

PIGWEED

Search

- Home
- Getting Started**
- Source Code
- Code Reviews
- Mailing List
- Chat Room
- Issue Tracker
- Contributing
- Code of Conduct
- Embedded C++ Guide
- Style Guide
- Automated Analysis
- OS Abstraction Layers
- Targets
- Build System
- Size Optimizations
- FAQ
- Module Structure
- Module Guides
- Third Party Support

Getting Started

This guide will walk you through the typical upstream development workflow.

Note

This documentation and the [sample project](#) show how to use Pigweed as a library in your existing project. Using Pigweed as the foundation for new projects is our intended use case, but you may need more guidance than this documentation provides to do that right now.

We're happy to help you get your project setup; just drop in our [chat room](#) or send a note to the [mailing list](#).

Express setup

If you'd like to skip the detailed explanations, below is the shorter version of getting setup for Pigweed. If you run into trouble, look at the more in-depth guide below, starting at [Prerequisites](#). The express setup configures Pigweed's watcher for three targets to give a taste of Pigweed:

- Host** - Mac, Linux, or Windows. Builds and runs tests
- Device/STM32F429** - Build only; Optionally, the STM32F429-DISC1 kit to follow along later in the guide to run tests directly on said device(s)
- Docs** - Builds the Pigweed docs

To get setup:

- Make sure you have Git and Python installed** and on your path.
- Clone Pigweed and bootstrap the environment** (compiler setup & more). **Be patient, this step downloads ~1GB of LLVM, GCC, and other tooling.**

```
$ cd ~
$ git clone https://pigweed.googlesource.com/pigweed/pigweed
...
$ cd pigweed
$ source ./bootstrap.sh (On Linux & Mac)
$ bootstrap.bat (On Windows)
...
```

- Configure the GN build.

```
$ gn gen out
Done. Made 1847 targets from 91 files in 114ms
```

- Start the watcher. The watcher will invoke Ninja to build all the targets

```
$ pw watch
[Progress bars and logs]

20200707 17:24:06 INF Starting Pigweed build watcher
20200707 17:24:06 INF Will build [1/1]: out
20200707 17:24:06 INF Attaching filesystem watcher to $HOME/work/pigweed/...
20200707 17:24:06 INF Triggering initial build...
...
```

- Congratulations, you're ready to go!** Now take Pigweed for a spin by making a test fail.
- With the watcher running in a separate window, edit `pw_status/status_test.cc` to make an expectation fail; for example, add `EXPECT_EQ(0, 1);` in a test.
- Save the file. Observe the watcher rebuild & retest, and fail. Restore the test if you feel like it.
- Open the generated docs in `out/docs/gen/docs/html/index.html` in your browser.
- Edit `docs/getting_started.rst` (this file) and make any change. Save. See the watcher rebuild the docs. Reload your browser, and see the changes.

See below for equivalent Windows commands, and for more details on what each part does.

Note: After running bootstrap once, use `source ./activate.sh` (or `activate.bat` on Windows) to re-activate the environment without re-bootstrapping.

Prerequisites

CONTENTS

- Express setup
 - Prerequisites
 - Bootstrap
 - Pigweed Environment
 - Build Pigweed for Host
 - Running Unit Tests
 - Building for a Device
 - Running Tests on a Device
 - Connect Device(s)
 - Launch Test Server
 - Configure GN
 - Done!
- Building the Documentation
- Next steps
 - Check out other modules
 - Check out the sample project
 - Check out the Hackaday Supercon talk about Pigweed
 - Set up Pigweed for your own project