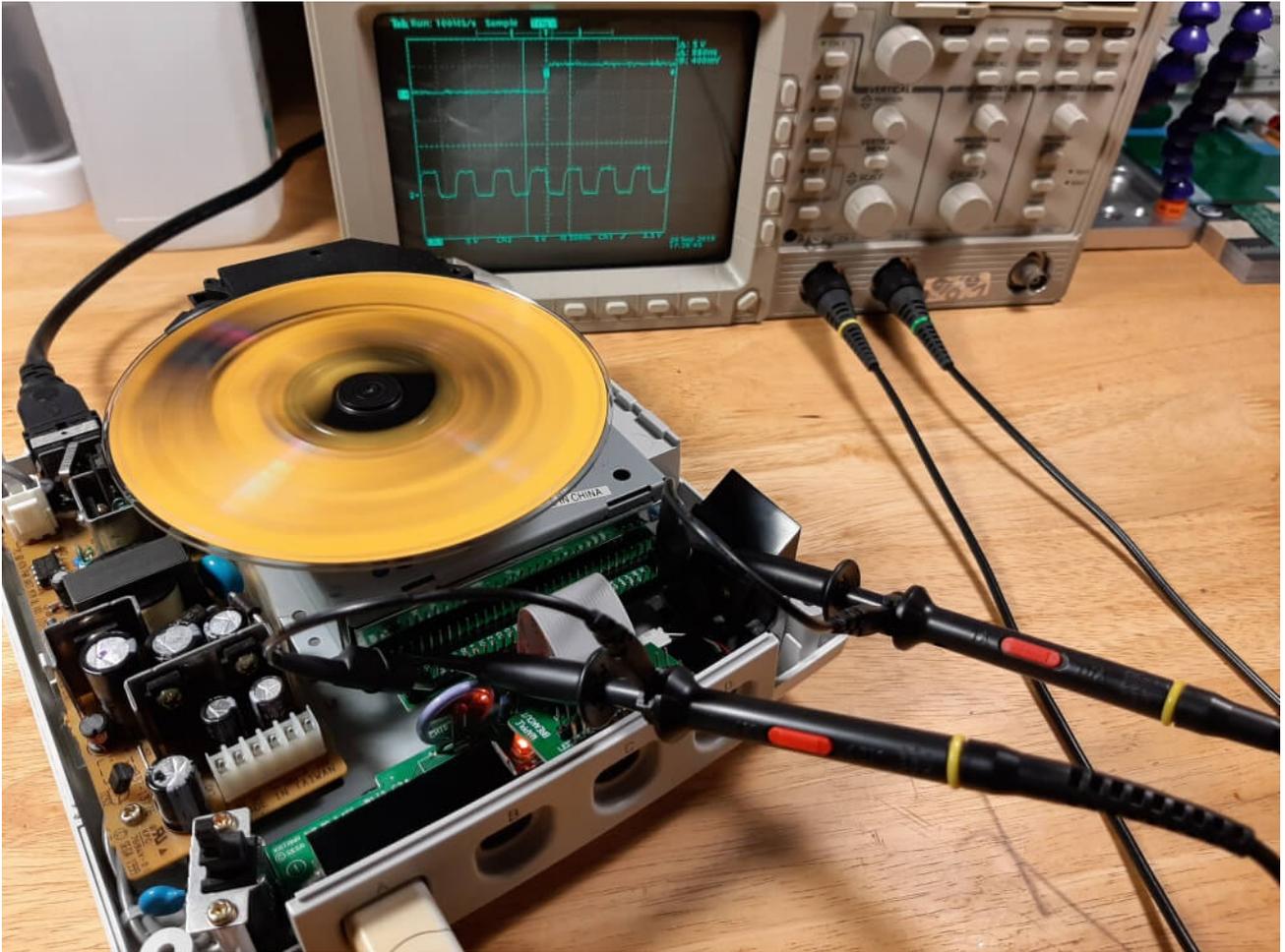
# Breakout Boards

- 1 The breakout boards are an easy way to monitor signals from either Dreamcast official hardware or development boards.

They also provide a more generic way of interacting with the Dreamcast.

