C++ Development using Docker

Pieter P

This is a guide on how to set up a C++ cross-compilation toolchain and development environment. It serves as the documentation for the RPi-Cpp-Toolchain repository. In this guide, we'll cross-compile all package such as Python, OpenCV and their dependencies from scratch.

> **Note:** building all dependencies from source might not be ideal for most use cases. It is often easier to use a sysroot specific to the operating system you plan to use on the Raspberry Pi (e.g. Raspberry Pi OS or Ubuntu) and use the system package manager (e.g. APT) to install the dependencies. This is explained on the Ubuntu to Raspberry Pi OS Cross C++ Development page.

> If you do want to build packages from source, Yocto and Buildroot are good solutions as well.

Overview

The first page guides you through the installation of the necessary tools and editors.

The second and third page explain how to build a cross-compilation toolchain using crosstool-NG, and how to use it to compile the following libraries:

- **Zlib**: compression library (OpenSSL and Python dependency)
- **OpenSSL**: cryptography library (Python dependency)
- **FFI**: foreign function interface (Python dependency, used to call C functions using ctypes)
- **Bzip2**: compression library (Python dependency)
- **GNU ncurses**: library for text-based user interfaces (Python dependency, used for the console)
- **GNU readline**: library for line-editing and history (Python dependency, used for the console)
- **GNU dbm**: library for key-value data (Python dependency)
- **SQLite**: library for embedded databases (Python dependency)
- **UUID**: library for unique identifiers (Python dependency)
- **libX11**: X11 protocol client library (Tk dependency)
- **Tcl/Tk**: graphical user interface toolkit (Python/Tkinter dependency)
- **Python 3.10.4**: Python interpreter and libraries
- **ZBar**: Bar and QR code decoding library
- **Raspberry Pi Userland**: VideoCore GPU drivers
- **VPX**: VP8/VP9 codec SDK
- **x264**: H.264/MPEG-4 AVC encoder
- **Xvid**: MPEG-4 video codec
- **FFmpeg**: library to record, convert and stream audio and video
- **OpenBLAS**: linear algebra library (NumPy dependency)
- **NumPy**: multi-dimensional array container for Python (OpenCV dependency)
- **SciPy**: Python module for mathematics, science, and engineering
- **OpenCV**: computer vision library and Python module
- **GDB Server**: on-target remote debugger
- **GNU Make**: build automation tool
- **Ninja**: faster, more light-weight build tool
- **CMake**: build system
- **Distcc**: distributed compiler wrapper (uses your computer to speed up compilation on the RPi)
- **CCache**: compiler cache
- **cURL**: tool and library for transferring data over the network (Git dependency)
- **Git**: version control system

Next, page four explains how to install these libraries to the Raspberry Pi.

The fifth page presents a small “hello world” example project in C++ that uses CMake and Google Test.

Finally, there's a page on remote on-target debugging using GDB and Visual Studio Code.

If you're in a hurry, or if you are already quite familiar with Linux, compilers, etc. you can skip straight to the “Speedrun” page, it contains all the commands you need to get the entire project running in just a couple of minutes.

Installation and setup

Installing the necessary tools and dependencies.

Building the Cross-Compilation Toolchain

To compile software for the Raspberry Pi, you need a cross-compilation toolchain. This page contains instructions for how to build one.

Cross-Compiling the Dependencies
Using the cross-compilation toolchain to build the libraries you need for your project, as well as their dependencies.

Installing the dependencies on the Pi
Installing the cross-compiled libraries from the previous step on the Raspberry Pi.

Cross-Compiling the C++ Example Project
Using the cross-compilation toolchain to build your own C++ project.

Debugging
Remote on-target debugging using GDB and VSCode.

Speedrun
Installing, building and setting everything up as fast as possible, in just a few commands.
