# C++ Development

*Pieter P*

To build a C++ application for the Raspberry Pi, you have to follow three main steps: 1. Build a cross-compilation toolchain, 2. Cross-compile the libraries you want to use, 3. Build your actual C++ application.

These pages explain how to build a cross-compilation toolchain using Crosstool-NG, and then uses it to compile the following libraries:

- Zlib: compression library (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)
- Python 3.7.4
- ZBar: Bar and QR code decoding library
- Raspberry Pi Userland (VideoCore): hardware acceleration for H.264
- FFmpeg: library to record, convert and stream audio and video
- OpenCV: computer vision library and Python module
- NumPy: multi-dimensional array container for Python (OpenCV dependency)

The third page presents a small "hello world" example project that uses CMake and Google Test.

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

## 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.