## G1 Bus

I'm currently revising the G1 Bus breakout board.



## G2 Bus

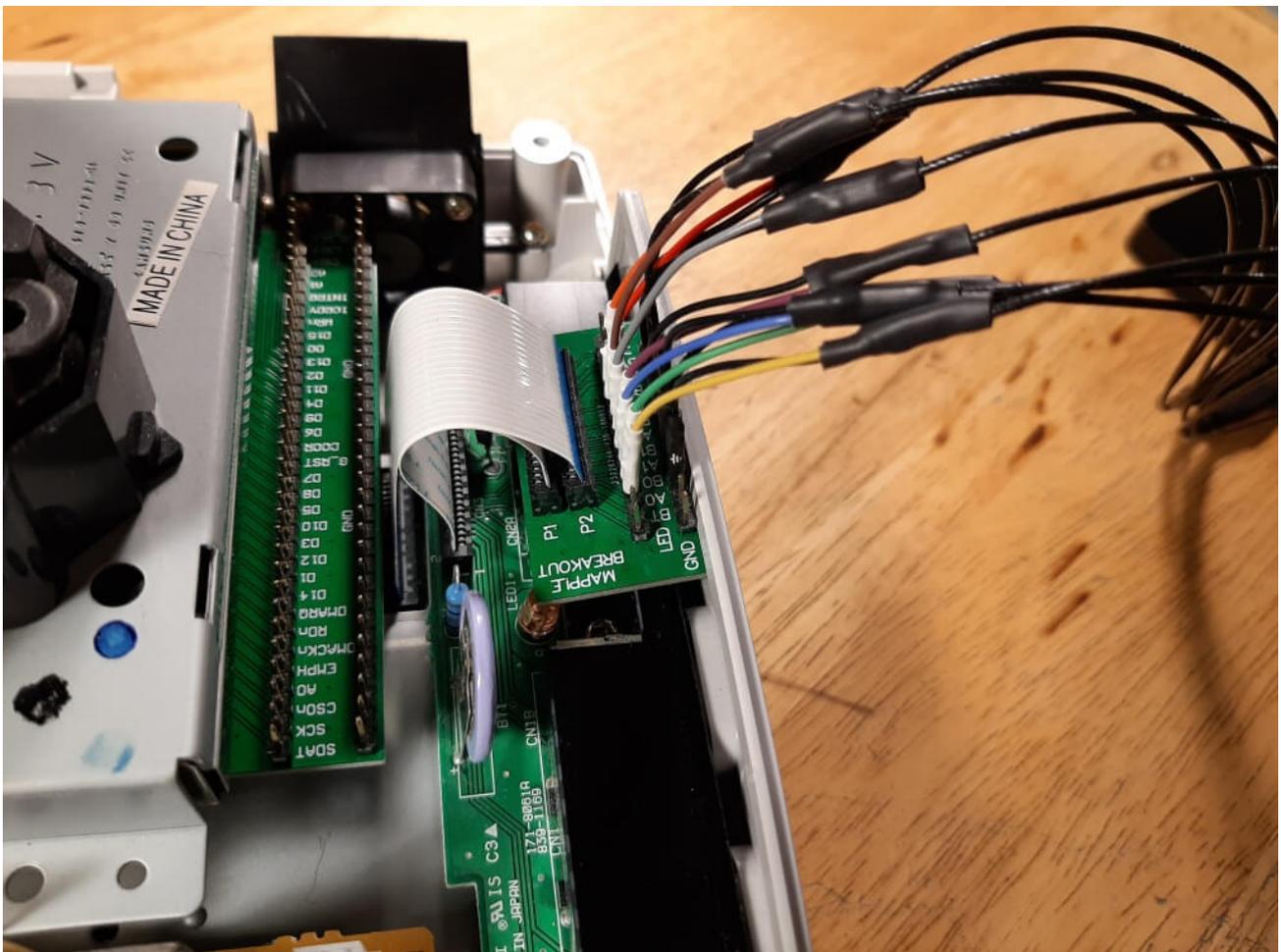
The G2 Bus breakout board is in the planning stage.

The difficult part here are the connectors, since there's currently no way to buy them. So I have to figure out a way to create a reliable connection without using a connector.

