OS Abstraction Layers

Pigweed's operating system abstraction layers are portable and configurable building blocks, giving users full control while maintaining high performance and low overhead.

Although we primarily target smaller-footprint MMU-less 32-bit microcontrollers, the OS abstraction layers are written to work on everything from single-core bare metal low end microcontrollers to asymmetric multiprocessing (AMP) and symmetric multiprocessing (SMP) embedded systems using Real Time Operating Systems (RTOS). They even fully work on your developer workstation on Linux, Windows, or MacOS!

Pigweed has ports for the following systems:

Environment	Status
STL (Mac, Window, & Linux)	✓ **Supported**
FreeRTOS	✓ **Supported**
Azure RTOS (formerly ThreadX)	✓ **Supported**
SEGGER embOS	✓ **Supported**
Baremetal	*In Progress*
Zephyr	Planned
CMSIS-RTOS API v2 & RTX5	Planned

Pigweed's OS abstraction layers are divided by the **functional grouping of the primitives**. Many of our APIs are similar or **nearly identical to C++'s Standard Template Library (STL)** with the notable exception that we do not support exceptions. We opted to follow the STL's APIs partially because they are relatively well thought out and many developers are already familiar with them, but also because this means they are compatible with existing helpers in the STL; for example, `std::lock_guard`.

Time Primitives

The pw_chrono module provides the building blocks for expressing durations, timestamps, and acquiring the current time. This in turn is used by other modules, including pw_sync and pw_thread as the basis for any time bound APIs (i.e. with timeouts and/or deadlines). Note that this module is optional and bare metal targets may opt not to use this.

Supported On	SystemClock
FreeRTOS	pw_chrono_freertos
ThreadX	pw_chrono_threadx

CONTENTS

embOS	pw_chrono_embos
STL	pw_chrono_stl
Zephyr	Planned
CMSIS-RTOS API v2 & RTX5	Planned
Baremetal	Planned