## Maple Bus

I'm currently revising the maple bus breakout board, mostly to make it easier to use.

I also need to design an alternate version for VA0 Dreamcasts.



## Serial Port

- 1 This is in the planning stage. Could be used to create system link cables.

See the Serial Port [Development](#) board section.

## Software

The software portion of this project has largely been related to the electronics portion, but it doesn't have to be.

- 2 Anyone working on a software Dreamcast project is free to use all the tools available here, the more the merrier!

This is where collaboration really comes into play, there's a lot of things to work on and no way I could do all of it on my own.

## GD-ROM ODE

This is where I've put in the most thought and time hence the openGDROM, but since I'm redesigning it there will be quite a few changes and I honestly didn't have much done in the way of software in the first place.

## Network Adapter

I only have some ideas for this.

## Wireless Controller

Currently in the proof of concept stage, then on to actually getting it to work with the development boards once they're done.

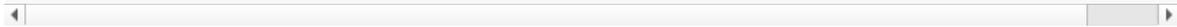
## Information

There's a lot of information and research out there about the Dreamcast, but it is all over the place and with the Dreamcast being over 20 years old a lot of websites with this information are fading away.

✔ Goal: Gather information about the Dreamcast to make it easy to access and easy to backup.

The primary focus is on the kind of things you wouldn't find on Wikipedia or video game database websites. Thoroughly covering the obscure and diving deep into the "quirks and features". I'll quote myself talking about the file contents of a MIL-CD as an example

“ PDATA10, likely padding data right? So it is! Inside you'll find repeating numbers and some repeating Japanese, here's an example 0010010,.. [Which is here](#). The padding file is filled with addresses from Sapporo



More general information is useful and still welcome, just not the primary focus as that information is easy to find.

## BookStack

📌 Most information and documentation will be hosted here on this BookStack instance.

Everything here can be easily exported as a PDF so you can make your own backups whenever you want.

## InvenTree

InvenTree is one of the most exciting parts of the project for me.

📌 It's used here as a Dreamcast hardware database, a way to catalog console variants and all the parts/assemblies used in the hardware itself.

The database can also be exported if you want a local copy.

## Preservation

📌 In order to preserve this information long term and avoid the fate of so many other sites (RIP Assemblergames) I've made it so this website in its entirety could be easily hosted by someone else.

The website consists of a number of easy to use Docker stacks and the website itself is regularly backed up. By giving a couple of trusted members of the Dreamcast community access to the encrypted backups the information here could be easily brought back online if I was no longer able to maintain it.

## Tools

This is all of the infrastructure used to make the project possible. Which primarily consists of a number of opensource projects and all the stuff used to host them and tie them together. I've spun this part of the project off into [kairohm](#) as it can be used for more than just the Keep Dreaming Project, although this is largely symbolic. To quote myself again

“ kairohm aims to bring together existing opensource tools to create a fully-featured ecosystem for collaboration without relying on third-party services

📌 This is inline with the preservation part of the project, when using third-party services you're at their whim and they typically don't have a good way for users to back up information.

## Hosted Tools

[phabricator.kairohm.dev](https://phabricator.kairohm.dev) brought to you by [Phabricator](#)

For code review, repository hosting, bug tracking, project management, and more.

This is the hub for all the software and electronics collaboration.

[bookstack.kairohm.dev](https://bookstack.kairohm.dev) brought to you by [BookStack](#)

As a knowledge base and for documentation. You're here!

[inventree.kairohm.dev](https://inventree.kairohm.dev) brought to you by [InvenTree](#)

As a Dreamcast hardware database.

## Webhook Bot

Simple chat bot for relaying messages from services.

## Matrix/Riot (official links)

For chat (planned).

# Supporting Tools

## tikatree

Directory tree metadata parser using [Apache Tika](#)

## QMSV

Currently a Sublime Text package for working with Quartus, ModelSim, and System Verilog.

# Third Party Tools

Leveraging third party tools is still useful, but the project doesn't rely on them.

## Discord Server

For chat.

## Docker Hub

As a docker registry and for auto building containers.

## GitHub

As a repository mirror and if people really don't want to contribute using Phabricator.

# Contributing to the Keep Dreaming Project

Thanks for helping out! This will be a general guide on how to contribute to the project, enjoy!

📌 Checkout the [overview](#) for more information about the project and the [how to join](#) section below, to well join.

## How Can I Help?

The project has a pretty big scope so I'm sure you'll find something you can help with! I'll outline what some of those areas are below.

✅ Checkout the [roadmap](#) and [tasks](#) on [Phabricator](#) for more specific ideas.

✅ Planning and collaboration for the project occurs on [Discord](#) and [Phabricator](#), that's the best way to contact us. Feel free to bring up any ideas you have for the project!

⚠️ GitHub is used solely as a repository mirror, issues and contributions should be done using [Phabricator](#). Exceptions will be noted.

## Hardware

### Design

I plan on doing all the hardware design work myself, but I'm open to any input or ideas. There are some areas I might need help in:

- Testing development boards
- Creating tests to make sure they work as intended

### Dreamcast Variants

Right now I only have a handful of NA and JP VA1 Dreamcasts to examine and test with, I'd like to get my hands on more variants. Randomly buying Dreamcasts on eBay trying to find specific versions doesn't work very well.

## Software

There are quite a number of different software projects however many of them are in the planning stages.

📌 When the development boards are done we'll need help in essentially all facets of developing software for them.

## Relevant Knowledge

GD-ROM ODE, Network Adapter, Wireless Controller, Logic Analyzer

- C
- C++
- SystemVerilog
- Quartus

Sigrok Protocol Decoders

- Python

📌 Contributing to any of the opensource projects that the Keep Dreaming Project relies upon is great too, which you can checkout below in the [tools](#) section!

# Information

## Dreamcast Information

- Collecting and organizing information about the Dreamcast
- Contributing any research that you've done
- Adding model, assembly, and part information to InvenTree
- Taking and gathering pictures

## Documentation

Writing up documentation on the various facets of the Keep Dreaming Project itself.

- Hardware Documentation
  - Design, Operation, and Assembly
- Software Documentation
  - Operation, Compilation Instructions

## Guides

▲ How to actually contribute and use the services or any of the other stuff under the Keep Dreaming Project umbrella.

- Phabricator and arcanist
  - How to contribute code
  - How to report issues
  - How to request a feature or enhancement
  - How to use Phabricator and arcanist
- BookStack
  - How to use BookStack
  - Style Guide?
- InvenTree
  - How to use InvenTree
  - How to add models, parts, and assemblies
  - Explanation of templates and variants and how parts tie together

## Templates

▲ Creating templates for BookStack and tasks on Phabricator.

# Tools

Thankfully most of the major stuff is taken care of but there's still a lot to improve.

## InvenTree

▲ I've been working on adding permission, registration, and authentication systems to InvenTree to make it usable for our use case. Right now it's pretty hacked together and needs a lot of work.

## Relevant Knowledge

- Python
- Django
- Docker
- HTML
- CSS
- JavaScript

tikatree

🔧 Currently adding features and cleaning things up, but it works as intended for the most part

## Relevant Knowledge

- Python

## Webhook Bot

⚠️ This works but is really poorly implemented, it either needs to be cleaned up or redone.

## Relevant Knowledge

- Python
- Docker
- Rust?
- Go?

## VSCoDe

I need to migrate [QMSV](#) from being a Sublime-Text plugin to a VSCode extension. There is also a relatively new [Phabricator](#) extension that I plan on contributing to in order to make using arcanist easier.

## Relevant Knowledge

- Typescript
- Python
- SystemVerilog
- Quartus

## Patreon

Anything helps!

Contributing to Patreon helps in a couple of ways

- Helps with hardware costs (parts, PCBs, etc...)
- Lets me spend more time on the project
- Helps cover server costs

Currently the server costs are really low. But as the project grows the tiny VPS I'm paying for won't be enough.

# Joining The Project

✅ You can create an account for [BookStack](#) and [Phabricator](#), alternatively you can login using a GitHub account.

By default you will only have viewing permissions, you'll have to contact me in order to start contributing. Even though I can restore things I'd still like to avoid griefers.

You can contact me at any of these places (e-mail or Discord work best):

- [contributing@kairohm.dev](mailto:contributing@kairohm.dev)
- [Discord](#)
- [ObscureGamers](#)
- [Reddit](#)
- Or any other place you know me from

In your message include:

- What username you used to sign up
- And at least one of the options below
  - How and why you want to contribute

- Send me a link to a project or post that shows you know something about the Dreamcast/Programming/Electronics

It doesn't need to be lengthy or very good. Once I review it I can give you the appropriate permissions so you can start contributing!

Otherwise everyone is welcome on the [Discord](#) server and you can chat with us there.

 InvenTree is still under construction and is currently closed to contributions.

## Code of Conduct

You can e-mail me at [conduct@kairohm.dev](mailto:conduct@kairohm.dev) or contact me on [Discord](#) if you have any concerns.

## Licenses

### Hardware

All of my hardware designs are licensed under the [CERN OHL v1.2](#) or later.

### Software

Licenses will vary per project, typically either the [MIT](#) or a [GPL](#) variant.

### HDL

Currently there isn't a good opensource license for HDL, many other projects use the [MIT](#) or [GPL](#) license although they aren't designed for it. I would like to use the [CERN OHL v2](#) but it is still a draft.

### Text, Images, Etc

Anything not covered by the previous licences will use the [CC-BY-SA-4.0](#) license, exceptions will be noted.

# G1 Bus Breakout

## Overview

Breakout board for the Dreamcast G1 Bus (what the GD-ROM drive is connected to).

[Sub-project](#) and [Repository](#)

The repository contains:

- Altium Design Files
- Bill of Materials
- Schematic Prints
- Draftsman Board Design Documents
- Gerber Files
- Licence Information

## Fabrication

When getting the PCB manufactured make sure to use one of the tagged revisions in the repository, otherwise the board will probably not be functional. Although version 2.0 works I recommend waiting for the next revision.

Check the Draftsman documents for information on some of the manufacturing requirements for the PCB like size, stack up, drill sizes, etc..

# openGDROM PCB

## Overview

Development board for the Dreamcast G1 Bus (what the GD-ROM drive is connected to).

[Sub-project](#) and [Repository](#)

The repository contains:

- Altium Design Files
- Bill of Materials
- Schematic Prints
- Draftsman Board Design Documents
- Gerber Files
- Licence Information

## Fabrication

When getting the PCB manufactured make sure to use one of the tagged revisions in the repository, otherwise the board will probably not be functional. Although version 1.0 works I recommend waiting for the next revision.

Check the Draftsman documents for information on some of the manufacturing requirements for the PCB like size, stack up, drill sizes, etc..

# Maple Bus Breakout

## Overview

Breakout board for the Dreamcast Maple Bus (what the controllers are connected to).

[Sub-project](#) and [Repository](#)

The repository contains:

- Altium Design Files
- Bill of Materials
- Schematic Prints
- Draftsman Board Design Documents
- Gerber Files
- Licence Information

## Fabrication

When getting the PCB manufactured make sure to use one of the tagged revisions in the repository, otherwise the board will probably not be functional. Although version 1.0 works I recommend waiting for the next revision.

Check the Draftsman documents for information on some of the manufacturing requirements for the PCB like size, stack up, drill sizes, etc..

# Phabricator

## Overview

The [Phabricator](#) instance uses [Phabricator](#) for code review, repository hosting, bug tracking, project management, and more. It's similar to GitHub or GitLab however it does operate a bit differently. I chose it because it has a lot of features while still being lightweight. This is the hub for all the software and electronics collaboration.

Phabricator uses [Projects](#) as an organizational tool. Each project has its own workboard for tasks, bug reports, etc... and each project can have specific members. A project doesn't necessarily need to be a "project" it can be used as a forum or just for general planning.

[Diffusion](#) is used to host repositories. It has the ability to pull repositories from external sources as well as push to them. Right now I have it setup to push repositories to GitHub and mirror some related Dreamcast projects.

[Differential](#) is used for pre-commit code [review](#), this is one of the big differences between Phabricator and GitHub or GitLab. Instead of forking a repository, creating a feature branch, and submitting a pull request with Differential you just submit a patch that can be reviewed before being merged. This can be done manually or ideally by using [Arcanist](#). Arcanist has a bunch of other useful features that I'm honestly still figuring out.

## Official Phabricator Documentation

[User Guide](#)

[Differential](#)

[Arcanist](#)

[Diffusion](#)

[Projects](#)

# InvenTree

## Overview

InvenTree is used to create a Dreamcast hardware database. A database of model numbers, parts, assemblies, variants, and how all of those things relate to each other. Everything can have a picture, attached datasheets, links to documentation, part parameters, notes about the specific thing, etc...

Having this catalog of information can be useful in for repairs, as you could look up a part and what parameters it has to find a suitable replacement. It could be a tool for reverse engineering as being able to see how designs changed overtime or oddball versions can be clues to how things work. It's also just cool!

## Notes

Major models, parts, or assemblies have a base template which will have the majority of the information about that object. Any variants based off of that template will only contain information about how that object differs from the template. Since InvenTree doesn't support nested templates/variants or any kind of automatic child inheritance this is the best thing I could think of to make things easier to find.

A part number ending in -X\* can indicate that the part is an alternate or revision.

On a PCB the number 171-\*\*\*\* is typically the PCB manufacturer/assembler part number.

## Resources

[GitHub](#)

[Website](#)

[Documentation](#)

[http://users.skynet.be/t\\_chan/dc/mylist.htm](http://users.skynet.be/t_chan/dc/mylist.htm)

<https://commons.wikimedia.org/> (for images)

# BookStack

## Overview

BookStack is used as a knowledge base and repository of Dreamcast information. Like Wikipedia but with a better interface and structure. As the name implies all information is stored in books. The books consist of chapters and pages that can be arranged onto multiple shelves.

This is for pretty much any information about the Dreamcast, not just hardware related (that's just what I'm into). Explanations about how the Dreamcast works on a hardware and software level. Investigations into weird oddball hardware. History and timelines. Deep dives into specific subjects. Official documentation along with improved versions and analysis. Documentation for anything Keep Dreaming Project related. Preserving existing research and information.

InvenTree serves as a companion with detailed information about parts and hardware, since it's better suited for that kind of information. However it's limited to just that and will link back to the relevant information for that hardware here on BookStack.

## Resources

[Website](#)

[Documentation](#)

[Repository](#)

# Arcanist

“ Arcanist provides command-line access to many Phabricator tools (like Differential, Files, and Paste), integrates with static analysis ("lint") and unit tests, and manages common workflows like getting changes into Differential for review.

Arcanist is a wrapper script that sits on top of other tools (e.g., Differential, linters, unit test frameworks, git, Mercurial, and SVN) and provides a simple command-line API to manage code review and some related revision control operations.

## Workflow

## Setup

### SSH

To authenticate over SSH, users should add their public keys under Settings → SSH Public Keys.

I've uploaded a text file of the full arcanist command reference which is in the attachments.

## VSCode

There is a pretty good [Phabricator Extension for VSCode](#) that I recommend using.

## Windows

To set VSCode as the editor for arcanist you'll need to use

```
arc set-config editor "\"C:\Program Files\Microsoft VS Code\bin\code.cmd\" --wait -n"
```

## Installation

Some quick installation tips, for more detailed instructions checkout the [Links](#).

When cloning arcanist and libphutil you'll want to so in a more permanent directory as you'll be adding it to your `PATH`. You should also clone the stable branch as the master branch is constantly under development

```
git clone -b stable https://github.com/phacility/libphutil.git
git clone -b stable https://github.com/phacility/arcanist.git
```

## Windows

On Windows you should use PHP 7.3 instead of 7.4 as the newer version can cause some problems, although it has been awhile since I tested it.

## Links

[Arcanist User Guide](#)

[Arcanist Quick Start](#)

[Arcanist User Guide: Windows](#)

[Arcanist User Guide: Mac OS X](#)

[Arcanist User Guide: Configuring a New Project](#)

[Arcanist User Guide: arc diff](#)

[Arcanist User Guide: Lint](#)

[Arcanist User Guide: Code Coverage](#)